Debian Handbuch

Frank Ronneburg

Frank Ronneburg<fr@debiananwenderhandbuch.de>

1999 bis 2014

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Debian GNU/Linux Anwenderhandbuch

Copyright und Lizenz

Copyright und Lizenz

Copyright und Lizenz

Haftungsausschluss

Copyright und Lizenz Warenzeichen

Für die in Richtigkeit der in diesem Werk beschriebenen Inhalte kann keine Garantie übernommen werden. Die aufgeführten Beschreibungen und Beispiele können Fehler enthalten oder ungenau formuliert sein. Dies kann zu Fehlern in Ihrem System führen. Handeln Sie immer vorsichtig und arbeiten Sie nur an dem System, wenn Sie ein Backup aller wichtigen Daten erstellt haben. Alle Angaben wurden mit der größten Sorgfalt erstellt, trotzdem wird die Richtigkeit nicht garantiert.

Haftungsausschluss

Copyright und Lizenz Warenzeichen

Warenzeichen

Haftungsausschluss Über dieses Buch

Warenzeichen, die nicht explizit angegeben wurden, gehören ihren jeweiligen Eigentümern. Linux ist eingetragenes Warenzeichen von Linus Torvalds. Debian ist ein eingetragenes Warenzeichen von Software in the Public Interest, Inc. UNIX ist ein eingetragenes Warenzeichen von The Open Group in den Vereinigten Staaten und anderen Ländern. 386, 386sx, 486, Pentium, Pentium Pro, Pentium II und Pentium III sind Eigentum von Intel. Windows, Windows95, Windows98, WindowsNT und WinModem sind Warenzeichen von Microsoft. ThinkPad und OS/2 sind Eigentum von IBM. Netscape und Navigator sind eingetragene Warenzeichen von Netscape Communications Corporation.

Warenzeichen

Haftungsausschluss Über dieses Buch

Über dieses Buch

Dieses Buch beschreibt die Installation der Debian Linux-Distribution und führt dann über die Installation von weiteren Paketen und deren Konfiguration zu einem kompletten, sinnvoll konfigurierten System. Hierbei wird besonders Wert darauf gelegt, dass ein enger Bezug zur Praxis gewahrt bleibt. Es wird weder auf jedes einzelne Debian-Softwarepaket eingegangen noch eine komplette Aufstellung geboten. Anhand einer gut zusammengestellten Auswahl sinnvoller Komponenten führt das Buch zu einem fertigen System, mit dem sowohl im Internet gesurft als auch Briefe geschrieben oder Grafiken erstellt werden können. Fortgeschrittene Benutzer und Administratoren finden wertvolle Informationen zur Verwaltung und Absicherung des Systems.

Dieses Buch wurde komplett mit freier Software erstellt. Als Betriebssystem diente dabei natürlich von Anfang an Debian GNU/Linux. Der Text wurde mittels Vim erfasst und im DocBook XML-Format erstellt. Grafiken und Bildschirmfotos wurden mit The Gimp erstellt.

Eine ständig aktualisierte Version dieses Buches finden Sie im Internet unter:

http://debiananwenderhandbuch.de/

Mit einem so dynamischen Thema wie einer von hauptsächlich ehrenamtlich arbeitenden Entwicklern gepflegten Linux-Distribution, ist es schwierig, auf einigen hundert Seiten ständig auf dem Laufenden zu bleiben. Wenn Sie einen Fehler (Fipptehler, veränderte URLs usw.) in diesem Buch finden, so würde ich mich über eine E-Mail (an <fr@debiananwenderhandbuch.de>) freuen. In der Regel bekommen Sie von mir innerhalb einer Woche (meist viel schneller) eine Antwort. Sollten Sie einmal länger auf eine Antwort warten müssen, so ist dies ungewöhnlich, und es schadet nicht, noch einmal nachzufragen. Mails gehen verloren oder auch ich mache mal ein paar Tage Urlaub ;-).

Dieses Buch wird im Internetauftritt „Die Deutsche Bibliothek“ (http://dnb.ddb.de) im Rahmen des Projektes Deutsche Nationalbibliografie online (Katalogdatenbank ILTIS) geführt.

Wie an der langen Liste der Danksagungen zu sehen ist, wird jeder nützliche Kommentar durch Erwähnung des Namens in diesem Werk gewürdigt. Dadurch, dass Sie dieses Buch lesen, Fehler erkennen und diese an mich melden, wird die Qualität weiter verbessert. Auch für Vorschläge zur Erweiterung des Inhalts bin ich dankbar.

Vielen Dank für die Unterstützung!

Über dieses Buch

Warenzeichen Unterstützung

Unterstützung

Über dieses Buch | Kapitel 1. Linux, Debian, freie Software?

Mein Dank gilt allen, die mich bei der Arbeit an diesem Buch mit Rat, Tat, Lob und Tipps auf technischer und moralischer Seite unterstützt haben.

Dies waren:

Kirsten Adler, Ludger Artelt, Samuel Bächler, Michael Banck, Harald Barth, Uwe Bauermann, Roman Beigelbeck, Christoph Berg, Wolfgang Berkemann, Boris Bierbaum, Felix Binder, Holger Blaschka, Andreas Borde, Wolfgang Borgert, Andreas Borutta, Christian Brehm, Thilo Benner, Robin Jérôme Bringewatt, Gisbert Burckardt, Omer C. Canitez, Christoph Claus, Florian Cramer, Frank Dalichow, Sedat Dilek, Tobias Domhan, Sascha Djokic, David Ebert, Dr. Hans Peter Engelhard, Marco Fleig, Marcus Franke, Egon Frerich, Christoph Fritsche, Malte Forkel, Hubert Gäßlein, Peter H. Ganten, Ulrich Gehring, Harald Genzel, John Paul Adrian Glaubitz, Bernhard Glomm, Peter Görtz, Nico Golde, Mario Gollub, Robert Griesel, Volker Grassmuck, Beate Groschupf, Lukas Grunwald, Sven Guckes, Thomas Güttler, Peter Guyan, Oliver Haake, Karl-Heinz Haag, Norbert Hackl, Marco Hager, Kay Häusler, Marco Hager, Oliver Hahn, Silvia Hasselbach, Heinrich Häussler, Andreas Heck, Jochen Heilingbrunner, Matthias Heinz, Ralf Hellberg, Markus Hempel, Guido Hennecke, Christian Herzberg, Sebastian Hetze, Guido Heumann, Werner Heuser, Georg Hiko, Thomas Hilke, Sebastian Hofer, Max Hölzer, David Huemer, Elias Hüsler, Peter Hunold, Wolfgang Jährling, Andreas Janssen, Sven Joachim, Boris Karnikowski, Uwe Kaufmann, Christian Kern, Uwe Kerstan, Lothar Ketterer, Matthias Kluwe, Ulrich Knauss, Christian Knoke, Stephan Knoops, Klaus Knopper, Gerd Knorr, Sascha Mester, Michael Kock, Leon König, Siegfried Kramer, Matthias Kranz, Marc Joel Krakowski, Dominik Kreutzmann, Peter Kühl, Thomas Künzel, Thomas Lange, Detlef Lechner, Daniel Leidert, Hardy Linke, Wolfgang Lonien, Daniel Leidert, Dirk Leifeld, Ola Lundqvist, Stefan Lukas, Matthias Maisenbacher, Thomas Maisl, Carsten Mann, Matthias Martin, Thomas Martinkewitz, Friedhelm Mehnert, Angelika Meier, Daniel Mewes, Steffen Möller, Michael Moller, Jürgen Mußmächer, Marc F. Neininger, Georg Neis, Mike Neuhaus, Orfeo Nicolai, Olaf Peters, Andreas Pitschi, Stefan Plüger, Julius Plenz, Markus Poguntke, Kevin Price, Cornelia-Ellen Probst, Daniel E. Atencio Psille, Nikolaus Puchhammer, Alois Raunheimer, Wolfgang Reiser, Marc Riese, Sebastian Rittau, Georg Rudolph, Thomas Roskam, Ingo Saitz, Wilhelm Schaefer, Bertram Scharpf, Thorsten Sauter, Alexander Schmehl, Sebastian Schmid, Frederik Schwarzer, Jens Seidel, Florian Sievers, Matthias Schmitt, Michael Schmitt, Manfred Schober, Andreas Schockenhoff, Friedemann Schorer, Martin Schroeter, Andrea Schweer, Frank Sertic, Max Moritz Sievers, R.G. Sidler, Henning Sprang, Tim Stegmann, Eckhard Stein, Stas Stelle, Alexander Stielau, Roland Stigge, Joachim Striebinger, Bernd Sommerfeld, Joachim von Thadden, Florian Tham, Jonas Thilemann, Andreas Tille, Janto Trappe, Jhair Tocancipa Triana, Urs Tränkner, Philipp von Vangerow, Torsten Vogt, Carsten Voigt, Georg Vollmers, Holger Wagemann, Florian Walch, Bernhard Walle, Holger Wansing, Daniel Weigl, Susanne Wenz, Claus R. Wickinghoff, Michael Wiedmann, Richard Wiemann, Jens Wilke, Sebastian Will, Matthias Wolle, Jutta Wrage, Manfred Yuyun, Siegmar Zander, Wolfgang Zeikat, Thomas Zenkel, Artem Zolochevskiy.

Besonders ist die Arbeit von Michael Wiedmann hervorzuheben: Er hat die Grundlagenforschung zur Konvertierung dieses Buches aus den DocBook-XML-Quellen nach PDF betrieben.

Der Abschnitt zu wvdial stammt von Nico Golde.

Mein Dank gilt auch allen Debian-Entwicklern und Helfern, die es mit ihrer Arbeit geschafft haben, eine herausragende und einzigartige Linux-Distribution zu schaffen.

Unterstützung

◁ Über dieses Buch △ Kapitel 1. Linux, Debian, freie Software? ▷

Kapitel 1. Linux, Debian, freie Software?

Unterstützung 1.2 Was ist Debian GNU?

Inhaltsverzeichnis

Dieser Abschnitt bietet eine Einführung in die Geschichte und Entwicklung von Linux und Freier Software bzw. Open-Source-Software. Es werden dabei geschichtliche, technische und nicht-technische Aspekte beleuchtet und auch einige Beispiele für die verschiedenen Lizenzen von freier und Open-Source-Software vorgestellt.

GNU/Linux ist ein Unix-ähnliches, Multiuser/Multitasking-Betriebssystem. Der Linux-Kernel wurde ursprünglich auf der Intel-x86-Plattform entwickelt und später auf andere Prozessor-Architekturen portiert. Das GNU-System wurde von Anfang an portabel ausgelegt und konnte so sehr schnell auch auf den Linux-Kernel angepasst werden.

Ein Betriebssystem ist ein Satz von grundlegenden Programmen, die ein Rechner zum Arbeiten benötigt. Der Kern (Kernel) ist das Stück Software, das für alle Basisaufgaben (Zugriffe auf die Hardware usw.) von anderen Programmen zuständig ist. Debian verwendet als Betriebssystemkern „Linux“, eine freie Software, die von Linus Torvalds geschaffen wurde und heute von vielen hundert Programmierern auf der ganzen Welt weiterentwickelt wird. Linus koordiniert noch immer die Entwicklung des Kernels. Ein großer Teil der grundlegenden Anwendungen eines Linux-Betriebssystems stammt aus dem GNU-Projekt und ist ebenfalls als freie Software unter einer entsprechenden Lizenz freigegeben. GNU steht für „GNU's Not Unix!“ und wurde von Richard Stallman ins Leben gerufen.

Das GNU-Projekt (http://www.gnu.org) wurde 1983 gestartet, um ein vollständig freies Unix-artiges Betriebssystem zu entwickeln - das GNU-System.

„Frei“ ist dabei nicht im Sinne von „Freibier“ oder gar kostenlos zu verstehen. Die Freiheit besteht vielmehr darin, mit der Software (und dem Quellcode) alles tun zu dürfen. Die Software darf verschenkt oder verkauft werden, es darf damit Bio- oder Atomwaffenforschung betrieben werden, genauso ist der Einsatz in Schulen und Kindergärten erlaubt. Und vor allem: Der Quellcode ist einsehbar, und es ist erlaubt - und erwünscht - diesen zu ergänzen.

Varianten des GNU-Systems, die den Linux-Kernel verwenden, sind weit verbreitet; obwohl diese Systeme oft als „Linux“ bezeichnet werden, sollte man diese korrekter als GNU/Linux-Systeme bezeichnen.

Richard Stallman kündigte den Start des Projekts durch folgenden Text in den Newsgroups net.unix-wizards und net.usoft an:

From CSvax:pur-ee:inuxc!ixn5c!ihnp4!houxm!mhuxi! eagle!mit-vax!mit-eddie!RMS@MIT-OZ From: RMS%MIT-OZ@mit-eddie Newsgroups: net.unix-wizards,net.usoft Subject: new UNIX implementation Date: Tue, 27-Sep-83 12:35:59 EST Organization: MIT AI Lab, Cambridge, MA Free Unix! Starting this Thanksgiving I am going to write a complete Unix-compatible software system called GNU (for Gnu's Not Unix), and give it away free to everyone who can use it. Contributions of time, money, programs and equipment are greatly needed. To begin with, GNU will be a kernel plus all the utilities needed to write and run C programs: editor, shell, C compiler, linker, assembler, and a few other things. After this we will add a text formatter, a YACC, an Empire game, a spreadsheet, and hundreds of other things. We hope to supply, eventually, everything useful that normally comes with a Unix system, and anything else useful, including on-line and hardcopy documentation. GNU will be able to run Unix programs, but will not be identical to Unix. We will make all improvements that are convenient, based on our experience with other operating systems. In particular, we plan to have longer filenames, file version numbers, a crashproof file system, filename completion perhaps, terminal-independent display support, and eventually a Lisp-based window system through which several Lisp programs and ordinary Unix programs can share a screen. Both C and Lisp will be available as system programming languages. We will have network software based on MIT's chaosnet protocol, far superior to UUCP. We may also have something compatible with UUCP. Who Am I? I am Richard Stallman, inventor of the original much-imitated EMACS editor, now at the Artificial Intelligence Lab at MIT. I have worked extensively on compilers, editors, debuggers, command interpreters, the Incompatible Timesharing System and the Lisp Machine operating system. I pioneered terminal-independent display support in ITS. In addition I have implemented one crashproof file system and two window systems for Lisp machines. Why I Must Write GNU I consider that the golden rule requires that if I like a program I must share it with other people who like it. I cannot in good conscience sign a nondisclosure agreement or a software license agreement. So that I

can continue to use computers without violating my principles, I have decided to put together a sufficient body of free software so that I will be able to get along without any software that is not free. How You Can Contribute I am asking computer manufacturers for donations of machines and money. I'm asking individuals for donations of programs and work. One computer manufacturer has already offered to provide a machine. But we could use more. One consequence you can expect if you donate machines is that GNU will run on them at an early date. The machine had better be able to operate in a residential area, and not require sophisticated cooling or power. Individual programmers can contribute by writing a compatible duplicate of some Unix utility and giving it to me. For most projects, such part-time distributed work would be very hard to coordinate; the independently-written parts would not work together. But for the particular task of replacing Unix, this problem is absent. Most interface specifications are fixed by Unix compatibility. If each contribution works with the rest of Unix, it will probably work with the rest of GNU. If I get donations of money, I may be able to hire a few people full or part time. The salary won't be high, but I'm looking for people for whom knowing they are helping humanity is as important as money. I view this as a way of enabling dedicated people to devote their full energies to working on GNU by sparing them the need to make a living in another way. For more information, contact me. Arpanet mail: RMS@MIT-MC.ARPA Usenet: ...!mit-eddie!RMS@OZ ...!mit-vax!RMS@OZ US Snail: Richard Stallman 166 Prospect St Cambridge, MA 02139

Um die juristischen und organisatorischen Aspekte des GNU-Projekts zu betreuen und um die Verbreitung von und das Verständnis für freie Software zu fördern, gründete sich die Free Software Foundation. Über die Free Software Foundation entstanden die GNU General Public License (GPL) und die GNU Lesser General Public License (LGPL, ursprünglich GNU Library General Public License genannt), die sich weltweit als die meistverwandten Lizenzen für freie Software etablieren konnten. Die wichtigste Bedingung dieser Lizenz ist sicherlich die Verpflichtung, dass alle aus einer unter GPL stehenden Software abgeleiteten Arbeiten ebenfalls unter der GPL stehen müssen. Dieses „virulente“ Verhalten führt zu einer sehr starken Verbreitung dieser Lizenz und somit (durch den Einblick in den Quellcode) zur Veröffentlichung von Wissen auf breiter Basis.

Neben der GPL gibt es noch etliche andere Lizenzen, die sich als Lizenzen für Freie Software qualifizieren.

Das GNU-Projekt setzt sich aus Unterprojekten zusammen, die von Freiwilligen oder Unternehmen betreut werden und zumeist dem Erstellen und Pflegen einer funktionalen Komponente dienen. Diese Unterprojekte werden wiederum als „GNU-Projekte“ oder „offizielle GNU-Projekte“ bezeichnet.

Der Name des GNU-Projekts leitet sich von dem rekursiven Akronym „GNU's Not Unix“, zu Deutsch „GNU ist nicht Unix“ ab. Da Unix ursprünglich nicht nur eine Art von Betriebssystem, sondern auch ein kommerzielles Produkt bezeichnete, war diese Bezeichnung dazu gedacht, klar zu machen, dass das GNU-Projekt ein System schaffen soll, das zwar kompatibel zu, aber nicht identisch mit Unix ist.

Die Free Software Foundation (http://www.fsf.org/) ist eine steuerfreie, gemeinnützige Organisation, die Kapital für die Arbeit am GNU-Projekt aufbringt. Die FSF ist der hauptsächliche organisatorische Sponsor des GNU-Projekts. Das Ziel der FSF ist die Sicherung, der Schutz und die Förderung der Freiheit, Computer-Software zu benutzen, zu studieren, zu kopieren, zu verändern und zu verteilen, und daneben will sie die Rechte der Benutzer von Freier Software verteidigen.

Die FSF Europe (http://www.fsfe.org/) widmet sich den europäischen Aktivitäten im Bereich freie Software. Als offizielle Schwesterorganisation der Free Software Foundation in den Vereinigten Staaten wird sie ihre Aktivitäten im Umkreis des GNU-Projekts konzentrieren, aber nicht auf diese beschränken. Die FSF Europe nahm am 10. März 2001 den Betrieb auf.

Die Hauptaufgaben der FSF Europe sind es, Initiativen Freier Software in Europa zu koordinieren, ein Kompetenzzentrum für Politiker und Journalisten bereitzustellen und Infrastruktur für Freie-Software-Projekte und speziell das GNU-Projekt zur Verfügung zu stellen.

Leider ist heute nicht mehr der genaue Tag bekannt, an dem Linus Benedict Torvalds (alle Welt nennt ihn nur Linus, wundern Sie sich also nicht, wenn nicht immer der ganze Name genannt wird) mit der Entwicklung des Linux-Kernels begann. Jedoch lässt folgendes Posting aus dem Usenet schon recht genau auf die ersten Aktivitäten schließen:

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix Subject: Gcc-1.40 and a posix-question
Message-ID: <1991Jul3.100050.9886@klaava.Helsinki.FI> Date: 3 Jul

91 10:00:50 GMT Hello netlanders, Due to a project I'm working on (in minix), I'm interested in the posix standard definition. Could somebody please point me to a (preferably) machine-readable format of the latest posix rules? Ftp-sites would be nice.

Linus begann auf Basis von Minix (http://www.minix3.org/) mit der Entwicklung eines freien Unix-Kernels.

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds) Newsgroups: comp.os.minix Subject: What would you like to see most in minix? Summary: small poll for my new operating system Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.FI> Date: 25 Aug 91 20:57:08 GMT Organization: University of Helsinki Hello everybody out there using minix - I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things). I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-) Linus (torvalds@kruuna.helsinki.fi) PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT protable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-(.

Die erste Version des Linux-Kernels wurde von Linus Torvalds am 17. September 1991 im Usenet angekündigt. Bereits kurz darauf fanden sich einige kompetente Helfer, die begeistert an der Entwicklung teilnahmen.

Einen Überblick über die Geschichte der Linux-Entwicklung soll folgende Auflistung bieten:

Urknall

3. Juli 1991

Einige Gerätetreiber sowie der Festplattentreiber und einige User-Level-Funktionen sind implementiert.

0.01

17. September 1991

Linus veröffentlicht die Version 0.01 des Kernels für einen kleinen Kreis von Leuten, die Interesse an der weiteren Entwicklung bekundet haben. Natürlich ist das Archiv mit dieser historischen Version auch heute noch verfügbar: http://www.kernel.org/pub/linux/kernel/Historic/linux-0.01.tar.gz.

0.02

5. Oktober 1991

Die erste „offizielle“ Version des Linux-Kernels erscheint. Mit dieser Version laufen bereits die bash, gcc, gnu-make, gnu-sed und compress. Linus kündigt diese Version durch folgendes Posting im Usenet an:

> From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds) Newsgroups: comp.os.minix Subject: Free minix-like kernel sources for 386-AT Message-ID: <1991Oct5.054106.4647@klaava.Helsinki.FI> Date: 5 Oct 91 05:41:06 GMT Organization: University of Helsinki Do you pine for the nice days of minix-1.1, when men were men and wrote their own device drivers? Are you without a nice project and just dying to cut your teeth on a OS you can try to modify for your needs? Are you finding it frustrating when everything works on minix? No more all-nighters to get a nifty program working? Then this post might be just for you :-) As I mentioned a month(?) ago, I'm working on a free version of a minix-lookalike for AT-386 computers. It has finally reached the stage where it's even usable (though may not be depending on what you want), and I am willing to put out the sources for wider distribution. It is just version 0.02 (+1 (very small) patch already), but I've successfully run bash/gcc/gnu-make/gnu-sed/compress etc under it.

Sources for this pet project of mine can be found at nic.funet.fi (128.214.6.100) in the directory /pub/OS/Linux. The directory also contains some README-file and a couple of binaries to work under linux (bash, update and gcc, what more can you ask for :-). Full kernel source is provided, as no minix code has been used. Library sources are only partially free, so that cannot be distributed currently. The system is able to compile "as-is" and has been known to work. Heh. Sources to the binaries (bash and gcc) can be found at the same place in /pub/gnu. ALERT! WARNING! NOTE! These sources still need minix-386 to be compiled (and gcc-1.40, possibly 1.37.1, haven't tested), and you need minix to set it up if you want to run it, so it is not yet a standalone system for those of you without minix. I'm working on it. You also need to be something of a hacker to set it up (?), so for those hoping for an alternative to minix-386, please ignore me. It is currently meant for hackers interested in operating systems and 386's with access to minix. The system needs an AT-compatible harddisk (IDE is fine) and EGA/VGA. If you are still interested, please ftp the README/RELNOTES, and/or mail me for additional info. I can (well, almost) hear you asking yourselves "why?". Hurd will be out in a year (or two, or next month, who knows), and I've already got minix. This is a program for hackers by a hacker. I've enjouyed doing it, and somebody might enjoy looking at it and even modifying it for their own needs. It is still small enough to understand, use and modify, and I'm looking forward to any comments you might have. I'm also interested in hearing from anybody who has written any of the utilities/library functions for minix. If your efforts are freely distributable (under copyright or even public domain), I'd like to hear from you, so I can add them to the system. I'm using Earl Chews estdio right now (thanks for a nice and working system Earl), and similar works will be very wellcome. Your (C)'s will of course be left intact. Drop me a line if you are willing to let me use your code. Linus PS. to PHIL NELSON! I'm unable to get through to you, and keep getting "forward error - strawberry unknown domain" or something.

0.03

26. Oktober 1991

Eine weitere schon benutzbare Version.

0.10

November 1991

Eine neue Version mit vielen Verbesserungen. Auch für diese Version war noch ein fremdes Betriebssystem zur Installation notwendig.

0.11

19. Dezember 1991

Dieses war die erste Version, die eigenständig, ohne Zuhilfenahme eines anderen Betriebssystems, lauffähig war. Es gab keinen SCSI-Support, aber einige Entwickler arbeiteten bereits daran. Voraussetzung war also eine AT-Bus-Festplatte.

Es gab weder init noch login, nach dem Systemstart landete man direkt in einer bash. Es gab Ansätze für die Implementierung von Virtual Memory, es waren aber mindestens 4 MByte RAM (vier!) notwendig, um GNU-Programme, insbesondere den GCC, benutzen zu können. Ein einfacher Systemstart war aber auch schon mit 2 MByte möglich.

0.11+VM

Dezember (Weihnachten) 1991

Da viele Leute versuchten, den Kernel mit nur 2 MByte RAM zu übersetzen, und dies fehlschlug, wurde diese Version einigen Leuten zugänglich gemacht, um die virtuelle Speicherverwaltung zu testen.

0.12

5. Januar 1992

Dies war die erste Version, die mehr Funktionen hatte als unbedingt benötigt werden. Mit dieser Version wurde der Kernel unter die GPL gestellt. Die ältere Lizenz, unter der der Kernel stand, war in vielen Punkten deutlich strenger. Dies war sicherlich durch den „virulenten“ Charakter ein wichtiger Schritt zum Erfolg von Linux.

0.95

März 1992

Eine fortgeschrittene Version, noch war allerdings der für grafische Oberflächen notwendige X-Server nicht lauffähig.

0.96

April 1992

Dies war die erste Version, mit der es möglich war, das X Window-System zu betreiben.

Im Laufe der Jahre ist die Entwicklung des Linux-Kernels weit fortgeschritten. Einen Überblick über die einzelnen Teile des Kernels bietet die Seite http://www.makelinux.net/kernel_map/. Auf dieser Seite werden die Zusammenhänge der einzelnen Bausteine des Kernels dargestellt werden. Die Bereiche können auch vergrößert werden.

Am 29. Januar 1992 postete Professor Andrew Tanenbaum, der Entwickler von Minix (http://www.minix3.org/), in der Newsgroup comp.os.minix einen Artikel, der sich zu einer längeren Diskussion ausweitete. Minix war zu dieser Zeit ein Unix-artiges Betriebssystem für x86 Computer, das von Tanenbaum zu Lehrzwecken entwickelt wurde. Es ist mit dem AT&T Unix verwandt, enthält aber keinen lizenzpflichtigen Quellcode davon, so dass es kostenlos eingesetzt und vertrieben werden darf. Im Januar 1987 wurde Minix erstmals veröffentlicht, und Anwender tauschten sich über Newsgroups im Usenet aus.

Hier nun das ursprüngliche Posting von Professor Tanenbaum; der interessierte Leser findet den kompletten Thread im Abschnitt 9.1, „LINUX is obsolete“.

From: ast@cs.vu.nl (Andy Tanenbaum) Newsgroups: comp.os.minix Subject: LINUX is obsolete Date: 29 Jan 92 12:12:50 GMT Organization: Fac. Wiskunde & Informatica, Vrije Universiteit, Amsterdam I was in the U.S. for a couple of weeks, so I haven't commented much on LINUX (not that I would have said much had I been around), but for what it is worth, I have a couple of comments now. As most of you know, for me MINIX is a hobby, something that I do in the evening when I get bored writing books and there are no major wars, revolutions, or senate hearings being televised live on CNN. My real job is a professor and researcher in the area of operating systems. As a result of my occupation, I think I know a bit about where operating are going in the next decade or so. Two aspects stand out: 1. MICROKERNEL VS MONOLITHIC SYSTEM Most older operating systems are monolithic, that is, the whole operating system is a single a.out file that runs in 'kernel mode.' This binary contains the process management, memory management, file system and the rest. Examples of such systems are UNIX, MS-DOS, VMS, MVS, OS/360, MULTICS, and many more. The alternative is a microkernel-based

system, in which most of the OS runs as separate processes, mostly outside the kernel. They communicate by message passing. The kernel's job is to handle the message passing, interrupt handling, low-level process management, and possibly the I/O. Examples of this design are the RC4000, Amoeba, Chorus, Mach, and the not-yet-released Windows/NT. While I could go into a long story here about the relative merits of the two designs, suffice it to say that among the people who actually design operating systems, the debate is essentially over. Microkernels have won. The only real argument for monolithic systems was performance, and there is now enough evidence showing that microkernel systems can be just as fast as monolithic systems (e.g., Rick Rashid has published papers comparing Mach 3.0 to monolithic systems) that it is now all over but the shoutin`. MINIX is a microkernel-based system. The file system and memory management are separate processes, running outside the kernel. The I/O drivers are also separate processes (in the kernel, but only because the brain-dead nature of the Intel CPUs makes that difficult to do otherwise). LINUX is a monolithic style system. This is a giant step back into the 1970s. That is like taking an existing, working C program and rewriting it in BASIC. To me, writing a monolithic system in 1991 is a truly poor idea. 2. PORTABILITY Once upon a time there was the 4004 CPU. When it grew up it became an 8008. Then it underwent plastic surgery and became the 8080. It begat the 8086, which begat the 8088, which begat the 80286, which begat the 80386, which begat the 80486, and so on unto the N-th generation. In the meantime, RISC chips happened, and some of them are running at over 100 MIPS. Speeds of 200 MIPS and more are likely in the coming years. These things are not going to suddenly vanish. What is going to happen is that they will gradually take over from the 80x86 line. They will run old MS-DOS programs by interpreting the 80386 in software. (I even wrote my own IBM PC simulator in C, which you can get by FTP from ftp.cs.vu.nl = 192.31.231.42 in dir minix/simulator.) I think it is a gross error to design an OS for any specific architecture, since that is not going to be around all that long. MINIX was designed to be reasonably portable, and has been ported from the Intel line to the 680x0 (Atari, Amiga, Macintosh), SPARC, and NS32016. LINUX is tied fairly closely to the 80x86. Not the way

to go. Don`t get me wrong, I am not unhappy with LINUX. It will get all the people who want to turn MINIX in BSD UNIX off my back. But in all honesty, I would suggest that people who want a **MODERN** "free" OS look around for a microkernel-based, portable OS, like maybe GNU or something like that. Andy Tanenbaum (ast@cs.vu.nl) P.S. Just as a random aside, Amoeba has a UNIX emulator (running in user space), but it is far from complete. If there are any people who would like to work on that, please let me know. To run Amoeba you need a few 386s, one of which needs 16M, and all of which need the WD Ethernet card.

Schon sehr früh nachdem die ersten Versionen des Linux-Kernels verbreitet wurden, ließen verschiedene Personen die Marke „Linux“ für sich registrieren. Um daraus folgende Rechtsstreitigkeiten schon früh abzuwenden, ließ Linus Torvalds mit Hilfe von Linux International gerichtlich gegen diese Eintragungen vorgehen und bekam die Marke Linux in den USA, Deutschland und anderen Ländern zugesprochen.

Die Verwaltung der Markenrechte lag zunächst bei Linux International, bis später die nicht gewinnorientierte Organisation Linux Mark Institute (LMI, http://www.linuxfoundation.org/programs/legal/trademark) gegründet wurde. Linus Torvalds übertrug LMI das Recht, Lizenzen für die Marke zu vergeben und Lizenzverstöße zu verfolgen. Die im Jahre 2000 von Linus formulierten Grundlagen (http://lwn.net/2000/0120/a/trademark.html) sind im Wesentlichen vom LMI übernommen worden und heute noch gültig.

1996-1997 gab es große Verwirrung um den Begriff Linux. Das bei IDG Books erschienene Buch von Naba Barkakati, „Linux Secrets“ trug auf dem Cover den Text: „Linux is a registered trademark of William R. Della Croce, Jr.“. Tatsächlich hatte William R. Della Croce den Begriff Linux schützen lassen. Glücklicherweise klärte sich das Problem aber dadurch, dass die Rechte an Linus Torvalds abgetreten wurden. Auch zu dieser Geschichte finden sich einige Dokumente im Netz:

Action Taken on Linux Trademark, Linux Journal, Mar 1997

Linux Trademark Dispute, Linux Journal, Aug 1997

Ownership of Linux Trademark Resolved, Linux Journal, Nov 1997

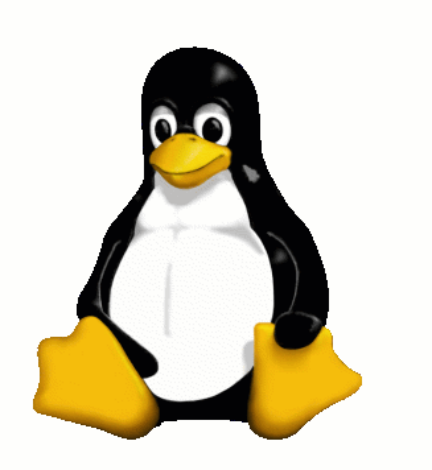
Anfang 1996 waren einige Leute auf der Linux-Kernel-Mailingliste der Meinung, es sei Zeit für ein Linux-Logo oder ein Maskottchen.

Linus legte seine Vorstellungen in folgender Mail am 9. Mai 1996 dar:

Re: Linux Logo prototype. Linus Torvalds (torvalds@cs.helsinki.fi) Thu, 9 May 1996 17:48:56 +0300 (EET DST) On Thu, 9 May 1996, Dale Scheetz wrote: > > I was unable to find a globe that displayed more than the Americas. As a > result I resorted to a map of the world instead. This is just a cut and > paste prototype made from available clip art (I am no artist) but I think > that it portrays the right idea. > Give me some feedback on this and I'll see if I can find a technically > good artist to work up a better version. > Sorry that I don't know a better place to put this. Somebody had a logo competition announcement, maybe people can send their ideas to a web-site.. Anyway, this one looks like the poor penguin is not really strong enough to hold up the world, and it's going to get squashed. Not a good, positive logo, in that respect.. Now, when you think about penguins, first take a deep calming breath, and then think "cuddly". Take another breath, and think "cute". Go back to "cuddly" for a while (and go on breathing), then think "contented". With me so far? Good.. Now, with penguins, (cuddly such), "contented" means it has either just gotten laid, or it's stuffed on herring. Take it from me, I'm an expert on penguins, those are really the only two options. Now, working on that angle, we don't really want to be associated with a randy penguin (well, we do, but it's not politic, so we won't), so we should be looking at the "stuffed to its brim with herring" angle here. So when you think "penguin", you should be imagining a slighly overweight penguin (*), sitting down after having gorged itself, and having just burped. It's sitting there with a beatific smile - the world is a good place to be when you have just eaten a few gallons of raw fish and you can feel another "burp" coming. (*) Not FAT, but you should be able to see that it's sitting down because it's really too stuffed to stand up. Think "bean bag" here. Now, if you have problems associating yourself with

something that gets off by eating raw fish, think "chocolate" or something, but you get the idea. Ok, so we should be thinking of a lovable, cuddly, stuffed penguin sitting down after having gorged itself on herring. Still with me? NOW comes the hard part. With this image firmly etched on your eyeballs, you then scetch a stylizied version of it. Not a lot of detail - just a black brush-type outline (you know the effect you get with a brush where the thickness of the line varies). THAT requires talent. Give people the outline, and they should say [sickly sweet voice, babytalk almost]"Ooh, what a cuddly penguin, I bet he is just _stuffed_ with herring", and small children will jump up and down and scream "mommy mommy, can I have one too?". Then we can do a larger version with some more detail (maybe leaning against a globe of the world, but I don't think we really want to give any "macho penguin" image here about Atlas or anything). That more detailed version can spank billy-boy to tears for all I care, or play ice-hockey with the FreeBSD demon. But the simple, single penguin would be the logo, and the others would just be that cuddly penguin being used as an actor in some tableau. Linus

Die Meinungen, ob gerade ein Pinguin geeignet wäre, Linux nach außen hin darzustellen, gingen auseinander. Es gab die verschiedensten Entwürfe in den verschiedensten Abwandlungen.



Linus Torvalds setzte sich schlussendlich durch, und so kam es zu einem Pinguin als Logo für Linux. Folgendes Statement von Linus beendete schließlich die Diskussion:

Re: Linux Logo Linus Torvalds (torvalds@cs.helsinki.fi) Sun, 12 May 1996 09:39:19 +0300 (EET DST) Umm.. You don't have any gap to fill in. "Linus likes penguins". That's it. There was even a headline on it in some Linux Journal some time ago (I was bitten by a Killer Penguin in Australia - I'm not kidding). Penguins are fun. As to why use a penguin as a logo? No good reason, really. But a logo doesn't really ave to _mean_ anything - it's the association that counts. And I can think of many worse things than have linux being associated with penguins. Having a penguin as a logo also gives more freedom to people wanting to use linux-related material: instead of being firmly fixed with a specific logo (the triangle, or just "Linux 2.0" or some other abstract thing), using something like a penguin gives people the chance to make modifications that are still recognizable.

So you can have a real live penguin on a CD cover, for example, and people will get the association. Or you can have a penguin that does something specific (a Penguin writing on wordperfect for the WP Linux CD, whatever - you get the idea).

Compare that to a

> more abstract logo (like the windows logo - it's not a bad logo in itself). You can't really do anything with a logo like that. It just "is".
>
> Anyway, go to "http://www.isc.tamu.edu/~lewing/linux/" for some nice examples..
>
> Linus

Nun brauchte das Kind noch einen Namen. Der erste Hinweis auf den Namen „Tux“ findet sich in folgender E-Mail:

> Re: Let's name the penguin! (was: Re: Linux 2.0 really _is_ released..)
> James Hughes (jamesh@interpath.com) Mon, 10 Jun 1996 20:25:52 -0400 (T)orvolds (U)ni(X) -> TUX!

Dieser Name wurde dann auch von der Community akzeptiert. Es gibt noch viele nette Geschichten um das Linux-Logo und den Kult, der sich darum entwickelt hat. Neben Stofftierchen und Schlüsselanhängern gibt es mittlerweile Linux-Pinguine in allen nur denkbaren Erscheinungsformen.

Kapitel 1. Linux, Debian, freie Software?

◁ Unterstützung △ 1.2 Was ist Debian GNU? ▷

1.2 Was ist Debian GNU?

Kapitel 1. Linux, Debian, freie Software? 1.3 Freie Software / Open Source

Debian (http://www.debian.org) ist eine freie Betriebssystem-Distribution für Computer verschiedenster Hardware-Architekturen. Ein Betriebssystem ist eine Sammlung von grundlegenden Programmen, die Ihr Rechner zum Arbeiten benötigt. Debian verwendet den Linux-Betriebssystemkern, aber die meisten grundlegenden Systemwerkzeuge stammen aus dem GNU-Projekt; daher der Name GNU/Linux.

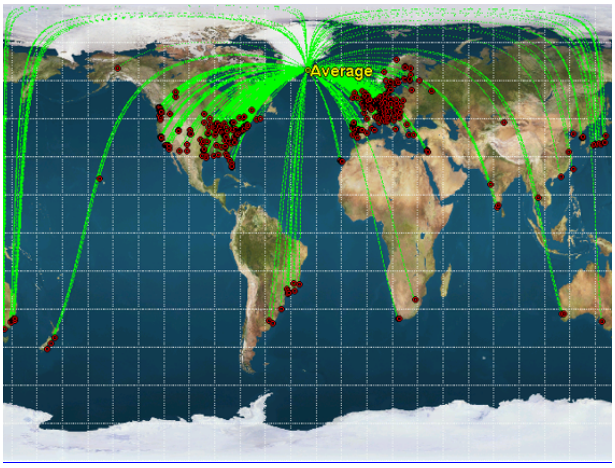
Debian GNU/Linux ist mehr als nur ein Betriebssystem: Es enthält einige tausend Softwarepakete, d.h. vorkompilierte Software in Form von einfach zu installierenden Paketen.

Das Debian-Projekt ist eine Gemeinschaft von Individuen, die in Zusammenarbeit ein freies Betriebssystem entwickeln. Dieses Betriebssystem wird Debian GNU/Linux genannt - oder einfach nur Debian.

Linux ist ein von Linus Torvalds begonnenes, komplett freies Stück Software, das von tausenden Programmierern weltweit unterstützt und weiterentwickelt wird.

Im Gegensatz zu anderen Linux-Distributionen wird Debian GNU, ähnlich wie der eigentliche Linux-Kernel, von einer großen Gruppe von Freiwilligen auf der ganzen Welt zusammengestellt. Eine Übersicht über die weltweite Verteilung der Entwickler ist in der Weltkarte der Entwickler dargestellt.

Auf Basis dieser Daten wurden von Edward Betts weitere Auswertungen vorgenommen und Grafiken erstellt. Aus diesen ist beispielsweise ersichtlich, dass der durchschnittliche Debian-Entwickler kurz vor der Küste von Grönland lebt.



Zur Kommunikation der Entwickler untereinander und zur Verteilung der Pakete dient das Internet. Für jedes der (mittlerweile einige tausend) Debian GNU-Pakete gibt es einen so genannten „Maintainer“, der dieses Paket betreut. Viele der Maintainer betreuen auch mehrere Pakete.

Die Debian Entwickler haben ihren hohen Anspruch an die freie Verbreitung von Debian im *Debian Social Contract* festgeschrieben. Die deutsche Übersetzung des Social Contracts findet sich im Abschnitt „Gesellschaftsvertrag“ Abschnitt 8.3, „Debian Gesellschaftsvertrag“

Der Name „Debian“ stammt vom Schöpfer der Distribution, Ian Murdock, der den Namen aus dem Namen seiner damaligen Freundin und späteren Frau (mittlerweile sind die beiden aber geschieden), Debra, und seinem eigenen Vornamen bildete (Deb-Ian). Die offizielle Aussprache für den Namen ist: „deb'ee`n“.

Debian 0.01 bis 0.90

(August-Dezember 1993): Das Debian-Projekt wurde offiziell von Ian Murdock am 16. August 1993 gegründet. Ian Murdock begann diese neue Distribution als offenes Entwicklungsprojekt, ganz im Sinne des GNU- oder auch des Linux-Kernel-Projekts. Dieses

Ziel erfüllte zu dieser Zeit keine andere Distribution. Das Debian-Projekt wurde vom November 1994 an vom GNU-Projekt der FSF (Free Software Foundation) für 12 Monate gesponsert.

Aus heutiger Sicht ist es spannend, noch einmal einen Blick auf die Ankündigung von Ian Murdock und die darin beschriebenen Ziele zu werfen.

Newsgroups: comp.os.linux.development From: imurdock@shell.portal.com (Ian A Murdock) Subject: New release under development; suggestions requested Message-ID: <CBusDD.MIK@unix.portal.com> Sender: news@unix.portal.com Nntp-Posting-Host: jobe.unix.portal.com Organization: Portal Communications Company - 408/973-9111 (voice) 408/973-8091 (data) Date: Mon, 16 Aug 1993 13:05:37 GMT Lines: 86 Fellow Linuxers, This is just to announce the imminent completion of a brand-new Linux release, which I'm calling the Debian Linux Release. This is a release that I have put together basically from scratch; in other words, I didn't simply make some changes to SLS and call it a new release. I was inspired to put together thisrelease after running SLS and generally being dissatisfied with much of it, and after much altering of SLS I decided that it would be easier to start from scratch. The base system is now virtually complete (though I'm still looking around to make sure that I grabbed the most recent sources for everything), and I'd like to get some feedback before I add the "fancy" stuff. Please note that this release is not yet completed and may not be for several more weeks; however, I thought I'd post now to perhaps draw a few people out of the woodwork. Specifically, I'm looking for: 1) someone who will eventually be willing to allow me to upload the release to their anonymous ftp-site. Please contact me. Be warned that it will be rather large :) 2) comments, suggestions, advice, etc. from the Linux community. This is your chance to suggest specific packages, series, or anything you'd like to see part of the final release. Don't assume that because a package is in SLS that it will necessarily be included in the Debian release! Things like ls and cat are a given, but if there's anything that's in SLS that you couldn't live without please let me know! I'd also like suggestions for specific features for the release. For example, a friend of mine here suggested that

undesired packages should be selected BEFORE the installation procedure begins so the installer doesn't have to babysit the installation. Suggestions along that line are also welcomed. What will make this release better than SLS? This: 1) Debian will be sleeker and slimmer. No more multiple binaries and manpages. 2) Debian will contain the most up-to-date of everything. The system will be easy to keep up-to-date with a 'upgrading' script in the base system which will allow complete integration of upgrade packages. 3) Debian will contain a installation procedure that doesn't need to be babysat; simply install the basedisk, copy the distribution disks to the harddrive, answer some question about what packages you want or don't want installed, and let the machine install the release while you do more interesting things. 4) Debian will contain a system setup procedure that will attempt to setup and configure everything from fstab to Xconfig. 5) Debian will contain a menu system that WORKS... menu-driven package installation and upgrading utility, menu-driven system setup, menu-driven help system, and menu-driven system administration. 6) Debian will make Linux easier for users who don't have access to the Internet. Currently, users are stuck with whatever comes with SLS. Non-Internet users will have the option of receiving periodic upgrade packages to apply to their system. They will also have the option of selecting from a huge library of additional packages that will not be included in the base system. This library will contain packages like the S3 X-server, nethack and Seyon; basically packages that you and I can ftp but non-netters cannot access. 7) Debian will be extensively documented (more than just a few READMEs). 8) As I put together Debian, I am keeping a meticulous record of where I got everything. This will allow the end-user to not only know where to get the source, but whether or not the most recent version is a part of Debian. This record will help to keep the Debian release as up-to-date as possible. 9) Lots more, but I'll detail later... Anyway, I'll provide more specifics in a week or so after I receive enough replies. Please, all replies by mail. I'll post a followup. If you wish to discuss this in the newsgroup, please don't turn it into a flamewar. :) Until later, Ian --

Ian Murdock
The Linux Warehouse
Internet: imurdock@shell.portal.com

Debian 0.91

(Januar 1994): Diese Version hatte ein einfaches Paketsystem, mit dem Pakete installiert und gelöscht werden konnten. Ein paar Dutzend Entwickler arbeiteten zu dieser Zeit an Debian.

Debian 0.93R5

(März 1995): Zu dieser Zeit waren für jedes Paket ein oder mehrere Entwickler zuständig. Das Paketmanagement wurde über dpkg abgewickelt, das nach der Basisinstallation eingesetzt wird.

Im August 1995 startete Hartmut Koptein die erste Portierung von Debian GNU/Linux auf eine Nicht-Intel-Plattform. Hartmut schrieb zu seinen ersten Versuchen mit der m68k-Architektur:

> Many, many packages were i386-centric (little endian, -m486, -O6 and all for libc4) and it was a hard time to get a starting base of packages on my machine (an Atari Medusa 68040, 32 MHz). After three months (in November 1995), I uploaded 200 packages from 250 available packages, all for libc5!

Später startete er, zusammen mit Vincent Renardias und Martin „Joey“ Schulze, eine weitere Portierung, diesmal auf die PowerPC-Plattform.

Debian 0.93R6

(November 1995): dselect wurde eingeführt. Dies war die letzte Debian Version, die noch auf dem Binärformat „a.out“ basierte. Circa 60 Entwickler arbeiteten an Debian GNU/Linux.

Debian 1.1

Buzz (Juni 1996): Die erste Version mit einem so genannten Kodenamen. Wie alle späteren stammt auch dieser Kodename aus dem Film „Toy Story“. Diese Idee wurde von Bruce Perens, der zu dieser Zeit der „Project-Leader“ war, eingeführt. Diese Version benutzte ausschließlich das neue ELF-Format sowie die Kernel-Version 2.0 und umfasste 474 Pakete.

Ian kommentierte den neuen „Project Leader“ wie folgt: „Bruce was the natural choice to succeed me, as he had been maintaining the base system for nearly a year, and he had been picking up the slack as the amount of time I could devote to Debian declined rapidly.“

Bruce initiierte viele wichtige Projekte innerhalb des Debian-Projekts. Hierzu gehören die „Debian Free Software Guidelines“, der „Debian Social Contract“, das „Open Hardware Project“. Weiterhin war Bruce maßgeblich an der Bildung von SPI (Software in the Public Interest, Inc.) beteiligt.

Debian 1.2

Rex (Dezember 1996): 848 Pakete, 120 Entwickler.

Debian 1.3

Bo (Juli 1997): 974 Pakete, 200 Entwickler.

Debian 2.0

Hamm (Juli 1998): Die erste Debian Version, die neben der *i386* -Architektur auch die *m68k* -Rechnerfamilie, also Amiga, Atari und Macintosh Computer, unterstützte. Diese Version, mit Ian Jackson als „Project Leader“, basierte bereits auf der glibc2 (oder aus historischer Sicht: libc6). Diese Version hatte über 1500 Pakete, und circa 400 Entwickler stellten die Pakete zusammen.

Wichert Akkerman löste Ian Jackson als „Project Leader“ im Januar 1999 ab.

Debian 2.1

Slink (9. März 1999): Ab dieser Version wurden zusätzlich die Architekturen *Alpha* und *Sparc* unterstützt. „Project Leader“ war Wichert Akkerman; 2250 Pakete gehörten zu dieser Version, die auf 2 offiziellen CDs ausgeliefert wurde. Weiterhin gehörte ab dieser Version apt - als neues Programm zur Paketverwaltung - zur Distribution.

Aufgrund einiger unerwarteter Fehler, die in letzter Minute entdeckt wurden, ist das Erscheinen von Debian 2.1 um eine Woche verschoben worden. Ursprünglich war der 02.03.1999 als Termin ins Auge gefasst worden. Der entscheidende Fehler war, dass dpkg, der Paketmanager, nur funktionierte, wenn die Ortsangabe „locale“ der Standardeinstellung (englisch) entsprach.

Es war den Entwicklern wichtiger, den Ruf der absoluten Fehlerfreiheit von Debian zu retten, als den Termin einzuhalten.

Am 21. April 1999 bildeten die Firma Corel, das K-Desktop-Projekt (KDE) und das Debian-Projekt eine Allianz, nachdem Corel verkündet hatte, eine eigene Linux-Distribution auf Debian Basis herauszubringen. Im Laufe des Sommers erschien mit Storm Linux eine weitere, auf Debian basierende Linux-Distribution.

Zur gleichen Zeit gab sich das Debian-Projekt auch ein neues Logo, diesmal in doppelter Ausfertigung: ein offizielles Logo für die Webseiten und alle von Debian erstellten Materialien und ein unoffizielles Logo für Materialien von anderen Herstellern.

Debian 2.2

Potato (15. August 2000): APT ist hier zentraler Bestandteil. Die Anzahl der Pakete wurde gegenüber der Version 2.1 fast verdoppelt. GNOME, glibc 2.1, Kernel 2.2.13/14 und XFree 3.3.6 sind integriert.

Mit dieser Version wurden auch die Architekturen *PowerPC* und *Arm* freigegeben.

Diese Version ist Joel Klecker gewidmet, einem Debian Entwickler, der unerwartet im Alter von 21 Jahren starb (http://www.debian.org/News/2000/20000815).

Das Release 1 (Debian GNU/Linux 2.2r1) wurde am 14.11.2000 freigegeben, das Release 2, folgte am 5.12.2000. Die letzte Version, Release 3, folgte am 17.4.2001.

Am 29. März 2001 wurde Ben Collins zum neuen „Debian Project Leader“ (DPL) gewählt.

Das Release 3 wurde vom Pressesprecher des Projekts, Martin „Joey“ Schulze, in folgender Mail angekündigt:

Date: Tue, 17 Apr 2001 00:43:00 +0200 From: Martin Schulze <joey@finlandia.infodrom.north.de> To: Debian Announcements <debian-announce@lists.debian.org> Subject: Debian GNU/Linux 2.2r3 released Content-Length: 4923 Lines: 134 ---------- The Debian Project http://www.debian.org/ Debian GNU/Linux 2.2r3 released press@debian.org April 17, 2001 ---------- The third revision of Debian GNU/Linux 2.2 (nickname `potato') has been released. This point release, version 2.2r3, mostly includes security updates, along with a few corrections to important bugs in the stable distribution. Upgrading to this revision online can be done by pointing the `apt' package tool (see sources.list(5) manual page) to one of Debian's many FTP mirrors. A list is available at: www.debian.org/distrib/ftplist Security Updates This release

contains the following security updates, for which the Security Team has released the advisory listed. Debian Security Advisory ID Package(s) DSA 004 nano DSA 005 slocate DSA 008 dialog DSA 009 stunnel DSA 010 gnupg DSA 011 mgetty DSA 012 micq DSA 013 mysql, mysql-gpl DSA 014 splitvt DSA 015 sash DSA 016 wu-ftpd DSA 017 jazip DSA 018 tinyproxy DSA 019 squid DSA 021 apache, apache-ssl DSA 022 exmh DSA 024 cron DSA 025 openssh DSA 026 bind DSA 027 openssh DSA 028 man-db DSA 029 proftpd DSA 030 xfree86 DSA 031 sudo DSA 032 proftpd DSA 033 analog DSA 034 eperl DSA 035 man2html DSA 036 mc, gmc DSA 037 nextaw, xaw3d, xaw95 DSA 038 sgml-tools DSA 039 glibc DSA 040 slrn DSA 042 gnuserv, xemacs21 DSA 044 mailx DSA 047 kernel 19 Nov 2000 cupsys 21 Nov 2000 ethereal 20 Nov 2000 tcpdump Miscellaneous Bugfixes acroread Corrected NLS handling aview Corrected dependencies boot-floppies Many improvements cslatex Important correction elvis-tiny Corrected file recovery glibc Corrected security upload mtools Corrected for arm architecture netpbm,-nonfree hpcdtoppm is non-free, sorry nvi Fixes potential file corruption postfix Fixes potential black hole for mails postgresql Fixes potential data loss python Fixes file creation problem syslog-ng Fixes potential DoS problem w3m, w3m-ssl Fixes old security problem watchdog Fixes suspicious reboots xpdf, xpdf-i Correction to security update xtide Fix copyright violation xviddetect Added support for more graphic cards yaboot Fixes a serious booting bug A complete list of all accepted and rejected packages together with rationale can be found here: people.debian.org/~joey/2.2r3/ URLs A complete list of the packages that have changed with this release can be

found at: http.us.debian.org/debian/dists/Debian2.2r3/ChangeLog non-us.debian.org/debian-non-US/dists/Debian2.2r3/non-US/ChangeLog The current stable distribution can be found at: <ftp://ftp.debian.org/debian/dists/stable> <ftp://non-us.debian.org/debian-non-US/dists/stable> Proposed updates to the stable distribution can be found at: <ftp:/ftp.debian.org/debian/dists/proposed-updates> <ftp:/non-us.debian.org/debian-non-US/dists/proposed-updates> Stable distribution information (release notes, errata, etc.): www.debian.org/releases/stable/ Security announcements and information: security.debian.org/ About Debian The Debian Project is an organization of free software developers who volunteer their time and effort in order to produce completely free operating systems Debian GNU/Linux and Debian GNU/Hurd. Contact Information For further information, please visit the Debian web pages at www.debian.org/ or send mail to <press@debian.org>.

Am 5. November 2001 erschien das Release 4 von Debian 2.2. Die dazugehörige Pressemitteilung findet sich unter http://www.debian.org/News/2001/20011105. Dieses Release, Revision 2.2r4, enthält hauptsächlich Sicherheits-Updates sowie Korrekturen von einigen ernsthaften Fehlern in der stabilen Distribution.

Die fünfte Revision von Debian GNU/Linux 2.2 (Kodename „Potato“) wurde am 10. Januar 2002 freigegeben. Diese Revision enthält hauptsächlich Sicherheitsaktualisierungen sowie Korrekturen von einigen ernsthaften Fehlern in der stabilen Distribution. Benutzer, die regelmäßig von http://security.debian.org aktualisierten, mussten kaum Pakete aktualisieren. Eine Übersicht der aktualisierten Pakete findet sich unter http://www.debian.org/News/2002/20020110.

Mit dem sechsten Release wurde am 3. April 2002 eine weitere Aktualisierung des „Potato“ Release freigegeben. Eine Übersicht der veränderten Pakete (auch bei diesem Release handelt es sich dabei größtenteils um sicherheitskritische Veränderungen) ist unter http://http.us.debian.org/debian/dists/Debian2.2r6/ChangeLog und http://non-us.debian.org/debian-non-US/dists/Debian2.2r6/non-US/ChangeLog zu finden.

Noch vor der Freigabe von Debian 3.0 erschien am 13. Juli 2002 das siebte Release von Debian 2.2. Diese Revision enthält hauptsächlich Sicherheitsaktualisierungen sowie Korrekturen von einigen ernsthaften Fehlern in der stabilen Distribution.

Debian 3.0

Woody wurde am 19.07.2002, nach fast zwei Jahren Entwicklungszeit, freigegeben. Der Kodename für diese Version wurde mit dem Codefreeze von *Potato* am 15.1.2000 festgelegt. Am 1.7.2001 startete Anthony Towns (der Release-Manager dieser Version) mit folgender Mail die „Freeze"-Phase dieser Version:

To: debian-devel-announce@lists.debian.org Subject: Debian 3.0 (woody) Freeze Begins From: Anthony Towns <ajt@debian.org> Date: Sun, 1 Jul 2001 11:04:48 +1000 Mail-Followup-To: debian-devel-announce@lists.debian.org Sender: Anthony Towns <aj@azure.humbug.org.au> User-Agent: Mutt/1.2.5i Hello world, and welcome to a new week, a new month, and a new phase of woody's development cycle. Welcome to the woody freeze. As previously proposed, the freeze will proceed in four phases: first policy will be frozen, followed by the base system, followed by standard installs, and concluding with the remainder of Debian. The aim of this first part of the freeze is to finalise our expectations of the release (what we want packages to look like, what architectures we're going to release) and to prepare ourselves for the freezing the base system by ensuring that the base system is releasable. Note that this does *not* involve a freeze on package development yet: bugfixes, and new features are still welcome, and will continue being added to woody in the usual way. What it does mean is that your packages will be frozen in the near future, so now is probably a good time to limit yourself to only introducing new features that have already been heavily tested upstream, and fixing bugs. In detail, the goals for this phase are: * Finalise debian-policy: accept any further proposals that woody packages should concern themselves with; and ensure -policy is a useful document for people working on quality assurance. Deadline: final version of debian-policy for woody needs to be uploaded to the archive by July 21st. * Finalise our target architectures. As well as alpha, arm, i386, m68k, powerpc and sparc, we have the opportunity to include ia64 (Intel's new 64bit Itanium architecture), hppa (HP's PA-RISC architecture), mips and mipsel (SGI and Decstation machines), too. Requirements for inclusion in woody are fairly simple and have been met, or are close to being met, by all those architectures. For reference, they are: a working, relatively stable toolchain, a usable system (including all of base and standard; and a fair chunk of

optional and extra), and a functional install. (Hurd people, see below) Deadline: someone from each architecture that wants to release needs to mail -release with their current status, and a successful install report by July 24th. * Determine whether cryptographic software can be moved from non-US/main to main. Ben Collins (project leader) is hustling this through the appropriate avenues. Deadline: legal advice needs to be obtained by July 21st. * Ensure the base system is releasable on all architectures: this means making sure we know what packages, exactly, the base system consists of on all architectures; and fixing any and all release critical bugs (ie, with severities critical, grave or serious) in those packages. Deadline: base packages need to be free of RC bugs by July 21st. If all goes well, the next phase will begin on the 1st of August. If all goes incredibly well, we'll release in November. Ha ha ha. The main risk that may affect moving on to the next phase is the possiblity of finding release critical bugs in the base system that take significant amounts of time to fix. As you've noticed by a careful analysis of the subject line, the woody release will be numbered Debian 3.0, in recognition of the large number of changes made since potato. This is, to put it mildly, a somewhat controversial decision, but it's one I get to make. Personally, I'm pretty happy with the way woody's progressing, and I think by the time it's released it'll easily live up to that number - and by that I mean the "3", not the ".0". On the subject of controversial decisions, one I'm not going to make today is what to call the release after woody. That one will be made when woody is released and a new testing distribution is forked from woody. Besides which, I still haven't gotten around to rewatching Toy Story. While I may not be too concerned one way or another about the name of the next release, I do have some ideas about how it might be good to handle the next release. My overriding goal for this release was to manage to get a short, controllable freeze; one that we can get over and done with in a few months, rather than letting it drag on for seven months with no end in sight, but this came at a cost of letting the development cycle go on for quite a while: ten and a half months, as it turned out. For the next cycle (assuming this freeze actually turns out to be relatively short and controlled), I think it would be interesting to see if we can do the same thing again, with a short (2 or 3 month) development cycle, for a 5 to 7

month release cycle. Which would mean you mightn't need to worry too much about not getting the neat new feature you were planning on working on into woody, if that's any consolation. And on that note, I'm inclined to think Hurd is probably better off targetting the next freeze, (in, say, six to eight months from today) rather than woody. In particular, Hurd is at present both a difficult target to port to (and thus has a quite limited range of software when compared to the Linux ports of Debian) and isn't able to self install. In short, the freeze, she is begun. Have at it. Cheers, aj -- Anthony Towns <ajt@debian.org> Debian Release Manager

Am 16. Dezember 2002 folgte das erste Release von Debian 3.0 (3.0r1). Hier wurden einige kleinere Probleme behoben und einige Pakete aktualisiert. Mit dem sofort darauf folgenden Release 1a erschienen einige neu erstellte Pakete auf den Servern, nachdem sich kleinere Probleme ergeben hatten. Die meisten Nutzer bemerkten dies jedoch gar nicht. Details zu dieser Version von Debian finden sich unter http://lists.debian.org/debian-announce/debian-announce-2002/msg00005.html.

„Project Leader“ war von 2002 bis 2003 Bdale Garbee.

Das zweite Release von Woody (3.0r2) wurde am 21. November 2003 freigegeben. Die Ankündigung hierzu findet sich unter http://lists.debian.org/debian-announce/debian-announce-2003/msg00002.html.

„Project Leader“ war von 2003 bis 2005 Martin Michlmayer.

Das dritte Release von „Woody“ (3.0r3) wurde am 26. Oktober 2004 freigegeben. Eine komplette Übersicht der Änderungen findet sich unter http://people.debian.org/~joey/3.0r3/, die Ankündigung zu diesem Release unter http://lists.debian.org/debian-announce/debian-announce-2004/msg00000.html.

Am 1. Januar 2005 wurde das vierte Release (3.0r4) von Debian 3.0 freigegeben. Unter http://lists.debian.org/debian-announce/debian-announce-2005/msg00000.html findet sich die Ankündigung dieses Release sowie eine Liste der mit diesem Release beseitigten Fehler.

Am 10. April 2005 wurden die Ergebnisse der Wahlen des neuen Project-Leaders veröffentlicht (http://lists.debian.org/debian-devel-announce/2005/04/msg00013.html). Branden Robinson wurde als neuer Projekt-Leader gewählt.

Das fünfte Release von Woody (3.0r5) wurde am 16. April 2005 herausgegeben. Unter http://lists.debian.org/debian-announce/debian-announce-2005/msg00001.html findet sich die Ankündigung dieses Release sowie eine Liste der mit diesem Release beseitigten Fehler.

Mit dem sechsten und letzten Release von Woody (3.0r6) endet die Entwicklung dieser Debian Version am 2. Juni 2005. Kurz vor der Freigabe von Debian 3.1 „Sarge“ wurde „Woody“ nochmals aktualisiert. Die Ankündigung dazu findet sich unter http://lists.debian.org/debian-announce/debian-announce-2005/msg00002.html.

Die Sicherheitsunterstützung für Debian 3.0 wurde am 30. Juni 2006 beendet, mehr als ein Jahr nach der Veröffentlichung von Debian GNU/Linux 3.1 (Sarge) und fast vier Jahre nach der Veröffentlichung von Debian GNU/Linux 3.0 (Woody) wurde die Sicherheitsunterstützung für die alte Distribution (3.0, auch bekannt als Woody) beendet. http://www.us.debian.org/News/2006/20060601

Am 10. Januar 2007 wurde Debian 3.0 archiviert, damit wird diese Version nicht mehr auf den aktuellen Debian Servern vorgehalten und ist nun nur noch auf http://archive.debian.org/debian-archive/dists/woody/ zu finden.

Debian 3.1

Diese Debian Version wird als Sarge bezeichnet. Das Release-Datum von Sarge wurde zunächst auf den 15.09.2004 festgelegt (http://lists.debian.org/debian-devel-announce/2004/08/msg00001.html), musste aber aufgrund von verschiedenen Problemen auf einen späteren Zeitpunkt verschoben werden. Nachdem eine funktionierende Infrastruktur für Sicherheitsupdates in Sarge aufgebaut wurde und verschiedene andere Probleme beseitigt wurden, konnte Sarge am 03.05.2005 „eingefroren“ werden, und der Erscheinungstermin wurde auf den 30.05.2005 festgelegt http://lists.debian.org/debian-devel-announce/2005/05/msg00001.html. Leider konnte dieser Termin nicht ganz eingehalten werden, „Sarge“ wurde am 6. Juni 2005 freigegeben und stellt aktuell die stabile Version dar (3.1r1a, http://lists.debian.org/debian-announce/debian-announce-2005/msg00003.html). Die nur als CD-Image kurz erhältliche Version 3.1r1 wurde aufgrund eines minimalen Fehlers durch die Version 3.1r1a ersetzt.

Am 10.04.2005 wurde bekanntgegeben (http://lists.debian.org/debian-devel-announce/2005/04/msg00013.html), dass Branden Robinson die Wahlen zum neuen Debian Project Leader (DPL) gewonnen hat. Nachdem er sich bereits seit 2001 als Kandidat aufgestellt hatte, gewann Branden Robinson schließlich im Jahre 2005 die Wahl. 504 Entwickler haben ihre Stimme abgegeben, das sind über 50% aller möglichen Stimmen.

Mit der Freigabe von Sarge ist der Zweig „non-US“ auf den Debian Servern entfallen. Die zuvor in diesem Zweig enthaltenen Pakete sind zum größten Teil in den „main“-Zweig integriert worden, einige wenige Pakete sind komplett entfallen.

Die erste Aktualisierung von Sarge erfolgte noch im Jahr 2005 mit dem Release 1, siehe http://lists.debian.org/debian-announce/debian-announce-2005/msg00005.html. Die Vorbereitung für das erste Update von Sarge zogen sich über einen Zeitraum von sechs Monaten hin und kamen kurz nach dem ersten Satz an Kernel-Aktualisierungen (http://www.us.debian.org/security/2005/dsa-921 und http://www.us.debian.org/security/2005/dsa-922) für diese Veröffentlichung zu einem Ende. Diese Aktualisierung umfasst 172 Sicherheitsaktualisierungen und 16 wichtige Korrekturen.

„Project Leader“ war von 2006 bis 2007 Anthony Towns.

Am 19. April 2006 wurde die zweite Aktualisierung von Sarge freigegeben. (http://lists.debian.org/debian-announce/debian-announce-2006/msg00001.html)

Zuwachs bekam die „Sarge“-Familie mit der „Dzongkha“-Version am 19. Juli 2006. Der Informations- und Kommunikationsminister der Königlichen Regierung von Bhutan, Lyonpo Leki Dorji, startete DzongkhaLinux, eine komplett lokalisierte GNU/Linux-Distribution, die auf Debian GNU/Linux 3.1 basiert. Dies ist das erste Betriebssystem, das die Nationalsprache des Landes komplett unterstützt und komplett in Bhutan entwickelt wurde.

Das Bhutaner Amt für Informationstechnik wählte Debian aufgrund seiner hohen Vielseitigkeit und Zuverlässigkeit sowie der Garantie, dass es immer 100% freie Software bleibt. DzongkhaLinux-Entwickler haben bereits ihre Übersetzungen und Entwicklungen (Schriften, Eingabemethoden, ...) sowohl an Debian als auch die Endanwender-Anwendungen, wie GNOME, OpenOffice.org und Mozilla, weitergegeben.

Die Entwicklung von DzongkhaLinux wurde von dem International Development Research Center in Kanada und dem PANl10n-Projekt unterstützt, das darauf abzielt, lokalisierte Computer-Verwendung in viele asiatische Länder zu bringen. Das System besteht aus einer CD, die entweder installiert oder als Live-System verwendet werden kann.

Neue Gesetze in dem Land erzwingen die Verwendung der Nationalsprache in allen offiziellen Veranstaltungen und aller offiziellen Kommunikation. DzongkhaLinux ist die erste Gelegenheit für die gesamte bhutanesische Bevölkerung, das Informations- und Kommunikationszeitalter unter Verwendung ihrer eigenen Sprache zu erreichen.

Am 3. August 2006 konnte ein weiterer großer Erfolg für die Verbreitung von Debian GNU/Linux gefeiert werden. Debian GNU/Linux wurde in Extremadura (Autonome Region von Extremadura, Spanien) als verpflichtendes Betriebssystem eingesetzt. Innerhalb eines Jahres sollen alle Computer der Junta von Extremadura (Regierung der autonomen Region von Extremadura, Spanien) Office-Werkzeuge aus freier Software und gnuLinEx, die lokale Ausprägung von Debian GNU/Linux 3.1, einsetzen.
http://www.us.debian.org/News/2006/20060803

Weitere Aktualisierungen von Debian 3.1 erfolgten am 1. September 2006, 6. November 2006, 18. Februar 2007 und am 7. April 2007
(http://www.us.debian.org/News/2006/20060901,
http://www.us.debian.org/News/2006/20061106,
http://www.us.debian.org/News/2007/20070218,
http://www.us.debian.org/News/2007/20070407).

Eine weitere Debian "Sarge" Aktualisierung erfolgte am 28.12.2007 mit der Release 7, im wesentlichen Sicherheitsaktualisierungen zur alten stabilen Veröffentlichung veröffentlicht, gemeinsam mit einigen Korrekturen für ernste Probleme. (http://www.us.debian.org/News/2007/20071228) Dies ist das erste Mal, dass eine alte stabile Distribution aktualisiert wurde, während die stabile Distribution (Debian 4.0) aktuell ist.

Eine letzte Aktualisierung von Debian "Sarge" erfolgte am 13.04.2008, obwohl bereits am 29.02.2008 angekündigt wurde, daß die Sicherheitsunterstützung zum 31.03.2008 endet. Dieses achte und letzte Update beseitigt einige sicherheitsrelevante Lücken sowie einige schwerwiegende Fehler in den Programmen.

Debian 4.0

Die Debian Version 4.0 wurde von den Entwicklern „Etch“ getauft. „Etch-a-Sketch“ ist ein bei Kindern beliebtes Zeichenbrett und spielt auch im Film „Toy Story“ eine wichtige Rolle. „Etch“ ist die erste offizielle Veröffentlichung von Debian GNU/Linux, die die AMD64-Architektur enthält. Die Distribution wurde zeitgleich für insgesamt 11 Hardware-Architekturen veröffentlicht.

Debian Etch/4.0 unterstützt die KDE-, GNOME- und Xfce-Arbeitsumgebungen. Es bietet auch kryptographische Software und Kompatibilität mit dem FHS v2.3 und Software, die für Version 3.1 der LSB entwickelt wurde.

Mit dem nun komplett integrierten Installationsprozess bietet Debian GNU/Linux 4.0 von Haus aus Unterstützung für verschlüsselte Partitionen. Dieses Release führt eine neu entwickelte graphische Oberfläche für das Installationssystem ein, das Schriften mit zusammengesetzten Zeichen und komplexe Sprachen unterstützt; das Installationssystem für Debian GNU/Linux wurde nun in 58 Sprachen übersetzt.

Des Weiteren wurde, beginnend mit Debian GNU/Linux 4.0, das Paketverwaltungssystem bezüglich Sicherheit und Effektivität verbessert. Secure APT erlaubt die Überprüfung der Integrität von Paketen, die von einem Spiegel heruntergeladen wurden. Aktualisierte Paketindex-Dateien werden nicht mehr als Ganzes heruntergeladen, sondern mit kleineren Dateien gepatcht, die nur die Unterschiede zu früheren Versionen enthalten.

Etch wurde am 8. April 2007, nach 21 Monaten Entwicklungszeit, veröffentlicht (http://www.us.debian.org/News/2007/20070408). Durch den diesmal sehr kurzen Zeitraum zwischen „Freeze“ und Veröffentlichung sind viele Software-Pakete auf einem sehr aktuellen Stand.

„Project Leader“ war von 2007 bis 2008 Sam Hocevar.

Ein erstes Update von Debian 4.0 „Etch“, die Release 1 (R1), wurde am 17. August 2007 herausgegeben (http://www.debian.org/News/2007/20070817). Im wesentlichen wurden einige Aktualisieren vorgenommen die die Sicherheit des Systems verbessern.

Gleiches gilt für das zweite Update, Release 2, vom 27.12.2007. Diese Update dient ebenfalls der Sicherheit und einigen Verbesserungen.
(http://www.us.debian.org/News/2007/20071227)

Am 17.02.2008 folgte das dritte Update von Debian 4.0. Der Installer wurde aktualisiert, um die in dieser Veröffentlichung enthaltenen aktualisierten Kernel zu verwenden und zu unterstützen. Diese Änderung führt dazu, dass die alten Netzboot- und Disketten-Images nicht mehr funktionieren. Aktualisierte Versionen sind von den üblichen Stellen verfügbar.

Diese Aktualisierung enthält auch Stabilitätsverbesserungen und zusätzliche Unterstützung für SGI O2-Maschinen mit 300 MHz RM5200SC (Nevada) CPUs (Mips), welche in der zweiten Aktualisierung angekündigt, aber leider doch nicht enthalten waren. Aktualisierte Versionen des Pakets bcm43xx-fwcutter werden über www.debian.org/volatile/ vertrieben. Das Paket selbst wird von Etch mit der nächsten Aktualisierung entfernt.

Das Paket flashplugin-nonfree wurde entfernt, da hierfür keine Quellen verfügbar sind und auch keine Sicherheitsunterstützung vom Hersteller geleistet wird. Aus Sicherheitsgründen wird empfohlen, sofort alle Version von Flashplugin-nonfree und alle verbleibenen Dateien des Flash-Players von Adobe zu entfernen. Getestete Aktualisierungen werden via http://backports-master.debian.org/ zur Verfügung gestellt.

Die Ankündigung zu diesem Release findet sich unter folgender Adresse: http://www.debian.org/News/2008/20080217.

Zum 17.04.08 wurde Steve McIntyre als neuer Project-Leader gewählt. Auch bei dieser Wahl setzten die Debian-Entwickler auf die Schulze-Methode (http://de.wikipedia.org/wiki/Schulze-Methode), bei der die Wähler nicht nur einem Kandidaten ihre Stimme geben dürfen.

Bereits vor der Veröffentlichung von „Etch“ wurde der Name für das folgende Release festgelegt: „Lenny“.

Debian 5.0

Diese Version wurde am 14. Februar 2009 nach 22 Monaten fortlaufender Entwicklung veröffentlicht. Neu ist die Unterstützung der Orion-Plattform von Marvell, die in vielen Speichergeräten verwendet wird. Unterstützte Speichergeräte sind u.A. die QNAP Turbo Station, HP Media Vault mv2120 und Buffalo Kurobox Pro. Zusätzlich unterstützt Lenny jetzt mehrere Netbooks, insbesondere den Eee PC von Asus. Debian GNU/Linux 5.0 Lenny enthält auch Bauwerkzeuge für Emdebian, die es erlauben, Debian-Quellpakete auf einer Architektur für andere Architekturen zu bauen und zu verkleinern, um eingebetteten ARM-Systemen zu genügen.

„Project Leader“ war von 2008 bis 2010 Steve McIntyre. Ihm folgte von 2010 bis 2013 Stefano Zacchiroli.

Debian GNU/Linux 5.0 Lenny enthält jetzt die neue ARM EABI-Portierung Armel. Diese neue Portierung verwendet moderne und zukünftige ARM-Prozessoren effizienter. Daraus ergibt sich, dass die alte ARM-Portierung (Arm) jetzt missbilligt wird.

Die Verfügbarkeit und Aktualisierungen von OpenJDK, dem GNU Java Compiler, dem GNU Java Bytecode-Interpreter, Classpath und anderen freien Versionen der Java-Technologie von Sun in Debian GNU/Linux 5.0 erlaubt es uns, Java-basierte Anwendungen im main-Depot von Debian auszuliefern.

Weitere Verbesserungen in der Systemsicherheit stellen die Installation von verfügbaren Sicherheitsaktualisierungen vor dem ersten Neustart durch den Debian-Installer, die Reduzierung von setuid root-Programmen und offenen Ports in der Standardinstallation

und die Verwendung der Härtungsfunktionalität des GCCs beim Bau mehrerer sicherheitskritischer Pakete dar. Verschiedene Anwendungen haben auch spezielle Verbesserungen erfahren. Beispielsweise wird PHP jetzt mit dem Suhosin-Härtungspatch gebaut.

Debian 6.0

Das Debian-Projekt veröffentlichte am 06. Februar 2011, nach 22 Monaten fortlaufender Entwicklung die neue stabile Version 6.0, Codename Squeeze. Debian 6.0 erscheint das erstmals in zwei Variationen: neben Debian GNU/Linux wird mit dieser Version Debian GNU/kFreeBSD als Technologie-Vorschau eingeführt.

Debian 6.0 enthält den KDE Plasma-Desktop mit weiteren KDE-Anwendungen, die GNOME-, Xfce- und LXDE-Arbeitsplatz-Umgebungen wie auch alle Arten von Server-Applikationen. Es bietet außerdem Kompatibilität zum FHS v2.3 und enthält Software, die gemäß Version 3.2 der LSB entwickelt wurde.

Eine weitere Premiere ist der vollständig freie Linux-Kernel, der nun keine problematischen Firmware-Dateien mehr enthält. Diese wurden in separate Pakete ausgelagert und aus dem Debian-main-Archiv heraus in den non-free-Bereich unserer Archive verschoben, welcher standardmäßig nicht aktiviert ist. Auf diese Weise haben Debian-Benutzer die Möglichkeit, ein vollständig freies Betriebssystem zu verwenden, können aber die nicht-freien Firmware-Dateien falls nötig trotzdem verwenden. Firmware-Dateien, die während der Installation benötigt werden, können vom Installationssystem nachgeladen werden; besondere CD-Images und Tarball-Archive für USB-basierte Installationen sind ebenfalls verfügbar.

Debian 7.0

Nach vielen Monaten der Entwicklung wurde am 4.5.2013 vom Debian-Projekt die Version 7.0 (Codename Wheezy) veröffentlicht. Diese neue Version von Debian enthält verschiedene interessante Funktionen, von der gleichzeitigen Unterstützung mehrerer Architekturen (Multiarch) über spezielle Werkzeuge zum Einrichten privater Clouds und einem verbesserten Installer bis hin zu einer kompletten Ausstattung an Multimedia-Codecs und Front-Ends, sodass nun keine Paketdepots von Drittparteien mehr nötig sind.

Multiarch-Unterstützung, eines der Hauptziele für die Veröffentlichung von Wheezy, erlaubt es Benutzern, Pakete unterschiedlicher Architekturen auf einem System zu installieren, wobei sämtliche Abhängigkeiten automatisch aufgelöst werden.

Der Installationsprozess wurde erheblich verbessert: Debian kann nun mit Hilfe von Sprachsynthese installiert werden, was vor allem sehbehinderten Personen entgegen kommt, die keine Braille-Zeile benutzen. Dank der gemeinschaftlichen Bemühungen einer großen Zahl von Übersetzern ist das Installationssystem in 73 Sprachen erhältlich und davon mehr als ein Dutzend ebenso mit Sprachsynthese. Außerdem unterstützt Debian erstmals die Installation und den Betrieb auf 64-Bit-PCs mit UEFI (amd64), das schließt allerdings noch keine Secure Boot-Unterstützung mit ein.

„Project Leader“ wurde im April 2013 Lucas Nussbaum.

Debian 7.1

Am 15. Juni 2013 wurde eine neue, stabile Version 7.1 veröffentlicht. Diese Aktualisierung behebt hauptsächlich Sicherheitslücken der Stable-Veröffentlichung sowie einige ernste andere techische Probleme. Für alle diese Probleme wurden bereits vorab separate Sicherheitsankündigungen veröffentlicht. Auch wurde der Installer komplett überarbeitet.

20 Jahre Debian

Am 16. August 2013 feiert das Debian-Projekt seinen 20 Geburtstag. Die offizielle Feier findet während der jährlichen Debian-Konferenz DebConf13 in Le Camp in Vaumarcus, Schweiz, statt. Siehe auch: debconf13.debconf.org/birthday.xhtml.

Neben den vorgenannten statischen Versionen von Debian, diese sind mit einer Versionsnummer und einem Kodenamen versehen, gibt es verschiedene dynamische Releases, welche hauptsächlich während der Entwicklung eines neuen, statischen Release verwendet werden.

Für diese dynamischen Releases werden Namen verwendet, welche auf verschiedene Bereiche der Distribution verweisen.

stable

Dieser Name steht immer für die aktuell stabile Version von Debian. Für diese Version werden Sicherheitsupdates verfügbar gemacht.

oldstable

Ist ein Verweis auf die der aktuellen stabilen Version vorhergegangenen Version von Debian.

unstable

Ist der aktuelle Stand der Debian Entwicklung, hier finden sich Pakete in den neuesten Versionen. Häufig werden hier Änderungen eingefügt, welche Auswirkungen auf andere Komponenten der Distribution haben. Dies schließt auch die Möglichkeit mit ein, dass Pakete nicht installierbar sind oder nicht alle Abhängigkeiten erfüllt werden können.

testing

Pakete aus diesem Zweig befanden sich zuvor im Zweig „unstable“, haben aber durch verschiedene Tests bereits einen gewissen Reifegrad erreicht. Diese Pakete sind für alle Architekturen verfügbar, und alle Abhängigkeiten zur Installation werden erfüllt. Für diese Version werden Sicherheitsupdates unter http://secure-testing-master.debian.net/

verfügbar gemacht. Siehe hierzu auch http://lists.debian.org/debian-devel-announce/2005/09/msg00006.html.

frozen

Als „frozen“ bezeichnet man einen Zweig von Paketen aus „testing“, welche einen Zustand erreicht haben, der es erlaubt, diese Pakete in absehbarer Zeit als neues Release von Debian freizugeben.

experimental

Vereinzelt wird „experimental“ als Vorstufe für „unstable“ benutzt. In „experimental“ werden Änderungen ausprobiert, die umfangreiche Auswirkungen auf das gesamte System haben können. So wurde der Übergang des X Window Systems von XFree86 auf X.Org in „experimental“ erprobt. „experimental“ ist keine vollständige Sammlung von Paketen. Es enthält nur, was gerade einer besonderen Untersuchung bedarf.

Wie schon kurz beschrieben, stammen die Kodenamen der Versionen der Debian Distribution aus dem Film „Toy Story“ der Firma Pixar. Folgende Charaktere wurden bisher benutzt:

Buzz

Buzz Lightyear, der Astronaut

Rex

der Dinosaurier

Bo

Bo Peep, das Mädchen, das das Schaf behütet

Hamm

das Schweinchen

Slink

Slinky Dog, der Spielzeughund

Potato

Mr. Potato Head, die Kartoffel

Woody

der Cowboy

Sarge

der grüne Soldat

Etch

Etch-A-Sketch, ein Zeichenbrett

Lenny

Lenny, das Fernglass

Squeeze

Squeeze, die kleinen, grüne Ausserirdische mit drei Augen

Wheezy

Wheezy, der Spielzeug Pinguin mit roter Krawatte

Weiterhin wird noch der Name *Sid* verwendet; dies ist der Junge von nebenan, der Spielzeug zerstört. Der Name Sid wird für die jeweils in der Entwicklung befindliche Version verwendet, deshalb steht der Name Sid auch für „Still in development".

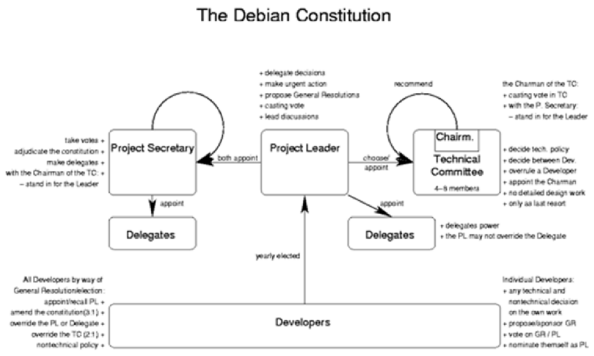
Als das heutige Sid noch nicht existierte, gab es im Debian Archiv nur einen Zweig für nicht ausgereifte Pakete. Geplant war damals, dass bis zum nächsten stabilen Release alle Pakete aus diesem Zweig reif für die Freigabe sind. Für viele Architekturen war das nicht der Fall, was dazu führte, dass diese Verzeichnisse während der Veröffentlichung

verschoben wurden. Dies war unpraktisch, da die Verschiebung zu einer großen Bandbreitenbelastung führte.

Die Debian Server-Administratoren umgingen das Problem einige Jahre, indem sie Pakete für nicht veröffentlichte Architekturen in einem speziellen Verzeichnis namens „sid“ bereitstellten. Für solche noch nicht veröffentlichten Architekturen wurde bei ihrer ersten Veröffentlichung ein Link vom aktuellen stable/ zu sid/ angelegt, und später wie üblich unter unstable/ veröffentlicht. Diese Vorgehensweise war zum Teil den Anwendern schwer zu vermitteln.

Mit Beginn der Paket-Pools während der Entwicklung der Woody-Distribution wurden Binärpakete unabhängig von der Distribution vorschriftsmäßig im Pool gehalten, so dass die Veröffentlichung einer Distribution nicht länger zu einer großen Bandbreitenverschwendung auf den Servern führte.

Der offene Charakter der Debian Distribution mag die Vermutung nahe legen, die Entwicklung sei ungeordnet oder gar chaotisch organisiert. Das Gegenteil ist jedoch der Fall. Eine fest definierte Struktur legt Aufgabenbereiche, Verantwortlichkeiten und Zuständigkeiten fest. Eine Übersicht veranschaulicht die Organisation des Debian-Projekts.



Debian GNU/Linux stellt in doppelter Hinsicht ein Betriebssystem für alle Anwender dar.

Die vollkommen freie Verfügbarkeit des Systems macht es auch für den kleinsten Geldbeutel möglich, an ein umfangreiches System mit vielen hundert Programmen zu kommen. CD-ROMs sind schon für wenige Euro erhältlich. Die Preise für Debian GNU/Linux-CDs liegen meist unter denen anderer Distributionen. Dies liegt nicht daran, dass Debian GNU/Linux minderwertiger ist. Hinter Debian steht keine Firma, die Geld verdienen muss, um zu existieren. Das Kopieren der CDs von Freunden oder Bekannten ist möglich und sogar erwünscht! Die Lizenz erlaubt die Installation von Debian GNU/Linux auf beliebig vielen Computern.

Ein weiterer Punkt ist die Verfügbarkeit von Debian GNU/Linux für die unterschiedlichsten Hardware-Architekturen. Somit lässt sich Debian GNU/Linux auf (fast) jedem vorhandenen Computer installieren. Die Palette reicht hierbei von den heute weit verbreiteten, auf der x86-Architektur basierenden Maschinen (Intel, AMD, Cyrix usw.) über PowerPC (Apple PowerMac, IBM RS/6000), Sun Sparc und UltraSparc, DEC Alpha bis hin zu älteren, auf dem Motorola 680x0er-Prozessor basierenden Systemen (Amiga, Atari oder ältere Apple-Rechner). Wichtig ist hierbei, dass mindestens eine Motorola 68020-CPU und eine -MMU (Memory Management Unit) in dem System installiert sind. Dies ist zum Beispiel beim Amiga 3000/4000, Macintosh SE/30 der Fall.

Momentan ist Debian GNU/Linux auf folgenden Architekturen verfügbar bzw. wird auf diese portiert:

Intel x86 („i386“, http://www.debian.org/ports/)

Die erste Architektur, für die Debian GNU/Linux verfügbar war.

AMD 64 („amd64“, http://www.debian.org/ports/amd64/)

Die AMD64-Portierung wartet derzeit darauf, in das offizielle Debian Archiv aufgenommen zu werden. Das Entwicklungsarchiv wird derzeit auf http://debian-amd64.alioth.debian.org/ bereitgestellt. Die Portierung besteht aus dem Kernel für alle AMD 64bit-CPUs mit AMD64-Erweiterung und für alle Intel CPUs mit EM64T-Erweiterung und gemeinsame Benutzeranwendungen.

Zurzeit besteht die Portierung aus einem inoffiziellen Archiv, welches eine komplette 64bit-Binär-Portierung der Unstable- und Testing-Distribution enthält.

Die AMD64-Portierung wird zu den offiziellen Archiven nach der Veröffentlichung von Sarge hinzugefügt. Die stabile Debian Veröffentlichung, Kodename Etch, wird zukünftig eine native AMD64-Unterstützung haben.

IA-64 („ia64“, http://www.debian.org/ports/ia64/)

Die Intel-64-Bit-Architektur, eine relativ neue Portierung.

DEC Alpha („alpha“, http://www.debian.org/ports/alpha/)

Das erste Release erfolgte mit Debian 2.1. Eine der länger bestehenden Portierungen und ziemlich stabil.

ARM („arm“, http://www.debian.org/ports/arm/)

Wurde mit „potato“ freigegeben. Eine neue Portierung, motiviert durch Corels interessante NetWinder-Maschine.

Motorola m68k („m68k“, http://www.debian.org/ports/m68k/)

Erstes Release mit der Debian Version 2.0. Der am besten etablierte Port nach dem Intel x86. Der Debian m68k-Port läuft auf einer großen Bandbreite von Computern, die auf der Motorola-68k-Prozessorfamilie basieren - im Besonderen auf der Sun3-Workstationfamilie, den Apple Macintosh Personal-Computern und den Atari und Amiga Personal-Computern.

MIPS („mips“, http://www.debian.org/ports/mips/)

Relativ neue Portierung, die in SGI-Computern (debian-mips - big-endian) und Digital DECStations (debian-mipsel - little-endian) verwendet wird.

HP PA-RISC („hppa“, http://www.debian.org/ports/hppa/)

Eine Portierung von Debian GNU/Linux auf PA-RISC wurde gerade begonnen. Die Webseiten des Projekts finden sich unter http://parisc-linux.org/.

Motorola/IBM PowerPC („powerpc“, http://www.debian.org/ports/powerpc/)

Wurde mit Debian 2.2 („potato“) das erste Mal offiziell veröffentlicht. Die Portierung läuft auf vielen der Apple Macintosh/PowerMac-Modelle, den CHRP- und PReP-Rechnern sowie beispielsweise auf einigen IBM RS/6000-Maschinen.

Sun SPARC („ sparc“, http://www.debian.org/ports/sparc/)

Zum ersten Mal mit Debian 2.1 veröffentlicht. Diese Portierung läuft sowohl auf der SPARCstation-Familie von Workstations als auch auf einem Teil ihrer Nachfolger in der Sun4-Architektur.

Sun UltraSPARC („ sparc64“, http://www.debian.org/ports/sparc64/)

Dies ist auch der Beginn einer Portierung auf die Sun UltraSPARC-(sun4u)-Workstation-Familie. Dieser 64-Bit-Prozessor hat den Vorteil der Rückwärtskompatibilität mit seinem Vorgänger, so dass die UltraSPARC-Portierung in der Lage sein wird, Sparc-Binärfiles auszuführen.

S/390 („ s390“, http://www.debian.org/ports/s390/)

Ein Port von Debian GNU/Linux auf die S/390-Architektur (IBM-Großrechner) wird gerade begonnen.

Weitere Portierungen des Debian-GNU-Systems, die nicht die Hardware betreffen, sondern statt des Linux-Kernels einen alternativen Kernel benutzen, sind:

Debian GNU/Hurd („ hurd-i386“, http://www.debian.org/ports/hurd/)

GNU/Hurd verwendet die bereits im GNU-Projekt vorhandenen Softwarepakete und bildet das letzte fehlende Stück Software zu einem kompletten Unix-artigen Betriebssystem. Das gegenwärtige Projekt ist auf der i386-Architektur aufgebaut. Der Hurd-Kernel ersetzt in diesem Projekt den in allen anderen Architekturen eingesetzten Linux-Kernel.

Debian GNU/NetBSD (http://www.debian.org/ports/netbsd/)

Dies ist eine Portierung des Debian Betriebssystems inklusive dpkg, apt und GNU-Software auf den NetBSD-Kernel. Diese Portierung befindet sich im Augenblick in einem sehr vorläufigen Stadium. Aber da NetBSD ein Produktionslevel-Kernel ist, sollte sich die Verwendbarkeit von Debian GNU/NetBSD sehr rasch entwickeln. Im Augenblick ist Debian GNU/NetBSD für Intel x86 am weitesten fortgeschritten, aber die Arbeit an der Unterstützung für Alpha-basierte Computer hat bereits begonnen.

Debian GNU/FreeBSD (http://www.debian.org/ports/freebsd/)

Es gibt derzeit zwei separate Bemühungen, eine Debian-Distribution basierend auf dem FreeBSD-Kernel zu erstellen. Beide sind noch in einem experimentellen Stadium, und es

ist noch nicht endgültig entschieden, welche von beiden das offizielle Debian GNU/FreeBSD werden wird. Auf den Port-Seiten sind die Details über den Status des jeweiligen Ports zu finden:

http://www.debian.org/ports/freebsd/gnu-libc-based Debian GNU/kFreeBSD, basierend auf der GNU-C-Bibliothek (Glibc).

Eine ausführliche Installationsanleitung ist ebenfalls verfügbar (http://glibc-bsd.alioth.debian.org/kfreebsd/install.html).

http://www.debian.org/ports/freebsd/bsd-libc-based Debian GNU/FreeBSD ist eine Version, die auf der BSD C-Bibliothek (libc5) basiert.

Debian GNU/MiNT (http://debian-mint.nocrew.org/)

Dies ist eine experimentelle Portierung von Debian auf Basis des MiNT-Kernels, ein Unix-artiger Kernel für Atari-m68k-Computer.

Debian Beowulf (http://www.debian.org/ports/beowulf/)

Obwohl nicht wirklich eine Portierung, so wird Beowulf doch ein Ersatz für viele Großrechner in Forschung und Entwicklung sein. Dieses Projekt arbeitet daran, Beowulf-Cluster unter Debian laufen zu lassen.

Die Installation unterscheidet sich auf den verschiedenen Architekturen nur in sehr wenigen Punkten. Abweichungen gibt es beim Starten des Linux-Kernels und der späteren Installation des Bootloaders.

Sobald das Debian GNU/Linux-Installationsprogramm gestartet ist, gestaltet sich die weitere Installation auf allen Architekturen im Wesentlichen gleich.

Natürlich gibt es durch die unterschiedliche Hardware der verschiedenen Architekturen einige Details zu beachten, aber dies ist auch bei unterschiedlichen i386-basierten Systemen der Fall. Beispielsweise trifft man bei der i386er-Familie auf diverse verschiedene Typen von Mäusen. Verbreitet sind hier Geräte mit seriellen und mit PS/2-Anschlüssen. Auf Apple Macintosh-Rechnern (m68k und powerpc) wird die Maus über den so genannten ADB-Anschluss versorgt, die entsprechende Gerätedatei hierzu ist /dev/adbmouse.

Man sollte sich also keinesfalls von der Lektüre dieses Buches oder der Installation von Debian GNU/Linux abhalten lassen, auch wenn man keinen i386-basierten Computer sein eigen nennt.

Debian GNU/Linux wird aber auch den verschiedenen Ansprüchen an freie Software gerecht. Soll ein System mit 100% Freier Software aufgesetzt werden, so ist dies bereits durch das Debian Team vorbereitet. Alle Programme im Bereich main sind von den Maintainern (Betreuern) der Pakete geprüft und entsprechend der Lizenz in diesen Bereich aufgenommen worden. Pakete in diesem Bereich unterliegen beispielsweise

keinerlei Einschränkungen in der Nutzung oder in der Verteilung. Alle Programme können zur Entwicklung von Gentechnik verwendet werden, und diese Software kann in jedes Land der Welt exportiert oder von dort importiert werden. Man kann geteilter Meinung darüber sein, ob man Gentechnik unterstützen will oder nicht, die Lizenz zur Software setzt aber keine Beschränkungen.

Die Debian Distribution ist allgemein als qualitativ sehr hochwertig anerkannt. Administratoren, die bereits auf Debian setzen, wissen die Vorteile zu schätzen. Trotzdem kommt es häufig zu Diskussionen, warum dieser Distribution der Vorzug gegenüber anderen, kommerziellen Distributionen gegeben werden sollte. Insbesondere bei strategischen Entscheidungen, beispielsweise der Umstellung oder Aktualisierung einer großen Anzahl von Systemen, können die folgenden Fakten als Argumentationsgrundlage helfen.

Das Debian Team hat die Vorteile von Debian auf der Webseite http://www.debian.org/intro/why_debian zusammengefasst.

Debian wird von den Benutzern gewartet

Das komplette Debian System wird von den Benutzern selbst entwickelt. Einige hundert Entwickler betreuen jeweils ein oder mehrere Programmpakete, die auch von dem jeweiligen Entwickler selbst eingesetzt werden. Daraus resultiert ein hohes Interesse an der Funktions- und Integrationsfähigkeit eines jeden Paketes. Sollte dennoch ein Problem auftauchen, so wird der Entwickler mit Sicherheit ein hohes persönliches Interesse an der schnellen Beseitigung des Problems haben.

Zuverlässigkeit zuerst

Debian setzt auf eine konservative Auswahl von Komponenten. Es ist wichtiger, dass das Gesamtsystem in jedem Fall funktioniert, auch wenn dafür an manchen Stellen auf etwas Komfort verzichtet werden muss. Als Beispiel sei hier die Verwendung des Kernels 2.2.x auch noch im Debian Release 3.0 genannt. Auf den Einsatz des Kernels in der Version 2.4.x wurde verzichtet, da das Team der Meinung ist, dass die neue Version noch nicht ausgereift genug ist und auch der Kernel 2.2 alle notwendigen Funktionen mitbringt. Natürlich stehen Pakete mit dem Kernel 2.4 zur Verfügung, die bei Bedarf einfach installiert werden können.

Lange Release-Zyklen

Die Lebensdauer einer Debian Version beträgt bisher mindestens ein Jahr. Debian 3.0 (woody) „überlebte“ sogar fast drei Jahre. Dies bedeutet einen guten Investitionsschutz in die geleistete Arbeit. Natürlich heißt das nicht, dass diese Version veraltet oder gar unsicher ist; Software-Updates werden umgehend bei Bedarf zur Verfügung gestellt und können sehr einfach installiert werden.

Sicherheit

Debian nimmt die Sicherheit sehr ernst. Der größte Teil der Probleme, auf die das Team aufmerksam wird, ist innerhalb von 48 Stunden korrigiert.

Die Erfahrung zeigt, dass „Sicherheit durch Verschleierung“ (englisch: „security through obscurity“) nicht funktioniert. Die öffentliche Entschleierung erlaubt es, bessere Lösungen für Sicherheitsprobleme schneller zu finden. Die Seite http://www.debian.org/security/ zeigt den Stand von Debian in Bezug auf verschiedene bekannte Sicherheitsmängel, die auch Debian betreffen könnten.

Um die neuesten Debian Sicherheitsmeldungen zu erhalten, dient die Mailingliste debian-security-announce. In jeder Meldung ist beschrieben, für welches Debian Release die aktualisierten Pakete verfügbar sind, wo die entsprechenden Debian Pakete zu finden sind und wie diese installiert werden.

Wenn Sie selbst eine sicherheitsrelevante Lücke in einem Debian Paket oder einer Software, die mit Debian ausgeliefert wird, entdecken, so kann diese an die E-Mail-Adresse security@debian.org gemeldet werden. (Debian Entwickler sollten die Mailinglisten „debian-security“ und „debian-security-private“ verwenden, um die Mitglieder des Security-Teams über Probleme in ihren Paketen zu informieren.)

Einfache Installation

Seit langer Zeit hält sich das Gerücht, dass ein Debian-GNU-System schwierig zu installieren sei. Wer das behauptet, hat noch keine aktuelle Debian Version ausprobiert. Der Installationsvorgang wird stetig verbessert. Die Installation kann direkt von DOS, von CD oder über das Netzwerk erfolgen. Die Beschränkung auf den kleinsten gemeinsamen Nenner garantiert die Funktion auf jeder Hardware und Prozessorarchitektur.

Upgrade-Fähigkeit

Ein Debian GNU-System kann mit verschiedenen Methoden auf eine neue Version (beispielsweise von 2.2 auf 3.0) aktualisiert werden. Dies kann per CD-ROM oder per Netzwerk von einem Server aus erfolgen. Ein Neustart des Systems ist dabei nicht erforderlich. Auch Aktualisierungen über mehrere Versionen hinweg, beispielsweise von 1.3 direkt auf die Version 3.0, sind durch das ausgereifte Paketmanagement möglich. Dies ist im kommerziellen Umfeld ein nicht zu unterschätzender Vorteil, da so bestehende Systeme mit einer minimalen Downtime auf den aktuellen Stand gebracht werden können.

Vergleiche mit anderen Distributionen haben gezeigt, dass dies ein einmaliges Merkmal von Debian ist.

Verfügbarkeit

Debian ist nicht nur auf CD-ROMs verfügbar. Über 150 FTP-Server weltweit stellen diese Distribution zur Verfügung. Einer davon steht sicher auch in Ihrer Nähe. Eine Liste findet sich unter http://www.debian.org/misc/README.mirrors.

Architekturen

Debian ist für die verschiedensten Hardware-Architekturen verfügbar und unterstützt neben dem Linux-Kernel beispielsweise auch den Hurd-Kernel (siehe „Debian für alle“).

Ein Administrator eines heterogenen Netzwerks hat damit den Vorteil, das gleiche Betriebssystem und sogar die gleiche Version auf allen Maschinen betreiben zu können. Dies senkt den Wartungsaufwand deutlich.

Große Auswahl an Software-Paketen

Debian beinhaltet einige tausend verschiedene Software-Pakete. Jedes einzelne Software-Paket ist freie Software. Wird proprietäre Software eingesetzt, die unter Linux läuft, dann können Sie diese immer noch benutzen - es kann sogar sein, dass es eine Installationsroutine in Debian gibt, die proprietäre Software automatisch installiert und einrichtet.

Klare Trennung von nicht-Freier Software

Debian legt großen Wert auf den Einsatz von Freier Software. Sie können sicher sein, dass ausschließlich freie Software auf Ihrem System installiert wird, sofern, wie vorab beschrieben, ausschließlich Software aus dem Bereich „main“ installiert wird. Wenn Sie dieses System im kommerziellen Einsatz betreiben oder Systeme an Kunden verkaufen, so müssen Sie sich nicht weiter um Lizenzen kümmern.

Software, die nicht unter einer Lizenz steht, die den Anforderungen an freie Software gerecht wird, kann natürlich auch eingesetzt werden; hierfür wurde der Bereich „non-free“ eingerichtet.

Integration

Debian Pakete sind sehr gut in das Gesamtsystem integriert. Diverse Programme und Werkzeuge tragen zu einem konsistenten und aufgeräumten System bei.

Die Basis hierfür sind fein abgestimmte Abhängigkeiten (dependencies) zwischen den Paketen. Das Menüsystem, das alle populären Programme in die Menüs der verschiedenen Windowmanager integriert, die Dokumentation zu allen Paketen, die sowohl auf der Kommandozeile als auch über Web verfügbar ist und die Unterstützung verschiedener Versionen eines Programms (beispielsweise mehrerer vim-Versionen) machen Debian einmalig.

Die Installation eines Pakets erzwingt durch die definierten Abhängigkeiten auch die Installation der benötigten Bibliotheken. Nicht-funktionierende Pakete nach der Installation gibt es somit unter Debian nicht.

Die übersichtliche Implementation der SysV-Bootskripte vereinfacht die Administration und erleichtert die Fehlersuche. Updates werden so ebenfalls einfacher, auch wenn Sie selbst Veränderungen an den Skripten vorgenommen haben.

Zusätzlich können update-rc.d und rc benutzt werden, um die SysV-Initskripts durch eigene zu ersetzen.

Quellcode

Für alle Pakete in der Distribution ist der Quellcode (Source) verfügbar. Die gesamte Distribution ist freie Software nach den Debian Free Software Guidelines (siehe „Debian Free Software Guidelines“ Abschnitt 8.4, „Debian Richtlinien für freie Software “). Das bedeutet: Jeder kann Pakete verbessern und weitergeben.

Der Quellcode der Pakete und die vom Debian Team vorgenommenen Veränderungen werden in getrennten Dateien gehalten. Somit ist es nicht notwendig, die kompletten Quellen zu besorgen, wenn ein neues Debian Paket erstellt werden soll. Dies ist insbesondere bei großen Paketen (emacs, XFree86, gcc usw.) wichtig und kann einiges an Kosten sparen.

Hohe Qualität

Die Betreuer (Maintainer) der Pakete haben ein hohes persönliches Interesse an den von ihnen betreuten Paketen. Die meisten von ihnen betreuen ein solches Paket, weil sie es selbst einsetzen. Das Ergebnis sind Pakete von höchster Qualität, die von hochmotivierten und technisch versierten Personen erstellt werden. Insgesamt führt dies zu einer qualitativ hochwertigen Distribution.

Vorkonfiguration

Jedes Debian Paket ist bereits sinnvoll vorkonfiguriert. Dadurch ist jedes Paket sofort nach der Installation einsatzbereit. Natürlich kann auch jedes Paket noch nachträglich von Hand konfiguriert werden. Die vorgenommenen Änderungen werden bei einem Update beibehalten.

Fehlerdatenbank

Das Bug Tracking System verwaltet alle bekannten Fehler in der Debian Distribution und ist öffentlich zugänglich. In den meisten Fällen werden Fehler in wenigen Tagen beseitigt.

Es wird nicht versucht, die Tatsache zu verheimlichen, dass Software nicht immer so funktioniert, wie sie funktionieren soll. Benutzer können Fehlerberichte einschicken und werden informiert, wenn der Fehlerbericht abgeschlossen wird. Dieses System erlaubt es dem Debian Team, auf Probleme sehr schnell und ehrlich zu reagieren. Die Geschichte zeigt, dass das Prinzip „Sicherheit durch Undurchschaubarkeit“ („security through obscurity“) nicht funktioniert.

Qualitätssicherung

Das Debian-Projekt legt mehr Wert auf Qualität und Stabilität als auf schnelle Neuerscheinungen. Wenn eine neue Debian Version erscheint, sind alle schwerwiegenden Fehler beseitigt, und es wurde eine ausgiebige Testphase durchlaufen.

Die gesamte Distribution wird von den Entwicklern und von interessierten Benutzern vom Anfang der Entwicklung an getestet. Die gesamte Entwicklungsversion steht während der gesamten Entwicklungszeit zum Download bereit.

Einen Überblick über den aktuellen Stand der Pakete in Bezug auf bekannte Fehler können Sie sich auf den Webseiten der „Debian Quality Assurance“ und im Bug Tracking System verschaffen.

Remote-Administration

Das gesamte Debian-GNU-System kann remote administriert werden. Dies umfasst sowohl die Konfiguration und die Paketverwaltung als auch das Hinzufügen und Löschen von Paketen.

Dieses einmalige Feature von Debian kann ohne einen Reboot des Systems zur Aktualisierung eines Systems genutzt werden. Lediglich die soeben aktualisierten Dienste stehen unter Umständen für einige Minuten nicht zur Verfügung.

Syslog

Debian benutzt auch für die Logdateien der Programme eine einheitliche Struktur. Im Verzeichnis /var/log/ finden sich zu den jeweiligen Programmen die entsprechenden Logdateien, ggf. in einem Unterverzeichnis. Somit ist sichergestellt, dass alle Logdateien an einer fest definierten Stelle im Dateisystem zu finden sind.

Support

Debian hat kein Büro in irgendeinem Land dieser Welt mit einem Supportteam. Diese Tatsache hat sich jedoch als unproblematisch erwiesen, da alle Fragen auf den Mailinglisten innerhalb weniger Stunden beantwortet werden.

Die Verantwortung für den Support wird somit von der Gemeinschaft der Debian Benutzer getragen und nicht in fremde Hände gegeben.

Ein Frage, die an eine der Mailinglisten geschickt wird, wird oft innerhalb einer Viertelstunde beantwortet, gratis und durch Entwickler.

Dabei ist aber immer zu bedenken, dass diese Arbeit der Debian Entwickler aus Freude an einem guten System erbracht wird. Es besteht keinerlei Anspruch auf Hilfe; die Erfahrung zeigt aber, dass eine kompetente Antwort nicht lange auf sich warten lässt.

Auch wenn man die Größe betrachtet, liegt die Debian Distribution ganz vorne. Hierbei bezieht sich Größe nicht nur auf die Anzahl der Pakete, sondern auch auf die Anzahl der Codezeilen der Pakete. Ein Team von spanischen Debian Benutzern hat sich die Mühe gemacht, verschiedene Daten zum Potato-Release zusammenzutragen. Die Ergebnisse

wurden bereits in verschiedenen Magazinen veröffentlicht und sind natürlich auch im Web unter http://people.debian.org/~jgb/debian-counting/ einzusehen.

Debian Popularity Contest

Mit dem „Debian Popularity Contest“ kann ermittelt werden, welche Software-Pakete auf Debian Systemen installiert sind und häufig verwendet werden. Durch die Vielzahl von Paketen, von denen viele Überschneidungen in den Aufgabenbereichen mit sich bringen, ist es oft nicht einfach zu ermitteln, welches Paket die beste Lösung für ein Problem bietet. Hier ist es hilfreich zu wissen, auf welches Programmpaket viele andere Debian Anwender setzen. Man kann dann davon ausgehen, mit diesem Paket eine gute Lösung zu finden.

Um die eigenen Systeme an dieser Auswertung teilnehmen zu lassen, muss das Paket popularity-contest installliert sein. Die Ergebnisse werden auf einem zentralen Server gesammelt (http://popcon.debian.org/). Dabei wird auch unterschieden, auf welcher Hardwarearchitektur die Pakete eingesetzt werden, falls nicht noch das „Woody“-Release von Debian eingesetzt wird.

Das Skript sammelt verschiedene Informationen und versendet diese (auf Wunsch auch automatisch) per E-Mail an den Debian-Server. Nützlich ist das Skript aber auch, um Programme zu ermitteln, die lange nicht oder noch nie benutzt wurden. Diese lassen sich mit dem Kommando

```
popularity-contest | grep '<OLD>'
```

herausfinden. Das Ergebnis ist nicht zu 100% korrekt, da beispielweise bei Bibliotheken nicht ermittelt werden kann, wann zuletzt auf diese zugegriffen wurde, aber ein guter Anhaltspunkt ist dies in jedem Fall.

1.2.9.1 Ubuntu

1.2.9.2 Univention Corporate Server (UCS)

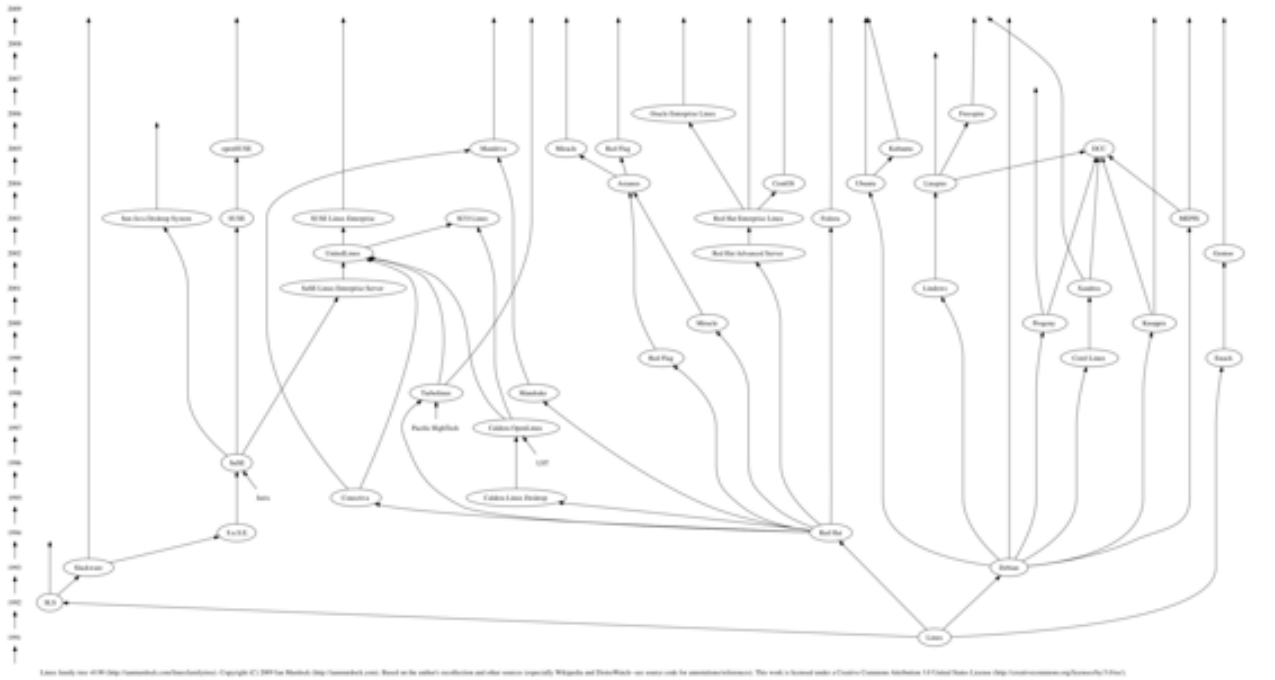
Debian GNU/Linux wird als Basis von verschiedenen Distributionen eingesetzt. Diese Distributionen ergänzen Debian an einigen Punkten, wie zum Beispiel bei der Installation

und der Auswahl der Software-Pakete, oder werden mit angepassten Konfigurationen für spezielle Einsatzzwecke ausgeliefert.

Dem Open-Source-Gedanken entsprechend stellen die Hersteller Teile oder auch die gesamten Eigenentwicklungen wieder als freie Software zur Verfügung. Einige dieser Entwicklungen sind auch bereits in die Debian-GNU/Linux-Distribution eingeflossen.

Natürlich stellen die Entwickler während der Arbeit an dem Produkt auch Fehler in der Software oder in den Debian Paketen fest. Auch diese Ergebnisse werden gemeldet bzw. von den Entwicklern gleich gefixt und fließen wieder in das Gesamtprojekt ein.

Eine Übersicht aller Linux Distributionen in Form eines Stammbaumes findet sich auf der Seite von Ian Murdock: http://ianmurdock.com/linuxfamilytree/.



Ubuntu (http://www.ubuntu.com/) ist eine auf Debian basierende Linux-Distribution. Ziel der Entwickler ist es, eine einfach zu installierende Distribution zu schaffen. So ist es beispielsweise während der Installation möglich, bestehende Windows Partitonen zu

verkleinern. Auch werden während der Installation nur die absolut notwendigen Fragen zur Konfiguration gestellt. Alles Weitere lässt sich über leicht verständliche Konfigurationsdialoge anpassen.

Ubuntu ist in der Anzahl der Software-Pakete eingeschränkt. Es wurden nur die (aus Sicht der Entwickler) besten Software-Pakete in die Distribution aufgenommen. Damit genügt eine einzige CD für die Installation. Ubuntu ist so konzipiert, dass im Lieferumfang der Basisdistribution für jede Anwendungsaufgabe lediglich ein bewährtes Programm zum Einsatz kommt. Dies bedeutet beispielsweise den kompletten Verzicht auf die KDE Umgebung, Ubuntu nutzt stattdessen GNOME. Es ist aber möglich, KDE zusammen mit Ubuntu einzusetzen.

Ubuntu wird von der Firma Canonical Ltd (http://www.canonical.com/) finanziell unterstützt. Ziel ist es, dass Canonical über bezahlte Dienstleistungen in Form von Unterstützung bei der Installation und Konfiguration von Ubuntu finanziert wird. Weitere von Canonical Ltd. unterstützte Projekte sind die Versionsverwaltungssoftware Bazaar, eine Webanwendung zur Unterstützung bei der Entwicklung von Software namens Launchpad und das TheOpenCD-Projekt.

Die Ubuntu Entwickler arbeiten (mehr oder weniger) eng mit den Debian Entwicklern zusammen, einige Entwickler arbeiten auch für beide Projekte, so dass viele Ubuntu Entwicklungen auch in die Debian Distribution Einzug halten.

Die Firma gehört dem südafrikanischen Unternehmer Mark Shuttleworth und hat ihren Hauptsitz auf der Isle of Man. Bekannt wurde Mark als erster Afrikaner im Weltraum und als zweiter Weltraumtourist an Bord der ISS.

Nach den ersten Erfolgen der Ubuntu Distribution, welche zum großen Teil in der eingeschränkten Paketanzahl und der sehr ausgereiften Konfiguration der Komponenten liegen, wurde sehr schnell der Wunsch nach Ubuntu Versionen für spezielle Einsatzbereiche geweckt. In erster Linie sind hier Kubuntu und Edubuntu zu nennen.

UCS (univention.de) ist eine moderne Enterprise-Linux-Distribution mit integrierter Open-Source-Lösung für das Identity- und Infrastruktur-Management, die auch in anspruchsvollen Umgebungen eine effiziente und zentral gesteuerte Verwaltung ermöglicht - im Unternehmen und in der Cloud. Das Kernprodukt UCS wird unter anderem durch darauf aufbauende Produkte für die plattformübergreifende Verwaltung von Thin Clients, einem Open Source Desktop für professionelle Anwender, sowie eine integrierte Server- und Desktop-Virtualisierungslösung ergänzt. Die Produkte passen sich dank mitgelieferter Konnektoren, etwa zu Microsoft Active Directory, gut in vorhandene Infrastrukturen ein und erlauben eine einfache Migrationen. Eine wachsende Zahl von Softwareherstellern, darunter die Groupware-Hersteller Open-Xchange, Zarafa und Kolab Systems, ECM-Hersteller und Anbieter von VoIP-Lösungen, bieten für den Betrieb mit UCS optimierte Pakete an, die sich u.a. in das UCS-Managementsystem integrieren lassen. UCS ist damit die Open-Source-Integrationsplattform für IT-Infrastrukturbetrieb und -management.

Der GPL-lizenzierte Quellcode zu aktuellen UCS-Versionen kann von dem Univention-Downloadserver heruntergeladen werden: download.univention.de/download/ucs-cds/.

univention
be open.

Eines der besonderen Merkmale von UCS ist der Single-Point-of-Administration: Das integrierte Managementsystem (www.univention.de/produkte/ucs/ucs-managementsystem/) macht Benutzer, Rechte, Rechner, Standorte und Betriebssysteme mittels des Verzeichnisdienstes OpenLDAP zentral administrierbar.

Zur einfachen Verwaltung auch von komplexen und heterogenen IT-Infrastrukturen bietet das UCS-Managementsystem eine grafische Web-Oberfläche. Das Managementsystem einschliesslich Webinterface und vielen anderen Werkzeugen ist eine Eigenentwicklung und wird von Univention unter der GPL im Quellcode zur Verfügung gestellt.

Das Managementsystem unterstützt auch den Einsatz Linux-fremder Betriebssysteme und anderer Linux-Distributionen sowohl auf der Server- als auch Client-Seite. Mit UCS lassen sich problemlos heterogene IT-Infrastrukturen mit Windows, Linux, Unix und MacOS betreiben. So kann beispielsweise ein UCS Serversystem als Primary Domain Controller (PDC) oder Backup Domain Controller (BDC) in eine Samba-Umgebung integriert werden. Auch die Integration und Verwaltung von bestehenden Active

Directory-Umgebungen ist bis hin zur vollständigen Ablösung realisierbar. Seit UCS 3.0 ist Samba 4 integriert (www.univention.de/samba-4/).

Ein Ziel der Entwickler von UCS war es, ein Linux-Betriebssystem mit integriertem Identity- und Infrastrukturmanagement zur Verfügung zu stellen, dass "out-of-the-box", ohne manuellen Konfigurationsaufwand funktioniert. So stehen sofort nach der Installation von UCS alle relevanten Serverdienste und Funktionen zur Verwaltung von IT-Infrastrukturen zur Verfügung. Damit verfolgt Univention ein ähnliches Ziel wie die Debian-Entwickler: durch sinnvoll vorkonfigurierte Pakete ein sofort benutzbares System nach der Installation zu haben.

Darüber hinaus erweitern die UCS-Komponenten das System um zahlreiche Funktionen, wie Thin-Client-Infrastruktur, Terminal-Services, Groupware, VoIP und Virtualisierungslösungen. (www.univention.de/produkte/)

Die komplette Dokumentation zu UCS ist zu finden unter:
www.univention.de/download/dokumentation/standarddokumentation/

Die Binärpakete von UCS können in einer "free-for-personal-use"-Edition heruntergeladen werden: www.univention.de/download/

Univention Corporate Desktop (www.univention.de/produkte/ucd/) ermöglicht zusammen mit Univention Corporate Server vom Server bis zum Client ein durchgängiges linuxbasiertes Gesamtkonzept. UCD kann über das UCS-Managementsystem aufgesetzt und administriert werden. Dadurch bringt Univention Corporate Desktop folgende Features und Möglichkeiten mit:

Zentrales Identity- und Berechtigungsmanagement

Zentrale Vorgabe und Verwaltung von Benutzerdesktops

Software-Verteilung: Vollautomatischer Roll-Out, Update- und

Patchmanagement

UCD deckt alle für den Desktop-Arbeitsplatz nötigen Anwendungen ab: Büroprogramme (OpenOffice), Applikationen zur Kommunikation (zum Beispiel Kontact, Firefox, Konqueror) und Multimedia-Anwendungen, wie Gimp oder K3B zum Brennen von CDs und DVDs. Als grafische Benutzeroberfläche wird KDE verwendet. Durch Applikationen wie rdesktop und Wine ermöglicht Univention Corporate Desktop die Anbindung von Fremdsoftware, welche nicht für Linux verfügbar ist.

Das Debian Logo ist in zwei Versionen, einer offiziellen und einer öffentlichen, verfügbar. Die öffentliche Version kann immer verwendet werden, wenn auf das Debian-Projekt verwiesen werden soll. Ein Link auf die Webseite des Projekts sollte dabei selbstverständlich sein.

Das offizielle Logo darf nur im Zusammenhang mit Produkten eingesetzt werden, die mittels einer dokumentierten Prozedur hergestellt wurden, beispielsweise die offiziellen Debian CD-ROMs, oder wenn eine besondere Genehmigung vom Projekt vorliegt.

Beide Logos finden sich in den verschiedensten Formaten auf der Seite http://www.debian.org/logos/.

Weiterhin ist das Debian-Logo auch in ASCII-Formaten im Netz verfügbar. Diese Version kann beispielsweise als Ersatz für die Datei /etc/issue dienen.

Hier eine einfache Version:

```
       _,met$$$$$gg.          ,g$$$$$$$$$$$$$$$P.       ,g$$P""
"""Y$$.".      ,$$P'              `$$$.   ',$$P      ,ggs.     `$$b:   `d$$'     ,$P"'
.    $$$    $$P     d$'     ,    $$P    $$:      $$.   -    ,d$$'         $$;      Y$b._
_,d$P'          _,          _,      ,'.    Y$$.    `.`"Y$$$$P"'          `$$'       `$$'
`.  ,'    `$$b      "-.__              $$         $$       `'     `Y$$b
$$          $$         _,          _     `Y$$.                 ,d$$$g$$   ,d$$$b.
$$,d$$$b.`$$' g$$$$$b.`$$,d$$b.        `$$b.             ,$P'  `$$ ,$P' `Y$.
$$$'  `$$ $$   "'    `$$ $$$' `$$         `Y$$b.          $$'    $$ $$'    `$$ $$'
$$ $$   ,ggggg$$ $$'   $$           `"Y$b._       $$     $$ $$ggggg$$ $$
$$ $$ ,$P"    $$ $$    $$             `""""       $$     ,$$ $$.       $$     ,$P $$
$$'   ,$$ $$    $$                      `$g. ,$$$ `$$._ _., $$ _,g$P' $$ `$b.
,$$$ $$    $$                        `Y$$P'$$. `Y$$$$P',$$$$P"'   ,$$.
`Y$$P'$$.$$.  ,$$.
```

1.2 Was ist Debian GNU?

Kapitel 1. Linux, Debian, freie Software? 1.3 Freie Software / Open Source

1.3 Freie Software / Open Source

1.2 Was ist Debian GNU? 1.4 Copyleft und Copyright

1.3.1 Open Source Initiative (OSI)

Debian GNU/Linux ist freie Software, auch Open Source Software genannt - aber warum muss man dann für Debian GNU/Linux etwas bezahlen?

Freie Software unterscheidet sich von Freibier in einem ganz entscheidenden Punkt: im Preis. Sicher kann es auch schon mal passieren, dass man eine Debian GNU/Linux-CD kostenlos bekommt, aber der Begriff freie Software beschreibt nicht das Verschenken von CDs. Der Preis, den Sie für eine Debian GNU/Linux-CD bezahlen, setzt sich aus den Kosten für die Zusammenstellung der CD sowie den Herstellungsprozess, also dem Pressen der CDs und dem Druck der Dokumentation und des Covers, sowie den Kosten, die dem Hersteller der CDs für den Download der Software entstehen, und schließlich auch den Kosten für den Vertrieb der Pakete zusammen. Sie können die Debian GNU/Linux-Software auf so vielen Rechnern installieren, wie Sie möchten; es fallen keinerlei Lizenzkosten pro Installation an. Sie können die Debian GNU/Linux-CDs beliebig an Freunde und Bekannte verleihen oder auch Kopien der CDs weitergeben - ohne Lizenzkosten. Sie können Debian GNU/Linux auf Ihrem Rechner zu Hause oder in der Firma installieren; Sie zahlen nichts extra dafür.

Natürlich können Sie Debian GNU/Linux aus dem Internet von einem der vielen Debian-GNU/Linux-FTP-Server kopieren und installieren. Bei den momentanen Kosten für Internetzugänge ist dies aber sicherlich teurer als eine CD-ROM.

Zu jedem Debian-GNU/Linux-Paket sind auf den FTP-Servern im Verzeichnis source/ die Programme im Quellcode (source) verfügbar. Diese stehen jedermann zur Verfügung, und jeder kann an der weiteren Entwicklung teilnehmen. Die Möglichkeit, Änderungen an den Quellen vorzunehmen und diese Änderungen an die Hauptentwickler zu geben, damit diese die Änderungen in das Programm aufnehmen, machen die hohe Effektivität der Open-Source-Software aus. Achten Sie darauf, wenn Sie an der Entwicklung Freier Software teilnehmen wollen: Nicht jedes Programm, das sich als „Open Source“ bezeichnet, erfüllt dieses Kriterium. Es geht nicht allein darum, Einblick in den Quellcode zu gewähren, wichtig ist auch, dass es jedermann freisteht, Änderungen vorzunehmen und dass auch diese wieder für jedermann zugänglich gemacht werden, indem die Änderungen in den Quellcode einfließen.

Ein dritter entscheidender Punkt ist, dass alle freie Software auch für alle Zeit freie Software bleiben muss. Dies schützt freie Software-Projekte davor, dass irgendjemand Freie Software zu seinem Eigentum erklärt und „eigene“ Produkte daraus erstellt, die wiederum einer Lizenz unterliegen, die diese Freiheiten nicht weitergibt.

Informationen zu Freier Software

Bei der Bundeszentrale für Politische Bildung ist das Buch „Freie Software“ von Volker Grassmuck sowohl als gedruckte Version als auch zum Download verfügbar. Dieses Buch bietet einen sehr umfassenden Überblick über die Geschichte Freier Software und behandelt darüber hinaus auch viele weitere Aspekte, die mit freiem Wissen zusammenhängen.

Auf den Seiten der FSF Europe wird erläutert, warum dem Begriff „Freie Software“ gegenüber dem Begriff „Open Source“ der Vorzug zu geben ist.

Freie Software zeichnet sich weiterhin dadurch aus, dass die Verwendung dieser Software wirklich frei ist. Sie können diese Software einsetzen, wofür Sie wollen; zu Hause, im Büro, privat, kommerziell, zum Spielen, zur Produktion von Filmen, zur Produktion von Waffen. Dieser letzte Punkt mag abschreckend klingen, aber niemand soll in der Verwendung Freier Software reglementiert werden.

Es gibt viele verschiedene Lizenzen für „freie Software“: Einige sind mehr, andere weniger „frei“. Wenn Sie sich für den Einsatz eines speziellen Programms in einem speziellen (zum Beispiel im kommerziellen Einsatz) Bereich interessieren, lesen Sie die Lizenzbedingungen der jeweiligen Software. Diese Lizenzen finden Sie unter /usr/doc/ beziehungsweise /usr/share/doc/.

Richard Stallman fasste dies in den frühen 80er Jahren zusammen und formulierte vier „Freiheiten“:

0. Freiheit:

Die Freiheit, ein Programm für jeden Zweck einsetzen zu dürfen.

1. Freiheit:

Die Freiheit, untersuchen zu dürfen, wie ein Programm funktioniert, und es den eigenen Bedürfnissen anzupassen.

2. Freiheit:

Die Freiheit, Kopien für andere machen zu dürfen.

3. Freiheit:

Die Freiheit, das Programm verbessern zu dürfen und diese Verbesserungen zum allgemeinen Wohl zugänglich zu machen.

1.3.1.1 Geschichte der OSI

1.3.1.2 Die Definition quelloffener Software („Open Source Software“)

Die OSI ist eine Non-Profit-Organisation mit dem Zweck der Organisation und Verbreitung der „Open Source“-Definition für frei verfügbare Software zum Nutzen der Gemeinschaft.

Informationen über die Struktur und Organisation der OSI sowie die verschiedenen Lizenzen für Open-Source-Software finden sich auf der Webseite der Open Source Initiative.

Die von der OSI veröffentlichte „Definition quelloffener Software“ geht auf die von Bruce Perens begonnenen ersten Versionen der „Debian Free Software Guidelines“ (DFSG) zurück. Die DFSG wurden im Juni 1997, nach einer monatelangen E-Mail-Konferenz, endgültig fertig gestellt. Bruce entfernte die Debian-spezifischen Passagen im Text und schuf so die erste Version der „Open Source-Definition“.

Die Vorgeschichte der Open Source Initiative beruht auf der gesamten Entwicklungsgeschichte von Unix, der im Internet entwickelten freien Software und der „Hacker-Kultur“.

Das „Open Source“-Label selbst entstand im Rahmen eines strategischen Treffens am 3. Februar 1998 in Palo Alto, California. Dabei waren unter anderem anwesend: Todd Anderson, Chris Peterson (Foresight Institute), John „maddog“ Hall und Larry Augustin (beide Linux International), Sam Ockman (Silicon Valley Linux User's Group) und Eric S. Raymond.

Dieses Treffen, welchem viele Diskussionen auf Mailinglisten vorangingen, war eine Reaktion auf die Ankündigung der Firma Netscape, die Sourcen ihrers Browsers freizugeben. Dieses Projekt wurde später unter dem Namen Mozilla bekannt. Eric S.

Raymond wurde von der Firma Netscape im Vorfeld der Ankündigung gebeten, bei der Freigabe des Quellcodes und den damit verbundenen Formalitäten zu helfen.

Die „Open Source“-Definition ist von den Debian Free Software Guidelines abgeleitet. Bruce Perens verfasste die erste Version, die durch den regen E-Mail-Kontakt mit verschiedenen Debian Entwicklern im Juni 1997 verfeinert wurde.

Die Open Source Initiative ist mittlerweile als „California public benefit“ (eine nicht-kommerzielle) Vereinigung anerkannt. Die Adresse für Spenden lautet:

Law Offices of Lawrence E. Rosen 702 Marshall St. Ste. 301 Redwood City, CA 94063

Version 1.9

Einführung

„Quelloffen“ („Open Source“) bedeutet nicht nur freien Zugang zum Quellcode. Bei quelloffener Software müssen die Lizenzbestimmungen in Bezug auf die Weitergabe der Software folgenden Kriterien entsprechen:

1. Freie Weitergabe

Die Lizenz darf niemanden in seinem Recht einschränken, die Software als Teil eines Software-Paketes, das Programme unterschiedlichen Ursprungs enthält, zu verschenken oder zu verkaufen. Die Lizenz darf für den Fall eines solchen Verkaufs keine Lizenz- oder sonstigen Gebühren festschreiben.

2. Quellcode

Das Programm muss den Quellcode beinhalten. Die Weitergabe muss sowohl für den Quellcode als auch für die kompilierte Form zulässig sein. Wenn das Programm in irgendeiner Form ohne Quellcode weitergegeben wird, so muss es eine allgemein bekannte Möglichkeit geben, den Quellcode zum Selbstkostenpreis zu bekommen, vorzugsweise als gebührenfreien Download aus dem Internet. Der Quellcode soll die Form eines Programms sein, die ein Programmierer vorzugsweise bearbeitet. Absichtlich unverständlich geschriebener Quellcode ist daher nicht zulässig. Zwischenformen des Codes, so wie sie etwa ein Präprozessor oder ein Konverter („Translator“) erzeugt, sind unzulässig.

3. Abgeleitete Software

Die Lizenz muss Veränderungen und Derivate zulassen. Außerdem muss sie es zulassen, dass die solcherart entstandenen Programme unter denselben Lizenzbestimmungen weitervertrieben werden können wie die Ausgangssoftware.

4. Unversehrtheit des Quellcodes des Autors

Die Lizenz darf die Möglichkeit, den Quellcode in veränderter Form weiterzugeben, nur dann einschränken, wenn sie vorsieht, dass zusammen mit dem Quellcode so genannte „Patch files“ weitergegeben werden dürfen, die den Programmcode bei der Kompilierung verändern. Die Lizenz muss die Weitergabe von Software, die aus verändertem Quellcode entstanden ist, ausdrücklich erlauben. Die Lizenz kann verlangen, dass die abgeleiteten Programme einen anderen Namen oder eine andere Versionsnummer als die Ausgangssoftware tragen.

5. Keine Diskriminierung von Personen oder Gruppen

Die Lizenz darf niemanden benachteiligen.

6. Keine Einschränkungen bezüglich des Einsatzfeldes

Die Lizenz darf niemanden daran hindern, das Programm in einem bestimmten Bereich einzusetzen. Beispielsweise darf sie den Einsatz des Programms in einem Geschäft oder in der Genforschung nicht ausschließen.

7. Weitergabe der Lizenz

Die Rechte an einem Programm müssen auf alle Personen übergehen, die diese Software erhalten, ohne dass für diese die Notwendigkeit bestünde, eine eigene, zusätzliche Lizenz zu erwerben.

8. Die Lizenz darf nicht auf ein bestimmtes Produktpaket beschränkt sein

Die Rechte an dem Programm dürfen nicht davon abhängig sein, ob das Programm Teil eines bestimmten Software-Paketes ist. Wenn das Programm aus dem Paket herausgenommen und im Rahmen der zu diesem Programm gehörenden Lizenz benutzt oder weitergegeben wird, so sollen alle Personen, die dieses Programm dann erhalten, alle Rechte daran haben, die auch in Verbindung mit dem ursprünglichen Software-Paket gewährt wurden.

9. Die Lizenz darf die Weitergabe zusammen mit anderer Software nicht einschränken

Die Lizenz darf keine Einschränkungen enthalten bezüglich anderer Software, die zusammen mit der lizenzierten Software weitergegeben wird. So darf die Lizenz zum Beispiel nicht verlangen, dass alle anderen Programme, die auf dem gleichen Medium weitergegeben werden, auch quelloffen sein müssen.

1.3 Freie Software / Open Source

1.2 Was ist Debian GNU? 1.4 Copyleft und Copyright

1.4 Copyleft und Copyright

1.3 Freie Software / Open Source 1.5 SPI - Software in the Public Interest

Die einfachste Art, ein Programm frei zu veröffentlichen, besteht darin, es als „Public Domain“ freizugeben, ohne Copyright. Das erlaubt es allen Personen, das ursprüngliche Programm und verbesserte Versionen zu verteilen, wenn dies gewollt ist. Aber es erlaubt auch unkooperativen Leuten, das Programm in proprietäre Software umzuwandeln. Es können kleinere oder größere Änderungen vorgenommen werden, und das Resultat kann als eigenes Produkt vertrieben werden, ohne dass die Weiterentwicklungen der Allgemeinheit zugute kommen müssen. Proprietäre Software erlaubt meistens keine Weitergabe des Quellcodes, so dass auch keine weiteren Verbesserungen von anderen Personen vorgenommen werden können.

Das GNU-Projekt hat zum Ziel, allen Benutzern die Freiheit zu geben, GNU-Software weiterzuverteilen und zu verändern. Daher steht GNU-Software nicht in der „Public Domain“, sondern unter einem so genannten „Copyleft“. Das Copyleft bedeutet, dass alle, die die Software (mit oder ohne Änderungen) weiter verteilen, auch die Freiheit zum Weitergeben und Verändern mitgeben müssen. Das Copyleft garantiert die notwendige Freiheit für alle Benutzer.

Um ein Programm unter das Copyleft zu stellen, wird es zuerst unter ein Copyright gestellt; danach werden als Rechtsmittel Vertriebsbestimmungen hinzugefügt, die es allen erlauben, den Quellcode des Programms oder jedes davon abgeleiteten Programms zu verwenden, zu ändern und weiterzuverteilen. Dies ist aber nur gestattet, wenn die Vertriebsbestimmungen unverändert bleiben und auf das verbesserte Werk angewendet werden. So werden der Code und die gewährten Freiheiten rechtlich untrennbar, das Copyleft hat einen virulenten Charakter.

Entwickler proprietärer Software verwenden das Copyright, um den Benutzern ihre Freiheit zu nehmen; Entwickler Freier Software verwenden es, um anderen Freiheiten zu garantieren. Deshalb wurde der Name umgedreht und aus dem „Copyright“ das „Copyleft“ gemacht.

Das Copyleft ist ein allgemeines Konzept; es kann auf viele Weisen konkret ausgestaltet werden. Für das GNU-Projekt sind die konkreten Vertriebsbestimmungen in der GNU General Public License geregelt. Häufig wird die GNU General Public License kurz als GNU GPL bezeichnet.

Eine alternative Form des Copylefts, die GNU Lesser General Public License (LGPL), wird für einige (aber nicht alle) GNU-Bibliotheken verwendet. Diese Lizenz wurde früher Library GPL genannt. Der Name wurde jedoch geändert, weil der alte Name dazu angeregt hat, die Lizenz häufiger zu verwenden als wünschenswert ist.

Die GNU Free Documentation License (FDL oder GFDL) ist eine Form des Copylefts, die für Handbücher, Lehrbücher und andere Dokumente gedacht ist. Sie sichert allen die Freiheit zu, diese Dokumente zu kopieren und weiterzuverteilen, mit oder ohne Änderungen, entweder kommerziell oder nicht-kommerziell.

Die GNU GPL wurde so konzipiert, dass Sie sie leicht auf Ihre eigenen Programme anwenden können, wenn Sie der Copyright-Inhaber sind. Sie müssen nicht die GNU GPL ändern, um dies zu tun, sondern lediglich Ihrem Programm Hinweise hinzufügen, die in der richtigen Form auf die GNU GPL verweisen.

Die Verwendung der gleichen Vertriebsbestimmungen für viele verschiedene Programme macht es einfach, Quellcode zwischen verschiedenen Programmen auszutauschen. Da sie alle die gleichen Vertriebsbestimmungen haben, muss man sich keine Gedanken darüber machen, ob die Bestimmungen zueinander kompatibel sind. Die Lesser GPL enthält eine Bestimmung, die es erlaubt, stattdessen die GPL zu verwenden, so dass der Quellcode in jedes unter der GPL stehende Programm übernommen werden kann.

1.4 Copyleft und Copyright

1.3 Freie Software / Open Source 1.5 SPI - Software in the Public Interest

1.5 SPI - Software in the Public Interest

1.4 Copyleft und Copyright 1.6 Standardisierungen

SPI ist eine Non-profit-Organisation, die gegründet wurde, um Projekten zu helfen, die Software für die Allgemeinheit entwickeln. Sie ermutigt Programmierer, die GNU General Public License (GPL) oder eine andere Lizenz zu benutzen, die eine freie Weiterverbreitung und den freien Gebrauch der Software erlaubt. Hardware-Entwickler ermutigt SPI, die Dokumentation zu ihrer Arbeit zu veröffentlichen, damit Treiber für ihre Produkte geschrieben werden können.

SPI wurde am 16. Juni 1997 als Non-Profit-Organisation im Staat New York der USA gegründet. Seitdem ist sie eine Schirmorganisation für verschiedene Projekte der Gemeinschaft. Die Statuten und das Gründungszertifikat definieren das Ziel und die Arbeitsweise von SPI. SPI hat einen Vorstand, der aus vier Mitgliedern besteht: einem Präsidenten, einem Vize-Präsidenten, einem Beisitzer und einem Kassierer.

SPI unterstützt momentan folgende Projekte: Berlin, Debian, GNOME, LSB, Open Source und Open Hardware.

Wenn Sie etwas an SPI spenden möchten oder nicht auf elektronischem Wege kommunizieren wollen, erreichen Sie SPI unter der Adresse:

Software in the Public Interest, Inc. PO BOX 273 Tracy, CA 95378-0273 USA

Weitere Informationen zu SPI finden Sie im Internet.

1.5 SPI - Software in the Public Interest

1.4 Copyleft und Copyright 1.6 Standardisierungen

1.6 Standardisierungen

1.5 SPI - Software in the Public Interest 1.7 Wie und wo bekomme ich Debian GNU/Linux?

Debian GNU/Linux hält sich an verschiedene Standards, und DebianEntwickler arbeiten in den verschiedensten Gremien mit, um die Entwicklung solcher Standards voranzutreiben.

1.6.1 Linux Standard Base

Linux Standard Base (http://www.linuxbase.org) ist eine herstellerübergreifende Initiative, um eine weitgehende Kompatibilität zwischen verschiedenen Linux-Distributionen sicherzustellen. Die Free Standard Group (http://www.freestandards.org) hat den Standard LSB 2.0 Ende August 2004 veröffentlicht. Ziel der LSB ist es, die verschiedenen Linux-Distributionen hinsichtlich Dateisystemstruktur und grundsätzlich notwendiger Bibliotheken zu vereinheitlichen, um eine zu starke Zersplitterung zu vermeiden und Software-Entwicklern eine einheitliche Basis zu bieten.

Für die nächste Version des LSB ist ein wichtiges Ziel, das Zusammenspiel zwischen RPM-basierten Distributionen (RedHat/Fedora, Novell/SuSE...) und Debian zu verbessern.

1.6.2 Filesystem Hierarchy Standard

Der Filesystem Hierarchy Standard (http://www.pathname.com/fhs/) beschreibt den Aufbau eines Unix-Dateisystems und dient Entwicklern (einer Linux Distribution) als Richtlinie für den Aufbau einer Verzeichnisstruktur.

1.6 Standardisierungen

1.5 SPI - Software in the Public Interest 1.7 Wie und wo bekomme ich Debian GNU/Linux?

1.7 Wie und wo bekomme ich Debian GNU/Linux?

1.6 Standardisierungen 1.8 Informationen im Netz

Die Bezugsquellen für Debian GNU/Linux sind recht vielfältig, auch wenn man nicht an jeder Ecke über diese Distribution stolpert, wie bei anderen.

Die aktuellste Bezugsquelle ist natürlich der Debian-FTP-Server sowie seine weltweiten Spiegel. Der Debian-FTP-Server ist unter

ftp://ftp.debian.org

zu erreichen, allerdings sollten Sie einen Spiegel in Ihrer Nähe benutzen; Sie erreichen mit ziemlicher Sicherheit hier bessere Übertragungsraten.

Eine aktuelle Liste aller Spiegel des Debian-FTP-Servers finden Sie unter:

http://www.debian.org/CD/http-ftp/.

Das Debian Team unterscheidet zwei Klassen von FTP-Servern: so genannte „primary“ und „secondary mirror sites“. Ein Primary-Server hat eine sehr gute Anbindung ans Netz, ist 24 Stunden am Tag verfügbar und hat einen leicht zu merkenden Namen in der Form ftp.<country>.debian.org. Natürlich werden diese Spiegel ständig automatisch mit dem Debian Master-Server abgeglichen.

Ein Secondary-Server ist meist schlechter angebunden und enthält oft keine komplette Kopie des Debian Servers.

Versuchen Sie immer, den Ihnen am nächsten gelegenen Server zu benutzen; hier haben Sie in der Regel die besten Übertragungsraten. Ein geografisch nahe gelegener Server muss dabei nicht zwingend der schnellste sein; das Programm netselect hilft Ihnen, den optimalen Server zu ermitteln.

Für den deutschsprachigen Raum stehen die beiden Server ftp://ftp.de.debian.org (auch als ftp://ftp.debian.de erreichbar) sowie ftp://ftp2.de.debian.org zur Verfügung. Beide führen auch den Bereich Non-US, der Programme enthält, die aufgrund von Exportbeschränkungen oder Software-Patenten nicht in den USA verteilt werden dürfen.

Die Benutzung eines solchen FTP-Servers ist kostenfrei. Bedenken Sie aber, dass der Download von großen Datenmengen trotzdem Kosten hervorruft. Sie finden diese spätestens auf Ihrer nächsten Rechnung für Ihren Telefonanschluss bzw. in der Rechnung

Ihres Internetproviders. Weitere Kosten können bei Ihrem Internetprovider entstehen, wenn dort nach Onlinezeit oder Übertragungsvolumen abgerechnet wird.

Einfacher und meist auch kostengünstiger ist es, sich Debian GNU/Linux auf einer CD-ROM oder DVD zu beschaffen. Auch für die Beschaffung eines solchen Mediums finden Sie Informationen auf den Debian-Webseiten.

Auch ist es möglich, die ISO-Dateien der CDs oder DVDs komplett aus dem Internet herunterzuladen. Hierzu sollte möglichst das Programm Jigdo oder BitTorrent verwendet werden. Informationen zum Herunterladen der Dateien und zu Jigdo und BitTorrent finden sich unter http://www.debian.org/distrib/cd und im Abschnitt zu Jigdo und BitTorrent .

Bereits seit mehreren Jahren vertreibt die Fachbuchhandlung Lehmanns die Debian GNU/Linux-Distribution auf CD-ROM. Sie können die CDs auch unter http://www.lehmanns.de online bestellen. Hier sind die aktuellen („stable“) Debian GNU/Linux-CD-ROMs und DVDs für i386 verfügbar.

Die Buchhandlung Lehmanns war auch Initiator und lange Jahre Betreiber einer deutschsprachigen Mailingliste zu Debian. Diese Liste wurde später (mit sehr vielen Benutzern) vom Debian-Projekt übernommen und wird heute als debian-user-de weiterhin genutzt. Die Liste ist deutschsprachig und kann über die Debian Webseiten abonniert werden.

Natürlich können Sie sich auch von einem anderen Debian Freund eine CD oder DVD kopieren, dies ist sicher die preiswerteste Variante, um an Debian zu kommen.

Von den vielen Linux-Usergroups werden häufig so genannte Installationspartys durchgeführt. Auf diesen besteht die Möglichkeit, unter fachlicher Anleitung neben anderen Distributionen auch Debian GNU/Linux auf dem mitgebrachten Rechner zu installieren. Eine Linux-Usergroup in Ihrer Nähe finden Sie über die Liste der deutschsprachigen Linux-Usergroups des Linux Magazins.

Grundsätzlich kann der Besuch bei einer solchen Linux-Usergroup nur jedem Interessierten nahe gelegt werden. Der dort mögliche Erfahrungsaustausch ist sehr hilfreich, und die meisten auftretenden Probleme bei der Installation und beim Betrieb von Debian GNU/Linux lassen sich dort schnell lösen.

1.7.4.3 Französischsprachige Bücher

Mittlerweile gibt es eine ganze Reihe Bücher zu Debian GNU/Linux. Erfreulicherweise sind auch einige deutschsprachige Werke darunter. Leider sind aber fast alle vergriffen.

Titel: Debian GNU/Linux

Autor: Ganten, Peter H.

Untertitel: Grundlagen, Installation, Administration und Anwendung

Datum/Seiten: 2000, 792

Verlag: Springer

Titel: Debian GNU/Linux

Autoren: Peter H. Ganten, Alex Wulf

Beschreibung: 2. überarbeitete Auflage

Datum/Seiten: 2004, 946

Verlag: Springer

Titel: Debian GNU/Linux Anwenderhandbuch

Autor: Ronneburg, Frank

Datum/Seiten: 2001, 600

Verlag: Addison-Wesley, München

Titel: Debian GNU/Linux Anwenderhandbuch

Autor: Ronneburg, Frank

Datum/Seiten: 2001, 600

Verlag: Lehmanns Sonderausgabe (Originalverlag Addison-Wesley)

Titel: Debian GNU/Linux Anwenderhandbuch, mit 6 CD-ROMs Debian GNU/Linux 2.2

Autor: Ronneburg, Frank

Datum/Seiten: 2001, 600

Verlag: Lehmanns Sonderausgabe (Originalverlag Addison-Wesley)

Titel: Debian GNU/Linux Anwenderhandbuch - Linux für Einsteiger, Umsteiger und Fortgeschrittene

Autor: Ronneburg, Frank

Datum/Seiten: 2003, 710

Verlag: Lehmanns Media

Titel: Debian GNU/Linux Anwenderhandbuch - Linux für Einsteiger, Umsteiger und Fortgeschrittene

Autor: Ronneburg, Frank

Datum/Seiten: 2004, 600

Verlag: Addison-Wesley, München

Titel: Debian GNU/Linux Guide

Autoren: John Goerzen, Ossama Othman

Beschreibung: Übersetzung des offiziellen "Debian GNU/Linux Guide" von Bramer.

Seiten: 362

Titel: Debian GNU/Linux

Autor: John Goerzen and Ossama Othman

Untertitel: Guide to Installation and Usage

Datum/Seiten: 1999, 200

Verlag: New Riders

Titel: Learning Debian GNU/Linux

Autor: McCarty, B.

Untertitel: A Guide to Debian GNU/Linux for New Users

Datum: 1999

Verlag: O'Reilly

Titel: Debian GNU/Linux for dummies

Autor: Bellomo, Michael

Untertitel: -

Datum/Seiten: 2000, 384

Verlag: IDG Books

Titel: Installing Debian GNU/Linux

Autor: Down, Thomas

Untertitel: -

Datum/Seiten: 1999, 250

Verlag: Sams Publishing

Titel: Debian GNU/Linux 2.1 Unleashed mit CD-ROM

Autor: Camou, Mario; Goerzen, John; VanCouwenberghe, Aaron

Untertitel: -

Datum: 2000

Verlag: Sams Publishing

Titel: Debian GNU/Linux Bible

Autor: Steve Hunger

Untertitel: -

Seiten: 696

Verlag: Hungry Minds, Inc

Titel: The Debian System

Autor: Martin F. Krafft

Untertitel: Concepts and Techniques

Datum/Seiten: 2005, 608

Verlag: open source Press

Titel: Debian GNU/Linux - Installation, Administration, Exploitation

Autor: Ronneburg, Frank

Datum/Seiten: 2005, 650

Verlag: Campus Press / Pearson Frankreich

Titel: Debian GNU/Linux (Cahiers de l'Admin GNU/Linux (Cahiers de l'Admin))

Autor: Raphaël Hertzog , Christophe Le Bars , Roland Mas

Datum/Seiten: 2005, 310

Verlag: Eyrolles

1.7 Wie und wo bekomme ich Debian GNU/Linux?

1.6 Standardisierungen 1.8 Informationen im Netz

1.8 Informationen im Netz

1.7 Wie und wo bekomme ich Debian GNU/Linux? Kapitel 2. Installation von Debian GNU/Linux

An dieser Stelle folgen einige Hinweise auf weitere Informationsquellen im Internet, die zu Rate gezogen werden können, falls Sie nicht die gewünschten Informationen in diesem Buch finden (was aber kaum vorstellbar ist :-)).

Auf den Seiten des Debian-Projekts finden Sie Informationen zur Unterstützung bei Debian Problemen unter http://www.debian.org/support.

Die deutschsprachige Mailingliste zu Debian wurde ursprünglich von Lehmanns Fachbuchhandlung initiiert. Diese Liste wurde später (mit sehr vielen Benutzern) vom Debian-Projekt übernommen und wird heute als debian-user-de weiterhin genutzt. Die Liste ist deutschsprachig und kann über die Debian Webseiten abonniert werden.

Aus dieser Mailingliste entstand eine deutschsprachige FAQ zu Debian unter http://channel.debian.de/faq/.

Eine Übersicht über die Mailinglisten des Debian-Projekts findet sich auf den Webseiten unter: http://www.debian.org/MailingLists/.

Mit Ausnahme einiger spezieller Listen, die sich mit den landessprachlichen Besonderheiten befassen, ist die Sprache auf diesen Listen englisch.

Debian-Projekt

http://www.debian.org

Die Webseite des Debian-Projekts.

Debian Weekly News

Neben der eigentlichen Debian Webseite sind die wöchentlich erscheinenden „Debian Weekly News“ eine weitere interessante Informationsquelle. In diesem Newsletter wird über wichtige Änderungen in der Debian Distribution, über Diskussionen auf den Mailinglisten und über anderen Klatsch und Tratsch berichtet. Zu finden sind die „Debian Weekly News“ unter: http://www.debian.org/News/weekly/.

Bereits vor der Veröffentlichung der neuen Ausgabe sind die bis dahin erstellen Artikel unter:

http://www.infodrom.org/~joey/Writing/DWN/

zu finden.

Deutschsprachiges Debian Forum

http://www.debianforum.de/

Debian-Administration.org - System Administration Tips and Resources

Viele interessante Artikel und Tipps zur Administration von Debian Systemen.
http://www.debian-administration.org/

debianHelp

http://www.debianhelp.org/ ist eine Seite in englischer Sprache, auf der sich viele Anleitungen und Lösungshinweise rund um Debian finden.

Debian Wiki

http://wiki.debian.org/ Wikis sind im World Wide Web verfügbare Seitensammlungen, die von den Benutzern nicht nur gelesen, sondern auch online geändert werden können. Der Name wurde vom hawaiianischen „wiki wiki“ abgeleitet, was so viel wie „schnell“ bedeutet. Innerhalb eines Wikis sind die einzelnen Seiten, die Artikel, durch Querverweise (Links) verbunden. Sie lassen sich jedoch praktisch sofort am Bildschirm ändern. Dazu existiert in der Regel eine Bearbeitungsfunktion, die ein Eingabefenster öffnet, in dem der Text des Artikels bearbeitet werden kann.

Debian Wiki beschäftigt sich dabei ausschließlich mit Themen rund um die Distribution.

Inoffizielle Debian Pakete

Auf der Seite http://www.apt-get.org finden Sie Hinweise zu nicht-offiziellen Debian Paketquellen im Netz. Betreuer von solchen inoffiziellen Debian Paketen können ihre Seiten dort selbst anmelden. Hier finden sich beispielsweise Debian Pakete zu Software-Paketen, die zum Zeitpunkt der Veröffentlichung des aktuellen Debian Release noch nicht verfügbar waren und die auch nachträglich nicht in diese Version von Debian aufgenommen wurden.

Debian GNU/Linux Referenz-Karte

„The 101 most important things when using Debian GNU/Linux“. Von Wolfgang Borgert zusammengestellte und in viele Sprachen übersetzte Referenz-Karte mit den wichtigsten Befehlen eines Debian Systems.

Debian-news

Neuigkeiten über Debian GNU/Linux und auf Debian basierende freie Distributionen. debian-news.net/

Debian Tutorials

Verschiedenste Tipps und Anleitungen rund um Debian. debian-tutorials.com/

Debian Konferenzen

http://www.debconf.org Debian Konferenzen und Veranstaltungen. DebConf ist die Debian Entwickler-Konferenz. Neben technischen, sozialen und politischen Gesprächen bietet DebConf eine Chance für Entwickler, Autoren und andere interessierte Menschen sich auch im echten Leben kennenzulernen. Sie findet seit 2000 jährlich verschiedenen Orten wie in Kanada, Finnland und Mexiko statt. Von vielen Vorträgen gibt es Aufzeichnungen unter: video.debian.net..

DebConf 2000

Informationen zur DebConf0 - 2000 in Bordeaux (Frankreich)

DebConf 2001

Informationen zur DebConf1 - 2001 in Toronto (Kanada)

DebConf 2002

Informationen zur DebConf2 - 2002 in Toronto (Canada)

DebConf 2003

Informationen zur DebConf3 - 2003 in Oslo (Norwegen)

DebConf 2004

Informationen zur DebConf4 - 2004 in Porto Alegre (Brasilien)

DebConf 2005

Informationen zur DebConf5 - 2005 in Helsinki (Finnland)

DebConf 2006

Informationen zur DebConf6 - 2006 in Oaxtepec (Mexico)

DebConf 2007

Informationen zur DebConf7 - 2007 in Edinburgh (Schottland)

DebConf 2008

Informationen zur DebConf8 - 2008 in Mar del Plata (Argentinien)

DebConf 2009

Informationen zur DebConf9 - 2009 in Cáceres, Extremadura (Spanien)

DebConf 2010

Informationen zur DebConf10 - 2010 in New York City (USA)

DebConf 2011

Informationen zur DebConf11 - 2011 in Banja Luka (Bosnien und Herzegowina)

DebConf 2012

Informationen zur DebConf12 - 2012 in Managua (Nicaragua)

DebConf 2013

Informationen zur DebConf13 - 2013 in Vaumarcus (Schweiz)

Planet Debian

http://planet.debian.net Planet Debian (Debian developers journal aggregation site)

DistroWatch

http://www.distrowatch.com/debian.php Überblick über alle Debian Versionen und wichtige dort enthaltene Pakete.

Apple PowerMac

Zu Linux oder auch Debian auf Apple PowerMacs gibt es eine Reihe von Webseiten im Netz. Zunächst ist die Seite http://penguinppc.org/ mit den neuesten Kerneln und Patches interessant.

An einer Installationsanleitung sowohl für den Woddy- als auch für den neuen Sarge-Installer wird unter http://linuxwiki.de/Debian/InstallationPPC gearbeitet.

DotDeb

Zusammenstellung von aktuellen „LAMP"-Paketen (Apache, PHP und MySQL) http://www.dotdeb.org/.

IRC (Internet Relay Chat) ist eine Möglichkeit, sich in Echtzeit mit anderen Personen auf der gesamten Welt zu unterhalten. Ein IRC-Kanal speziell für Debian kann im Open-Projects-IRC-Netzwerk genutzt werden. Das „Open Projects Network" wurde von Robert (lilo) Levin gegründet, um der Free-Software-Community bei der Arbeit zu helfen. Eine

ausführliche deutschsprachige Anleitung zum IRC finden Sie unter http://duplox.wz-berlin.de/texte/rps/.

Im IRC haben sich einige Verhaltensregeln etabliert, die auch von neuen Benutzern berücksichtigt werden sollten. Eine Übersicht, die in den deutschsprachigen Linux-Channels zusammengestellt wurde, findet sich unter

http://www.linuxger.de/LinuxGER.html.

Eine ähnliche Seite speziell für den Channel #debian ist unter

http://channel.debian.de/faq/

zu erreichen.

Um sich mit dem Netzwerk zu verbinden, benötigen Sie einen IRC-Client. Einige der bekanntesten sind ircII, BitchX, tkirc, X-Chat und Zircon; diese sind auch als Debian-Pakete verfügbar. Wenn Sie den Client installiert haben, müssen Sie ihm mitteilen, sich zu dem entsprechenden Server zu verbinden. Benutzen Sie dabei irc.debian.org oder irc.eu.openprojects.net. In den meisten Clients erreichen Sie dies, indem Sie Folgendes eingeben:

/server irc.debian.org

Wenn Sie verbunden sind, kommen Sie in den Kanal #debian durch Eingabe von:

/join #debian

Neben dem Kanal #debian befinden sich noch weitere Kanäle auf dem Server irc.eu.openprojects.net, die sich mit Debian beschäftigen. Das Debian-Projekt nutzt die Infrastruktur des Open-Projects-Network für folgende Kanäle:

#debian - Hier kommunizieren Benutzer und Entwickler, dieser Channel ist ziemlich überlaufen.

#debian-boot - In diesem Kanal werden Fragen zu den Debian-Bootdisketten und dem Debian Installationssystem (debian-installer) diskutiert. Dieser Kanal ist nicht für den Support von Benutzern gedacht.

#debian-jr - Hier finden sich interessierte Mitstreiter, die am internen Projekt „Debian Junior (Jr.)“ mitarbeiten wollen. Dieses Projekt beschäftigt sich mit Debian Paketen für Kinder in allen Altersstufen.

#debian.de - der deutschsprachige Debian-Kanal.

Weitere Kanäle befassen sich mit verschiedenen Landessprachen, wie zum Beispiel Französisch (#debian-fr) und Russisch (#debian-ru). Im Kanal #debian-br finden sich brasilianische Debian Nutzer; dort werden auch Themen der Übersetzung von Debian diskutiert.

Natürlich gibt es auch im deutschsprachigen Teil des Internets Server, auf denen man Debian Benutzer findet. Allen voran steht hier das „IRCnet“, das im Sprachgebrauch einfach als IRC bezeichnet wird.

Das IRCnet besteht in Deutschland hauptsächlich aus IRC-Servern, die auf Universitätsrechnern laufen. Leider nimmt die Zahl drastisch ab, seit viele dieser Server missbraucht wurden. Einige Server sind hier aufgelistet. Der Server irc.netsurf.de ist nicht über das deutsche Forschungsnetz angeschlossen und sollte von Leuten aus kommerziellen Netzwerken benutzt werden.

irc.fu-berlin.de

irc.uni-paderborn.de

irc.rz.uni-karlsruhe.de

irc.belwue.de

irc.netsurf.de

Im IRCnet finden sich neben dem Debian Kanal #Debian.DE noch die Kanäle #linux.de und #LinuxGER, die sich mit Themen rund um Linux befassen.

Debian ist freie Software und bietet ausschließlich Support auf freiwilliger Basis über das Internet (per WWW, IRC oder Mailinglisten) an. Das Projekt kann keine Mitarbeiter beschäftigen, die eine Telefon-Hotline besetzen oder Vor-Ort-Service beim Kunden leisten.

Natürlich gibt es aber auch genau in diesem Bereich Bedarf. Deshalb bietet das Debian-Projekt auf der Webseite

http://www.debian.org/consultants/

eine Aufstellung von Beratern an, die solche Dienstleitungen anbieten. Die Namen sind nach Ländern sortiert, innerhalb eines Landes aber einfach in der Reihenfolge der Anmeldung geordnet.

1.8 Informationen im Netz

1.7 Wie und wo bekomme ich Debian GNU/Linux? Kapitel 2. Installation von Debian GNU/Linux

Kapitel 2. Installation von Debian GNU/Linux

1.8 Informationen im Netz 2.2 Debian Installation

Inhaltsverzeichnis

Wenn Sie bereits eine Linux-Distribution installiert haben, sind Sie schon mit den grundsätzlichen Schritten einer Linux-Installation vertraut. Debian GNU/Linux unterscheidet sich bei der Installation nur in wenigen Details von anderen Distributionen: Sie haben sehr viele Freiheiten und können viele Schritte beeinflussen; Sie müssen dies aber nicht tun.

Ein wesentlicher Unterschied gegenüber anderen Linux-Distributionen ist, dass Debian GNU/Linux während der Installation nicht über eine grafische Benutzeroberfläche konfiguriert werden kann. Alle Anpassungen sind über die Tastatur einzugeben. Dies hat zwei wesentliche Gründe: Zum einen ist Debian GNU/Linux auch für ältere Hardwareplattformen, wie beispielsweise m68k, verfügbar. Auf diesen Systemen ist eine grafische Benutzeroberfläche nur mit großen Eingeständnissen an die Geschwindigkeit benutzbar. Zum anderen gehört es zum Grundgedanken von Debian, dass nur die unbedingt notwendigen Komponenten installiert werden. Dies bedeutet auch, dass der kleinste gemeinsame Nenner eines Debian Systems, der durch das Debian Grundsystem gebildet wird, ein System ohne grafische Oberfläche ist. Systemadministratoren verzichten auf Serversystemen grundsätzlich auf grafische Benutzeroberflächen aus Sicherheitsgründen, aber auch, weil eine solche Oberfläche einfach unnötig ist.

Wenn Sie auf Probleme mit dieser Schnellinstallation stoßen, lesen Sie das Kapitel Debian Installation, dort wird ausführlicher auf die Installation eingegangen. Dieser Abschnitt soll fortgeschrittenen GNU/Linux-Anwendern die Möglichkeit geben, relativ schnell zu einem lauffähigen System zu kommen.

Platz - Sie benötigen eine freie oder nicht mehr benötigte Partition oder eine komplett unbenutzte Festplatte. Wenn Sie keine Partition frei haben, benutzen Sie das Programm fips.exe (auf der CD im Verzeichnis install/ als Archiv zu finden), um eine bestehende Partition unter DOS zu verkleinern.

Boot - Starten Sie den Rechner neu. Sie können entweder direkt von der CD-ROM oder DVD booten (ändern Sie im BIOS die Bootreihenfolge!), oder erstellen Sie Bootdisketten aus den Dateien /install/floppy/boot.img, /install/floppy/root.img, /install/floppy/cd-drivers.img und /install/floppy/net-drivers.img auf der CD-ROM. Benutzen Sie hierzu unter Linux das Programm dd mit der Option if=boot.img of=/dev/fd0 oder unter DOS das Programm rawrite2.exe.

Fragen - Beantworten Sie alle Fragen des Debian GNU/Linux-Installationsprogramms mit der Eingabe-/Returntaste, wenn Sie sich mit der Antwort nicht sicher sind.

International - Wählen Sie die gewünschte Tastaturbelegung aus.

Teilung - Erstellen Sie auf dem freien Festplattenplatz mindestens zwei Partitionen. Benutzen Sie eine als swap-, die andere als root-Partition.

Basis - Installieren Sie das Grundsystem von der CD-ROM oder einem anderen Medium.

Abschluss - Starten Sie das System neu und folgen Sie den Anweisungen zur abschließenden Konfiguration des Systems. In diesem Schritt können auch weitere Software-Pakete auf dem System installiert werden.

Kapitel 2. Installation von Debian GNU/Linux

1.8 Informationen im Netz 2.2 Debian Installation

2.2 Debian Installation

Kapitel 2. Installation von Debian GNU/Linux 2.3 Alternative Debian Installer

Die Debian Installation ist sehr gut und detailiert auf den Webseiten des Debian-Projektes beschrieben auf die an dieser Steller verwiesen werden soll: http://www.debian.org/releases/stable/i386/.

2.2 Debian Installation

Kapitel 2. Installation von Debian GNU/Linux 2.3 Alternative Debian Installer

2.3 Alternative Debian Installer

2.2 Debian Installation 2.4 Migration von anderen Distributionen

Neben dem vom Debian Team bereitgestellten Installer sind noch verschiedene andere Installer verfügbar. Diese wurden oft für sehr spezielle Hardware-Anforderungen und -Konfigurationen erstellt, die vom normalen Debian Installer nicht abgedeckt wurden. Auch kommen im Laufe der Zeit weitere Funktionalitäten oder Treiber in aktuelleren Linux-Kerneln hinzu, die während der Lebenszeit eines stabilen Debian Release noch nicht im Debian Installer abgedeckt waren.

Eine Übersicht alternativer Installer finden Sie unter http://linuxmafia.com/faq/Debian/installers.html.

Dort finden Sie auch Beschreibungen, wie Debian in einer „chroot“-Umgebung (beispielsweise per Remote-Zugriff bei einem Provider) installiert wird.

2.3 Alternative Debian Installer

2.2 Debian Installation 2.4 Migration von anderen Distributionen

2.4 Migration von anderen Distributionen

2.3 Alternative Debian Installer 2.5 FAI - Fully Automatic Installation

Wenn Sie bereits eine andere Linux-Distribution auf einem System einsetzen, so ist im Allgemeinen die Migration zu Debian GNU/Linux recht einfach umzusetzen. Zunächst werden die noch benötigten Daten gesichert, danach wird Debian wie beschrieben installiert. Natürlich funktioniert dieser Weg nur, wenn Sie physikalischen Zugriff auf das System haben.

Problematischer ist der Vorgang, wenn das System in einem Rechenzentrum untergebracht ist und keine Möglichkeit besteht, direkt auf das System zuzugreifen. Viele Anbieter von so genannten „root"-Servern bieten ihren Kunden zwar einen vollen Remote-Zugriff auf das System, eine Neuinstallation mit einer Wunsch-Distribution ist aber meist nicht vorgesehen.

Für die Migration zu Debian GNU/Linux hat der Debian Entwickler Guillem Jover ein Programm entwickelt, mit dem sich laufende Systeme von einer anderen Distribution (Novell/SuSE, RedHat/Fedora, Mandrake, Slackware, Gentoo usw.) in ein Debian System umwandeln lassen. Bisher wird dabei aber nicht das gesamte System ausgetauscht, vielmehr wird mittels deboostrap lediglich ein Grundsystem installiert. Der ursprüngliche Kernel sowie der Bootloader, aber auch die installierten Applikationen bleiben erhalten und können später gezielt ausgetauscht werden.

debtakeover finden Sie unter http://www.hadrons.org/~guillem/debian/debtakeover/.

Die notwendigen Schritte zur Installation von Debian auf einem existierenden System sind im Detail auch unter http://d-i.alioth.debian.org/manual/de.i386/apds03.html beschrieben.

2.4 Migration von anderen Distributionen

2.3 Alternative Debian Installer 2.5 FAI - Fully Automatic Installation

2.5 FAI - Fully Automatic Installation

2.4 Migration von anderen Distributionen | Kapitel 3. Betrieb

Mit der Einführung von Installationsprogrammen durch die Hersteller der großen Linux-Distributionen wurde es für den Administrator deutlich einfacher, Systeme mit GNU/Linux zu installieren. Nach dem Booten des Systems von Diskette oder CD-ROM/DVD wird ein Installationsprogramm gestartet. Es leitet den Administrator durch die Installation und Konfiguration zu einem mehr oder weniger komplett konfigurierten System.

Hierzu ist es aber in jedem Fall notwendig, während der Installation am System zu verbleiben, diverse Einstellungen vorzunehmen und die gewünschten Softwarekomponenten auszuwählen. Bei einem Ausfall des Systems muss das System, wie beschrieben, komplett neu installiert werden. Ein (hoffentlich) vorhandenes Backup kann einige Schritte vereinfachen (wenn dann das Restore auch funktioniert...).

„FAI“ (Fully Automatic Installation) bietet die Möglichkeit, einzelne oder mehrere Systeme anhand von Profilen, die auf einem Server hinterlegt sind, vollautomatisch zu installieren.

Abweichend von der klassischen Installation wird hierzu ein (FAI-)Server im Netz benötigt. Auf diesem befinden sich die Konfigurationsdateien, mit denen die Installation auf den verschiedenen Zielsystemen gesteuert wird. Dieser Server kann weitere Dienste bereitstellen, die während der Installation benötigt werden (wie beispielsweise FTP/HTTP zum Download von Software-Paketen oder DHCP zur automatischen Vergabe von IP-Nummern). Wenn diese Dienste aber bereits von anderen Servern im Netz bereitgestellt werden, so kann natürlich auf die bestehenden Ressourcen zurückgegriffen werden. Im Folgenden gehen wir davon aus, dass alle Dienste auf einem zentralen FAI-Server installiert und konfiguriert werden.

FAI (http://fai-project.org) wurde von Thomas Lange an der Universität Köln entwickelt, um dort Cluster-Installationen vorzunehmen. FAI benutzt die Debian Distribution als Basis und verwendet verschiedene Shell- und Perl-Skripts während des Installationsprozesses. Änderungen an den Konfigurationsdateien, um das System optimal an den zukünftigen Einsatz anzupassen, können auch über cfengine-Skripts vorgenommen werden.

Für Sun Solaris wird auf den FAI-Webseiten ebenfalls eine Version angeboten.

Zunächst ist es sinnvoll, sich einen Überblick über die Vorgänge bei der Installation mittels FAI zu verschaffen. Die Installation gliedert sich im Wesentlichen in die folgenden Aufgaben:

Konfiguration des Clients auf dem FAI-Server: MAC-Adresse in der Konfigurationsdatei des DHCP-Servers aufnehmen und die gewünschten Konfigurationen (Klassen) für den Client erstellen.

Die Konfiguration der Klassen bildet dabei den zentralen Teil der gesamten Konfiguration und Installation. Hier wird zunächst definiert, wie die Festplatte(n) eingeteilt werden soll(en), welche Dateisysteme benutzt und welche Software-Pakete auf dem System installiert werden.

Am Ende der Installation sorgen Klassen für Anpassungen wie die Änderung von Passwörtern oder die Integration des Clients in das Netzwerk (NIS, NFS usw.) oder auch die Einrichtung eines Drucksystems und die Anpassung der GNOME- oder KDE-Menüs.

Booten des Client mittels Bootdiskette oder Netzwerkboot (PXE). Der Client erhält eine IP-Nummer via DHCP, BOOTP oder auch statisch über die Bootdiskette und lädt über das Netzwerk den Linux-Kernel vom FAI-Server. Der Linux-Kernel wird auf dem Client gebootet.

Das Root-Filesystem wird via NFS vom FAI-Server auf dem Client gemountet, und das System wird gestartet. Hierbei werden angepasste Init-Skripts verwendt, die im Wesentlichen ein Minimalsystem aktivieren und die FAI-Umgebung initialisieren. Das zentrale Skript ist dabei rcS_fai.

Anhand der Konfiguration für den Client werden die notwendigen Aufgaben ermittelt und Klassen definiert. Diese Aufgaben gliedern sich in notwendige (Partitionieren der Festplatte, Grundsystem installieren usw.) und optionale Aufgaben (wie beispielsweise das Hinzufügen von Software oder das Anlegen von Benutzer-Accounts).

Im nächsten Schritt werden die Festplatten partitioniert, formatiert und auf dem Client eingebunden.

Nun wird ein Grundsystem auf dem Client entpackt. Dieses wurde auf dem Server aus dem NFS-Root-Verzeichnis generiert.

Sollten weitere Klassen mit Software-Paketen, die auf dem Client installiert werden sollen, definiert sein, so werden diese nun hinzugefügt.

Im nächsten Schritt wird das System über verschiedene Skripts angepasst und konfiguriert. An dieser Stelle können beispielsweise Anpassungen an der Mailkonfiguration vorgenommen oder Benutzerkonten eingerichtet werden.

Abschließend wird das Client-System neu gebootet und steht komplett konfiguriert und mit allen notwendigen Software-Paketen ausgestattet bereit.

Automatische Windows-Installationen

Bei der Verwaltung von größeren Netzwerken kommt der Administrator nicht immer um die Verwaltung von Windows-Systemen herum. Natürlich sollte generell immer die Migration zu Linux im Vordergrund stehen, aber die Realität sieht leider anders aus. Es gibt verschiedene Software-Projekte, die eine automatische Installation von Windows-Systemen erlauben. Auf Sourceforge finden Sie die Projekte http://unattended.sourceforge.net/ und http://ani.sourceforge.net/, die sich mit der Installation von Windows-Systemen beschäftigen. Weiterhin ist opsi (open - PC ServerIntegration, http://www.uib.de/www/produkte/opsi/) in der Lage, Windows Systeme zu installieren. Als Windows-Client-Betriebssystem wird derzeit Windows 2000 und Windows XP Professional unterstützt.

FAI wird durch entsprechende Pakete innerhalb der Debian Distribution (ab Version 3.1 „Sarge“) bereitgestellt. Die Installation kann somit via apt-get erfolgen. Neben dem Paket fai, das die notwendigen Programme, Skripts und Konfigurationsdateien enthält, sollte auch das Paket fai-kernels mit speziell an FAI angepassten Kerneln installiert werden.

```
apt-get install fai fai-kernels Reading Package Lists... 100% Building Dependency Tree... Done The following extra packages will be installed: debootstrap nfs-user-server Suggested packages: debmirror Recommended packages: netboot bootp tftpd The following NEW packages will be installed: debootstrap fai fai-kernels nfs-user-server 0 packages upgraded, 4 newly installed, 0 to remove and 528 not
```

```
upgraded. Need to get 7575kB of archives. After unpacking 9611kB of
additional disk space will be used. Do you want to continue? [Y/n]
```

Wie hier gezeigt, wird auf dem FAI-Server zwingend ein NFS-Server benötigt. Über diesen wird das Root-Dateisystem für den Client während der Installation bereitgestellt. Andere Dienste wie beispielsweise DHCP und DNS oder FTP können auch auf anderen Systemen installiert werden. FAI stellt für den Client während der Installation ein NFS-Root-Verzeichnis zur Verfügung, aus dem alle weiteren Skripts aufgerufen werden. Dieses NFS-Root-Verzeichnis wird später mit `debootstrap` über das Kommando `fai-setup` erzeugt.

Alle für FAI relevanten Parameter werden in der Datei `/etc/fai/fai.conf` eingestellt.

In der FAI-Konfigurationsdatei können Sie eine ganze Reihe Anpassungen vornehmen. Viele Variablen sind aber bereits mit sinnvollen Werten vorbelegt. Die Datei ist auch sehr gut kommentiert, so dass hier nur auf einige wichtige Einstellungen eingegangen werden soll.

`installserver`

Der Name des FAI-Installationsservers, dieser muss per DNS von den Clients aufgelöst werden können. Andernfalls kann hier auch die IP-Nummer des Systems angegeben werden.

`httpserver oder ftpserver`

Der Name eines Debian FTP- oder HTTP-Servers, der die benötigten Pakete bereitstellt.

`debdist`

Die zu installierende Debian Distribution, also beispielsweise „Woody“, „Sarge“ oder „Sid“.

`FAI_DEBOOTSTRAP`

Die Quelle, aus der die von `debootstrap` benötigten Pakete bezogen werden können. Dies kann ein lokales Verzeichnis oder ein FTP- / HTTP-Server sein.

FAI_DEBOOTSTRAP_OPTS

Zusätzliche Optionen für debootstrap. Hier können alle von debootstrap verwendeten Optionen angegeben werden, beispielsweise die zu verwendende Architektur oder auch zusätzlich zu installierende bzw. zu löschende Pakete.

FAI_SOURCES_LIST

Diese Variable, die über mehrere Zeilen gehen kann, beschreibt die auf den Clients und im NFS-Root zu verwendende Datei /etc/apt/sources.list. Diese kann die für APT üblichen Einträge enthalten.

NFSROOT_PACKAGES

Zusätzliche Pakete, die in das NFS-Root-Verzeichnis aufgenommen werden sollen. Dies kann beispielsweise für Serversysteme sinnvoll sein, dort können Tools wie lvm, raidtools oder xfstools aufgenommen werden. Hierbei ist zu beachten, dass solche Erweiterungen unter Umständen einen angepassten Kernel benötigen.

FAI_ROOTPW

Das verschlüsselte Passwort für den Systemadministrator. Vorgegeben ist das Passwort „fai“; ein verschlüsseltes Password kann aus der Datei /etc/shadow kopiert werden oder mit dem Programm makepasswd erzeugt werden.

KERNELPACKAGE

Ein Debian Kernel-Paket, das auf dem Client installiert wird. Dieses wird erst nach der Installation und einem ersten Reboot aktiviert. Während der Installation arbeitet der Client mit dem Kernel, der im NFS-Root installiert ist.

Das Kommando fai-setup sollte nach jeder Änderung der Datei /etc/fai/fai.conf aufgerufen werden. Im Wesentlichen wird hiermit das Kommando make-fai-nfsroot aufgerufen. Damit wird ein bestehendes NFS-Root-Verzeichnis gelöscht und neu angelegt.

Soll lediglich das NFS-Root-Verzeichnis neu generiert werden, so kann auch das Kommando make-fai-nfsroot eingesetzt werden.

Dabei ist insbesondere zu beachten, dass bei Verwendung der „testing“-Distribution Probleme auftauchen können. Dies liegt meistens daran, dass durch die rasante Entwicklung der Debian Distribution die von debootstrap benötigten Pakete nicht mehr oder in neuen Konstellationen verfügbar sind. Hier hilft in den meisten Fällen ein Update

von debootstrap auf dem FAI-Server, um die (internen) Listen mit dem Spiegel (Mirror) abzugleichen.

Jeder zu installierende Client wird von FAI über den Hostnamen identifiziert. Für jede durchzuführende Aktion gibt es eine „DEFAULT“-Klasse, die aufgerufen wird, falls keine spezielle Klasse für diesen Client vorhanden ist.

Im Verzeichnis /usr/share/doc/fai/examples/simple/ finden Sie einfache Beispiele für die wichtigsten Klassen, die als Basis für Anpassungen dienen können.

Es ist jedoch zu empfehlen, einige wenige Klassen in jedem Fall anzupassen. Ein guter Einstieg ist dabei die Klasse zur Konfiguration der Festplatte.

Der erste Arbeitsschritt bei der Installation eines Clients durch FAI ist in jedem Fall das Partitionieren der Festplatten und das Einrichten von Dateisystemen. Beide Schritte werden über Einträge in einer Klasse, beispielsweise /usr/share/doc/fai/examples/simple/disk_config/SMALL_IDE, definiert.

```
# generic disk configuration for one ide disk (size should not matter) # disk size from 530Mb up to what you can buy today # # <type> <mountpoint> <size in mb> [mount options]    [;extra options]
disk_config hda primary  /          30-100     rw,errors=remount-ro ;-c -j ext3 primary  /fai-boot    7          rw                ; -j ext3 logical  swap  50-500     rw                logical  /var        50-1000    rw                ; -m 5  -j ext3 logical  /tmp         50-1000    rw                ;-m 0 -j ext3 logical  /usr         300-4000   rw                ; -j ext3 logical  /home  50-4000    rw,nosuid         ;-m 1 -j ext3 logical  /scratch     0-  rw,nosuid         ;-m 0 -i 50000 -j ext3
```

Diese Datei kann Konfigurationen für mehrere Festplatten enthalten. Für jede ist ein entsprechender Abschnitt (in diesem Beispiel zunächst disk_config hda, für die zweite Festplatte disk_config hdb usw.) mit der gewünschten Partitionierung notwendig. In der ersten Spalte wird zunächst angegeben, ob eine primäre oder eine logische Partition

angelegt werden soll. Dabei werden die Partitionsnummern aufsteigend anhand der Reihenfolge vergeben.

Die zweite Spalte beschreibt das Verzeichnis im Dateisystem, an dem diese Partition später in das Dateisystem eingehängt werden soll.

Die nächste Spalte beschreibt die Größe der Partition in Megabyte. Dabei können feste Werte ebenso wie Bereiche angegeben werden. Die Aufteilung wird dann anhand des tatsächlichen Festplattenplatzes prozentual vorgenommen. Wird keine maximale Größe angegeben, so wird die Partition mit der maximal möglichen Größe angelegt. Sollen einzelne Partitionen auf der Festplatte erhalten bleiben (beispielsweise weil auf diesen bereits ein anderes Betriebssystem oder Daten liegen), so können diese Partitionen geschützt werden. Hierzu ist die Option preserve mit der entsprechenden Partitionsnummer (also beispielsweise preserve3 anzugeben. Erweiterte Partitionen können auf diesem Wege nicht geschützt werden, wohl aber die in den Partitionen liegenden Bereiche. Ein „bootable flag“ wird ebenfalls nicht gerettet. Partitionen, die mit preserve gekennzeichnet werden, werden während der Installation read-only gemountet.

Die vierte Spalte beeinhaltet die Optionen für den mount-Befehl. Abschließend können in der letzten Spalte hinter dem Semikolon zusätzliche Optionen angeben werden, die bei der Erstellung des Dateisystems benutzt werden sollen. Hier kann beispielsweise der Typ des Dateisystems (ext2, ext3 oder auch XFS) ausgewählt werden. Hier sind folgende Werte erlaubt:

boot: Markiert diese Partition als Boot-Partition.

-i: Angabe der Bytes pro Inode (nur für ext2- und ext3-Dateisysteme)

-m <blöcke>: Anzahl der für den Administrator reservierten Blöcke (nur für ext2- und ext3-Dateisysteme)

-j: Erzeugt ein ext3-Journal

-c: Überprüft die Festplatte beim Formatieren auf defekte Blöcke

ext2: Statt auto wird der Eintrag ext2 in der Datei /etc/fstab verwendet

ext3: Statt auto wird der Eintrag ext3 in der Datei /etc/fstab verwendet

swap: Bezeichnet eine Swap-Partition

dosfat16: Erzeugt ein DOS 16-Bit-Dateisystem

winfat32: Erzeugt ein Windows 95-FAT32-Dateisystem

reiser: Erzeugt ein Reiser-Dateisystem

xfs: Richtet ein XFS-Dateisystem ein

format: Formatiert die Partition in jedem Fall. Auch wenn preserve angegeben ist.

writeable: Bindet eine als preserve markierte Partition zum Lesen und Schreiben in das System ein

lazyformat: Formatiert die Partition nur, wenn sie verschoben wurde

Bei der Verwendung eines XFS-Dateisystems ist zu beachten, dass im NFS-Root-Verzeichnis das Paket xfs-utils installiert sein muss. Soll das XFS-Dateisystem auch für sehr kleine Partitionen, wie beispielsweise /boot, eingesetzt werden, dann denken Sie daran, dass die Mindestgröße 70 MByte betragen muss.

FAI benötigt während der Installation eine oder mehrere Quellen, aus denen Debian Pakete bezogen werden können. Wenn ein direkter Zugang zum Internet besteht, so können natürlich die öffentlichen Debian Server verwendet werden. Ein DebianSpiegel im lokalen Netz, beispielsweise auf dem FAI-Server, kann jedoch die Installationszeiten deutlich verringern, wenn mit einer hohen Bandbreite auf diesen Server zugegriffen werden kann.

Die Installation und Konfiguration eines FTP-Servers wird am Beispiel von ProFTP ebenfalls an anderer Stelle in diesem Buch beschrieben. Diese Informationen können ebenso für den Einsatz zusammen mit FAI genutzt werden.

2.5.3.1 Bootfloppy

Das Booten eines Clients kann auf unterschiedliche Arten erfolgen. Für die ersten Versuche ist sicherlich die Bootdiskette ein sehr einfacher, aber auch langsamer Weg. Eleganter geht es mit einer bootfähigen Netzwerkkarte. Diese erfordert aber zusätzliche Konfigurationsarbeit am Server. Dort muss der DHCP-Server entsprechend angepasst werden.

Nach dem Erstellen des NFS-Root-Verzeichnisses auf dem Server kann eine für alle Clients passende oder aber auch für jeden Client individuelle Bootdiskette erzeugt werden. Leider sind die Ladezeiten des Linux-Kernels von diesem Medium etwas lang, dafür sind aber keine weiteren Konfigurationsschritte notwendig. Für die ersten Experimente sollten Sie auf dieses Medium zurückgreifen.

Die Bootdiskette muss auf dem FAI-Server erzeugt werden. Wenn eine Diskette im Laufwerk /dev/fd0 liegt, kann dies mit dem Kommando make-fai-bootfloppy erfolgen. Es können verschiedene Optionen angegeben werden. Neben der Auswahl der Bootloaders (normalerweise GRUB) oder des Angabe eines festen Servers sind insbesondere die für die Steuerung von FAI relevanten Optionen interessant.

Nach dem Kommando make-fai-bootfloppy können, in Anführungszeichen stehend, beliebige FAI-Variablen gesetzt werden. Diese werden beim Start von FAI ausgewertet, und die entsprechenden Umgebungsvariablen werden gesetzt.

Eine Übersicht aller Optionen erhält man mit make-fai-bootfloppy -h:

make-fai-bootfloppy, create a boot floppy for FAI. Version 2.8.5, 27-november-2006 Copyright (C) 2000-2006 by Thomas Lange Usage: make-fai-bootfloppy [parameter] -B make a big 2.88M floppy instead of the default 1.44M floppy. -c CFDIR use CFDIR instead of /etc/fai as configuration directory -d LABEL use LABEL when selecting the default boot kernel (and parameters). a for any boot protocol (kernel tries all compiled in) b for BOOTP d for DHCP f use fixed IP, needs companion option -s r for RARP Without this option DHCP is used. -h print this message. -F append default flags to kernel parameters. Same as "FAI_FLAGS=verbose,sshd,createvt" -f FILE make a floppy image in FILE -g use GRUB loader on bootfloppy (default) -i FILE make a iso9660 image in FILE -I IF use IF as the client's network interface -l use LILO loader on bootfloppy -m DIR use DIR as mountpoint [/floppy or /media/floppy] -s HOST use this static ip for FAI client; try to get all info from DNS -v print verbose output DESCRIPTION Creates a boot floppy for booting a FAI install client. No arguments are needed but you must be root. You may need to use "nfsroot=serverip:path" if you use RARP or if your BOOTP or DHCP server does not pass that info to the clients. All parameters are passed to the kernel via append in lilo.conf, or the kernel commandline when using grub. EXAMPLE Create a generic boot floppy for James ;-) # make-fai-bootfloppy "FAI_FLAGS=verbose,createvt,sshd BOND=007" Make a boot

```
floppy with some common flags and action sysinfo     # make-fai-bootfloppy -vF FAI_ACTION=sysinfo
```

Wie die Beispiele am Ende schon zeigen, lassen sich die vielfältigsten Variationen von Bootdisketten für alle möglichen Einsatzzwecke erzeugen.

FAI BootCD dient zum Erstellen von bootfähigen CD-ROMs, die eine komplette FAI-Umgebung inklusive aller benötigten Software-Pakete beinhalten. Somit kann von einer CD eine komplette Installation auch ohne eine Verbindung zu einem Netzwerk durchgeführt werden.

FAI BootCD greift auf viele Werkzeuge aus dem FAI-Paket zurück, es ist also zwingend erforderlich, dass FAI bereits auf dem System installiert und konfiguriert ist, auf dem FAI BootCD eingesetzt werden soll. Bevor also die ersten Schritte mit FAI BootCD unternommen werden, muss es möglich sein, aus dem FAI-System heraus eine komplette Installation durchzuführen.

Ist dies der Fall, so muss zunächst die FAI-Konfigurationsdatei für FAI BootCD bereitgestellt werden.

```
cp -f /etc/fai/fai.conf /etc/fai-bootcd/fai.conf
```

In dieser Datei können nun Anpassungen vorgenommen werden. Beispielsweise kann es sinnvoll sein, zwei Verzeichnisse für die NFS-Root-Umgebungen für FAI und FAI BootCD vorzuhalten.

Der Kernel auf dem System, auf dem das CD-Image erzeugt wird, muss die Unterstützung für das Loopback-Device als Modul enthalten. Zukünftige Versionen von mkinitrd-cd werden hoffentlich auch die Verwendung dieser Funktion als festen Bestandteil des Kernels erlauben.

Weiterhin muss die Unterstützung für das RAM-Dateisystem im Kernel enthalten sein.

Die notwendigen Optionen in der Kernel-Konfiguration (die Datei .config im Kernel-Source-Verzeichnis) lauten:

```
CONFIG_BLK_DEV_LOOP=m CONFIG_CRAMFS=y
```

Die bisher im Paket fai-kernel enthaltenen Kernel verfügen nicht über alle von FAI BootCD benötigten Funktionen. Es ist also ein Kernel zu erstellen, der die folgenden Funktionen enthält:

```
CONFIG_BLK_DEV_INITRD=y CONFIG_IDE=y
CONFIG_BLK_DEV_IDE=y CONFIG_ISO9660_FS=y
CONFIG_EXT2_FS=y CONFIG_ZISOFS_FS=y
```

Mit dieser Konfiguration wird nun ein Debian Kernel-Paket (mittels make-kpkg, siehe make-kpkg) erzeugt und über die Variable KERNELPACKAGE in der Datei /etc/fai-bootcd/fai.conf eingebunden.

Dieser Kernel wird ausschließlich auf der FAI BootCD und lediglich zur Installation verwendet. Soll dieser Kernel auch während der Installation auf dem Zielsystem installiert werden, so müssen Sie dafür sorgen, dass alle benötigten Treiber (insbesondere für IDE/SCSI-Festplatten, aber natürlich möglichst auch für alle anderen Hardware-Komponenten) in diesem Kernel enthalten sind. Natürlich kann aber auf den Zielsystemen auch jeder andere Kernel installiert werden.

Im Verzeichnis /etc/fai-bootcd/ können nun Anpassungen vorgenommen werden. In der Datei id.txt kann ein Text definiert werden, um die CD auf Systemen mit mehreren CD-ROM-Laufwerken eindeutig zu erkennen.

Die Datei bootmsg.txt enthält einen Text, der bei Verwendung von ISOLINUX bzw. SYSLINUX als Bootloader angezeigt wird. Normalerweise wird aber GRUB als Bootloader eingesetzt, so dass diese Datei ignoriert wird.

In der Datei updatebase.DEFAULT sind Angaben für FAI hinterlegt, um das Verzeichnis /fai/ auf der CD-ROM während der Installation in die chroot-Umgebung einzubinden. Wird eine weitere Datei updatebase.DEFAULT im Verzeichnis FAI_CONFIGDIR/hooks/ verwendet, so können diese beiden kombiniert werden.

Die Datei menu.lst ist die Konfigurationsdatei für den GRUB-Bootloader. Die Voreinstellung ist so gewählt, dass der Linux-Kernel (/vmlinuz) von der zweiten Partition der ersten Festplatte (hd0,1) geladen wird. Dies bewirkt, dass bei eingelegter FAI BootCD, zunächst geprüft wird, ob auf der lokalen Festplatte bereits ein System installiert ist. Ist dies der Fall, so wird dieses gestartet. Ist kein System (Kernel-Image) vorhanden, so wird eine Neuinstallation durchgeführt.

Soll also ein System neu installiert werden, müssen Sie einfach das Kernel-Image auf der Platte entfernen, und schon kann von der FAI BootCD direkt gebootet werden.

In der Datei fai-variables.conf sind FAI-Variablen enthalten, die normalerweise von der NFS-Root-Umgebung oder der FAI Bootdiskette bereitgestellt werden. Hier können Variablen angepasst werden, die die bisherigen Werte überschreiben.

Nun sind alle Vorbereitungen getroffen, um eine FAI Boot CD zu erzeugen: Das Kommando make-fai-bootcd -r -n erzeugt zunächst ein neues NFS-Root-Verzeichnis auf Basis der Klassen, die über FAI_CONFIGDIR definiert sind. Weiterhin wird ein Paket-Repository angelegt (mit make-fai-repository), das alle notwendigen Software-Pakete enthält.

Dieser Vorgang wird eine Datei fai-bootcd.iso im aktuellen Verzeichnis erzeugen. Diese Datei kann direkt auf eine CD-ROM gebrannt werden. Alternativ kann auch das Kommando

```
make-fai-bootcd -r -n -b
```

verwendet werden, dieses ruft abschließend cdrecord auf und schreibt das Image direkt auf eine CD.

Für die ersten Versuche ist es ratsam, das Image nicht sofort zu brennen, sondern die erzeugte Datei zunächst via Loopback-Device zu mounten und einen Blick auf den Inhalt zu werfen. Das Kommando hierfür lautet:

```
mount -o loop ./fai-bootcd.iso /mnt
```

Die von FAI BootCD verwendete GRUB-Konfiguration prüft beim Systemstart zunächst, ob bereits ein Kernel auf der Festplatte vorhanden ist. Ist dies nicht der Fall, so wird eine Installation durchgeführt. Andernfalls wird das vorhandene System gestartet, falls nicht aus dem Menü der Eintrag cdrom gewählt wird.

Nach Abschluss der Installation wird das neue System automatisch neu gestartet. Es ist ratsam, die CD-ROM zu entfernen, insbesondere wenn der Kernel nicht wie beschrieben auf der zweiten Partition der ersten Platte installiert wurde.

2.5 FAI - Fully Automatic Installation

2.4 Migration von anderen Distributionen | Kapitel 3. Betrieb

Kapitel 3. Betrieb

2.5 FAI - Fully Automatic Installation 3.2 Allgemeines zum neuen System

Inhaltsverzeichnis

Im Folgenden finden Sie eine Einführung in die wichtigsten Unix-Kommandos sowie eine Beschreibung des Editors vi. Im Gegensatz zu anderen Betriebssystemen speichert Debian GNU/Linux alle nötigen Einstellungen in reinen Textdateien und nicht in Binär-Dateien, wie dies bei anderen Betriebssystemen der Fall ist. Diese lassen sich mit den einfachsten Werkzeugen, wie zum Beispiel dem Editor vi, bearbeiten. Es werden keine Programme mit grafischer Benutzeroberfläche benötigt. Der Systemadministrator hat 100%igen Zugriff auf alle Einstellungen, zur Not auch von außerhalb über eine simple Telefonleitung oder über das Internet via telnet oder besser ssh. Es werden die wichtigsten Grundlagen zur Systemadministration vermittelt, so dass Sie in jedem Fall Ihr System wieder zum Leben erwecken können.

Sicher gibt es einfacher zu benutzende Editoren als vi; dieser hat jedoch den Vorteil, schon direkt nach der Basisinstallation von Debian GNU/Linux vorhanden zu sein. Auch auf anderen Unix-Systemen trifft man auf diesen Standard-Editor, so dass man die Vorkenntnisse auch hier nutzen kann.

Da bisher keine grafische Oberfläche installiert wurde (diese ist beispielsweise für den Betrieb eines Servers nicht notwendig), finden Sie im Folgenden ausschließlich textbasierte Programme.

Debian GNU/Linux stellt dem Systembetreiber aber auch zahlreiche grafische Werkzeuge zur Administration zur Verfügung, einige von ihnen werden später vorgestellt.

Kapitel 3. Betrieb

2.5 FAI - Fully Automatic Installation 3.2 Allgemeines zum neuen System

3.2 Allgemeines zum neuen System

Kapitel 3. Betrieb 3.3 Ein Multiuser-, Multitasking-Betriebssystem

Wenn Sie es bis hierher geschafft haben, Debian GNU/Linux auf Ihrem System zu installieren und das erste Mal zu starten, steht Ihnen nun ein lauffähiges Unix-artiges Betriebssystem zur Verfügung. Es wurden alle notwendigen Systemdienste und Werkzeuge installiert, und von dieser Stelle aus können mit den vorhandenen Mitteln weitere Programme installiert werden. Somit können Sie das System konfigurieren und an Ihre eigenen Bedürfnisse anpassen.

Alle installierten Programme wurden schon für Sie sinnvoll vorkonfiguriert. Dennoch ist es natürlich an vielen Stellen angebracht, dem System den letzten Schliff zu geben. Im Folgenden erfahren Sie einiges über den Aufbau und die Funktionalität des Debian GNU/Linux-Systems, danach wird auf einige spezielle Programme eingegangen.

3.2 Allgemeines zum neuen System

Kapitel 3. Betrieb 3.3 Ein Multiuser-, Multitasking-Betriebssystem

3.3 Ein Multiuser-, Multitasking-Betriebssystem

3.2 Allgemeines zum neuen System 3.4 Anmelden am System

Debian GNU/Linux basiert auf dem Design der in den 60er Jahren entstandenen Unix-Systeme. Anders als die im täglichen Gebrauch - zu Hause oder in Büros - verbreiteten DOS-, Windows- und MacOS-Betriebssysteme, ist Unix im Serverbereich und auf Systemen mit vielen Benutzern auf ein und demselben Rechner verbreitet.

Dies bedeutet u.a., dass Debian GNU/Linux von Haus aus viele Funktionen mitbringt, die den anderen Betriebssystemen fehlen oder die zusätzlich erworben und installiert werden müssen. Debian erlaubt mehreren Benutzern die gleichzeitige Nutzung eines Rechners (Multiuser); hierzu ist es nötig, mehrere, auch gleiche Programme, zur selben Zeit auszuführen. Diese Funktion nennt man Multitasking.

Ein Großteil der Komplexität und Leistungsfähigkeit von Unix-Systemen hat ihren Ursprung in diesen beiden Funktionen. Zum Beispiel muss das System bei mehreren Benutzern verhindern, dass ein Benutzer Dateien eines anderen versehentlich löschen kann.

Hat man dies einmal verstanden, fällt es viel leichter, viele der Vorgänge und Eigenheiten eines Unix-Systems zu verstehen. Sie werden im Folgenden lernen, diese beiden Funktionen sinnvoll zu nutzen.

3.3 Ein Multiuser-, Multitasking-Betriebssystem

3.2 Allgemeines zum neuen System 3.4 Anmelden am System

3.4 Anmelden am System

3.3 Ein Multiuser-, Multitasking-Betriebssystem 3.5 Anmelden als Administrator (root)

Um Debian GNU/Linux zu benutzen, müssen Sie sich am System anmelden. Dies geschieht über die Eingabe eines so genannten Usernamens und eines Passwortes. Über diese Kombination kann das System feststellen, welche Zugriffsrechte der Benutzer hat und welche persönlichen Einstellungen zu verwenden sind.

Wenn Sie Debian GNU/Linux selbst installiert haben, wurden Sie bereits bei der Installation nach einem Benutzernamen und einem Passwort sowie einigen anderen Angaben gefragt. Möchten Sie sich an einem anderen System anmelden, fragen Sie den Systembetreuer nach einer Zugangsberechtigung.

Sie sollten in jedem Fall nicht auf die Idee kommen, für Ihren Zugang zu einem Linux-System auf ein Passwort zu verzichten (auch wenn dies grundsätzlich möglich ist und Sie „sowieso nur allein zu Hause“ an dem Computer arbeiten!). Ein Zugang ohne Passwort steht für jeden anderen spätestens bei der ersten Verbindung ins Internet offen, bitte bedenken Sie das!

Auch ist es bei der Installation möglich, auf die Einrichtung eines „normalen“ Benutzerkontos zu verzichten. Dies mag in einigen wenigen Fällen sinnvoll sein, es ist aber meistens davon abzuraten. Der Administrator (root) hat generell alle Zugriffsrechte an einem System. Debian GNU/Linux installiert aus Sicherheitsgründen einige Pakete so, dass die zugehörigen Programme nicht als Administrator ausgeführt werden können. Weiterhin ist es standardmäßig unter Debian GNU/Linux nicht ohne Weiteres möglich, sich als Administrator über ein Netzwerk am System anzumelden. Es kann also zu Problemen und Verwirrung kommen, wenn Sie kein normales Benutzerkonto angelegt haben.

Wenn Sie Ihr Debian GNU/Linux-System starten, sehen Sie eine Aufforderung zur Anmeldung am System. Je nach Konfiguration kann dies eine textbasierte oder eine grafische Anmeldung sein. Im einfachsten Fall sehen Sie in etwa Folgendes:

```
Debian GNU/Linux woody surimi tty1

surimi login:
```

Bei einer unveränderten Debian Installation stehen in der ersten Zeile nach „Debian GNU/Linux“ der Name der Debian Version (hier „woody“ für die Version 3.0), der Name des Rechners („surimi“) sowie der Name und die Nummer der Konsole, auf der Sie sich befinden.

Geben Sie nun Ihren Benutzernamen ein, und drücken Sie **RETURN**: Sie werden dann nach Ihrem Passwort gefragt.

```
Passwort:
```

Geben Sie hier Ihr gewähltes Passwort ein. Beachten Sie, dass das Passwort bei der Eingabe aus Sicherheitsgründen nicht angezeigt wird. Sollte die Anmeldung fehlschlagen, prüfen Sie, ob eventuell die Feststelltaste (**SHIFT LOCK**) gedrückt wurde.

War die Anmeldung erfolgreich, sehen Sie eine kurze Willkommensnachricht, die in etwa so aussieht:

```
Linux surimi 2.4.4 #1 Wed May 16 09:21:01 EST 2001 i686 unknown
Copyright (C) 1993-1999 Software in the Public Interest, and others
Most of the programs included with the Debian GNU/Linux system are
freely redistributable; the exact distribution terms for each program are
described in the individual files in /usr/doc/*/copyright  Debian
GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the
extent permitted by applicable law. Last login: Sat Jul  3 21:53:40 on
tty4. You have mail. $
```

Das letzte Zeichen, $, wurde von einem Programm, der so genannten *Shell*, ausgegeben. Im Englischen bedeutet „to prompt“ so viel wie „veranlassen“. In diesem Fall soll der Benutzer zu einer Eingabe veranlasst werden. Deshalb nennt man dies auch den „Shell-Prompt“ oder im Deutschen „Eingabeaufforderung“. Hier können Sie verschiedene Kommandos eingeben und so das System steuern oder Programme starten.

Als erster Versuch eignet sich das Kommando whoami gut.

Der Cursor (deutsch: „Einfügemarke“) rechts neben dem Shell-Prompt (meist ein blinkender Unterstrich (_)) zeigt an, an welcher Stelle das nächste Zeichen eingefügt wird. Tippen Sie den Befehl whoami ein, und drücken Sie **RETURN**.

whoami zeigt Ihren Benutzernamen an; Sie gelangen dann wieder zum Shell-Prompt zurück.

Wenn Sie Ihre Arbeit mit Debian beendet haben, sollten Sie sich wieder vom System abmelden. Hierfür können Sie den Befehl „exit“ oder auch „logout“ benutzen. Eine weitere Möglichkeit ist das gleichzeitige Drücken der Tasten **STRG+d**; auch hiermit verlassen Sie die momentane *shell*, Ihre Benutzerumgebung.

Bedenken Sie, dass - wenn Sie sich nicht vom System abmelden - andere Leute Ihren Zugang missbrauchen können, falls Sie sich von Ihrem Rechner entfernen!

3.4 Anmelden am System

3.3 Ein Multiuser-, Multitasking-Betriebssystem 3.5 Anmelden als Administrator (root)

3.5 Anmelden als Administrator (root)

3.4 Anmelden am System 3.6 Benutzerverwaltung

Debian GNU/Linux ist als Multiuser-System dafür ausgelegt, mehrere Benutzer gleichzeitig bei der Arbeit zu unterstützen. Hierbei wurde besonderer Wert darauf gelegt, dass ein Programm eines Benutzers, wenn es denn mal abstürzt, nicht das komplette System abstürzen lässt. Über Zugriffsrechte, die den Benutzern eines Systems von dem Systemadministrator zugewiesen werden, wird verhindert, dass wichtige Dateien im System von Unbefugten verändert oder gar gelöscht werden.

Manchmal ist es aber nötig, selbst wichtige Systemdateien zu verändern. Hierzu gehört die Installation von neuen Programmen (normalerweise über das Programm dselect) oder auch die Konfiguration des Netzwerks. Ein solcher Eingriff ist ebenfalls nötig, wenn man eine serielle Maus angeschlossen hat und diese gegen ein Modell mit PS/2-Anschluss austauschen möchte.

Um diese Änderungen vornehmen zu können, müssen Sie mehr Zugriffsrechte zu Ihrem System erlangen als Ihnen als normaler Benutzer zur Verfügung stehen. Sie müssen „root“ werden, sich also als Systemadministrator am System anmelden.

Am einfachsten gelingt dies, wenn Sie sich bereits am Login-Prompt mit dem Benutzernamen root und dem dazugehörigen Passwort anmelden. Das Passwort haben Sie bereits bei der Installation gewählt, wenn Sie das System selbst installiert haben. Arbeiten Sie an einem fremden System, bitten Sie den Systemadministrator, die gewünschten Einstellungen für Sie vorzunehmen.

Manchmal ist das Passwort für den Administrator auch mehreren Personen bekannt. Dies ist z.B. an Universitäten oder in Firmen üblich, um zu gewährleisten, dass möglichst immer ein Systembetreuer ansprechbar ist.

Melden Sie sich nun als Administrator (root) an. Überprüfen Sie mit dem Befehl whoami Ihre Identität. Melden Sie sich möglichst bald wieder ab, wenn Sie als Administrator am System gearbeitet haben! Sie haben als Administrator alle Zugriffsrechte, um das gesamte System im schlimmsten Fall zu zerstören. Spielen Sie nicht mit dem System herum, solange Sie als Administrator angemeldet sind. Führen Sie nur die absolut notwendigen Arbeiten aus und melden Sie sich dann als normaler Benutzer wieder an!

Alternativ besteht auch die Möglichkeit, den Befehl su zu benutzen. Sie können so die Rechte eines anderen Benutzers erlangen, ohne sich am System ab- und wieder anmelden zu müssen.

Probieren Sie es einmal aus: Melden Sie sich mit Ihrem normalen Benutzernamen an (nicht root!). Geben Sie das Kommando su ein. Sie werden nun nach dem Passwort für den Administrator gefragt: Geben Sie es ein. Der Shell-Prompt sollte nun von $ auf # wechseln. Sie können so leicht feststellen, dass Sie nun als Administrator angemeldet sind. Natürlich können Sie dies auch jederzeit wieder mit whoami überprüfen.

Sie können mit dem Befehl `su` auch die Identität jedes anderen Benutzers annehmen, solange Sie das Passwort kennen oder als Administrator am System angemeldet sind. Benutzen Sie hierzu ebenfalls den Befehl `su` und geben Sie dahinter (durch ein Leerzeichen getrennt) den Namen des Benutzers an. Beispiel: `su donald`.

Vielleicht werden Sie bemerken (nachdem Sie das Kommando `su` einige Zeit benutzt haben), dass nicht alle Einstellungen so sind, wie Sie vielleicht erwarten. Um auch mit dem Kommando `su` alle Einstellungen so vorzufinden, als ob man sich direkt von einem Login-Prompt angemeldet hätte, benutzen Sie den Befehl `su - donald`, also mit dem Zeichen „ - “ nach dem Kommando `su`.

3.5 Anmelden als Administrator (root)

3.4 Anmelden am System 3.6 Benutzerverwaltung

3.6 Benutzerverwaltung

3.5 Anmelden als Administrator (root) 3.7 Virtuelle Konsolen

3.6.1 Benutzerkonten hinzufügen

3.6.2 Benutzerkonten löschen

Die Benutzerverwaltung eines Systems stellt verschiedene Anforderungen an den Systemadministrator. Bereits während der Installation von Debian GNU kann ein Benutzerkonto hinzugefügt werden. Dies sollte auch auf jeden Fall geschehen. Nachdem das System in den produktiven Betrieb übergegangen ist, sind Aufgaben wie das Anlegen oder Löschen von Benutzerkonten, das Anpassen von Zugriffsrechten oder die Vergabe von Quota (beispielsweise zur Beschränkung des Festplattenplatzes je Benutzer) durch den Systemadministrator zu erledigen.

Im Folgenden wird es um das Hinzufügen und Löschen von Benutzerkonten gehen.

Das Hinzufügen neuer Benutzerkonten ist Aufgabe des Systemadministrators und muss daher mit Administrator-Rechten durchgeführt werden. Neben dem klassischen Weg, dem Hinzufügen der erforderlichen Daten mittels eines Editors in den entsprechenden Dateien, steht unter Debian GNU das Programm adduser zur Verfügung.

adduser [--home DIR] [--shell SHELL] [--no-create-home] [--uid ID] [--firstuid ID] [--lastuid ID] [--gecos GECOS] [--ingroup GROUP | --gid ID] [--disabled-password] [--disabled-login] user Add a normal user adduser --system [--home DIR] [--shell SHELL] [--no-create-home] [--uid ID] [--gecos GECOS] [--group | --ingroup GROUP | --gid ID] [--disabled-password] [--disabled-login] user Add a system user adduser --group [--gid ID] group addgroup [--gid ID] group Add a system group adduser user group Add an existing user to an existing group Global configuration is in the file /etc/adduser.conf. Other options are [--quiet] [--force-badname] [--help] [--version] [--conf FILE].

adduser kennt eine Reihe von Optionen, die in der Manpage beschrieben sind oder, in verkürzter Form, mittels adduser -h angezeigt werden. Für den „Hausgebrauch“ ist es

ausreichend, das Programm ohne weitere Optionen aufzurufen; adduser verfügt über einen interaktiven Modus, in dem alle notwendigen Angaben erfragt werden.

```
sushi:~# adduser  Enter a username to add: dd Adding user dd... Adding new group dd (1001). Adding new user dd (1001) with group dd. Creating home directory /home/dd. Copying files from /etc/skel Enter new UNIX password:  Retype new UNIX password:  passwd: password updated successfully Changing the user information for dd Enter the new value, or press return for the default      Full Name []: Donald Duck       Room Number []:       Work Phone []:  Home Phone []:       Other []:  Is the information correct? [y/n] y
```

Zunächst müssen Sie die Bezeichnung für das Benutzerkonto (Loginname) angeben. Hier bietet es sich an, eine kurze Zeichenkette zu wählen; diese Angabe wird bei jeder Anmeldung am System benötigt.

Unter Debian GNU wird für jeden neuen Benutzer auch gleich eine eigene Gruppe erzeugt. Die Heimatverzeichnisse der Benutzer (mit Ausnahme von „root“) befinden sich unterhalb von /home/. Dort wird für jeden Benutzer ein Verzeichnis mit dem Namen des Benutzerkontos erzeugt. Weiterhin werden alle Dateien aus dem Verzeichnis /etc/skel/ in das Heimatverzeichnis des Benutzers kopiert. Hierbei handelt es sich um verschiedene Voreinstellungen, beispielsweise für die bash, die dann von Benutzern angepasst werden können.

```
sushi:~# ls -a /etc/skel/  .  ..  .alias  .bash_logout  .bash_profile  .bashrc  .cshrc
```

Es wird dann (zur Sicherheit doppelt) nach einem Passwort gefragt. Weiterhin sollte der „Full Name“ des Benutzers korrekt angegeben werden. Alle weiteren Angaben stammen aus der Urzeit der Unix-Systeme und müssen nicht ausgefüllt werden. Abschließend kann noch geprüft und bestätigt werden, ob alle Angaben korrekt ausgefüllt wurden. Ist dies der Fall, so wird das Benutzerkonto dem System hinzugefügt, und der Zugang steht ab sofort zur Verfügung. Wird das Programm abgebrochen (durch Eingabe von **n** oder auch **STRG+C**, so werden alle bereits erzeugten Dateien, Verzeichnisse und Einträge in Konfigurationsdateien wieder entfernt.

Das Entfernen von Benutzerkonten aus dem System ist deutlich einfacher. Hierzu steht das Kommando deluser zur Verfügung. Auch dieses Programm kann interaktiv benutzt werden; hier aber ein Beispiel mit der Angabe des Login-Namens auf der Kommandozeile:

```
sushi:~# deluser dd Removing user dd... done.
```

Zu beachten ist, dass dabei nicht das Heimatverzeichnis des Benutzers gelöscht wird. Dies bleibt dem Systemadministrator überlassen.

3.6 Benutzerverwaltung

3.5 Anmelden als Administrator (root) 3.7 Virtuelle Konsolen

3.7 Virtuelle Konsolen

3.6 Benutzerverwaltung 3.8 Systemstart und -stop bei Debian

Der Linux-Kernel unterstützt virtuelle Konsolen. Dies ist eine Methode, um Ihren Bildschirm und die Tastatur (sowie die Maus, falls das Programm gpm installiert wurde) so benutzen zu können, als würden Sie an mehreren Geräten gleichzeitig arbeiten und als wären diese alle mit dem gleichen Rechner-System verbunden.

Die Benutzung der virtuellen Konsolen ist sehr einfach. Über die Tastenkombinationen **ALT+F1** bis **ALT+F6** können Sie zwischen den verschiedenen virtuellen Bildschirmen umschalten. Probieren Sie es aus, indem Sie sich mehrfach einloggen und auf den verschiedenen Konsolen unterschiedliche Befehle ausführen.

Debian GNU/Linux ist standardmäßig für die Benutzung von sechs virtuellen Konsolen eingerichtet; auf diesen wird automatisch das Programm login gestartet. Sie können auf diese mit den Funktionstasten **F1** bis **F6** zugreifen (jeweils zusammen mit der **ALT**-Taste). Technisch sind auch noch mehr virtuelle Konsolen möglich. Debian GNU/Linux verwendet die siebte Konsole (**ALT+F7**) für das X Window-System, die grafische Benutzeroberfläche.

Wenn Sie X benutzen, so verwendet dies automatisch die siebte virtuelle Konsole. Diese wird beim Start von X automatisch aktiviert, unabhängig davon, ob Sie das Kommando startx benutzen oder X über xdm oder ein anderes Programm starten.

Um von der grafischen Oberfläche X wieder auf die textbasierte Konsole zu wechseln, drücken Sie die Tastenkombination **STRG+ALT+F1** (oder **F2**, **F3** usw.).

Sie müssen sich also nur merken, dass - wenn Sie von X auf eine Konsole wechseln möchten - zusätzlich die Taste **STRG** zu drücken ist.

Wenn Sie sich einmal mit der Arbeitsweise der virtuellen Konsolen vertraut gemacht haben, werden Sie diese zu schätzen wissen. Sie können so schnell zwischen einem Editor und dem Compiler umschalten und auf der dritten Konsole noch die Logdateien im Auge behalten. Dies können Sie auch unter X erreichen, indem Sie mehrere Fenster öffnen. In vielen Fällen ist es aber gar nicht nötig oder sogar unerwünscht, auf einem Rechner X zu installieren - beispielsweise bei einem Server.

Es ist sogar möglich, mehrere X-Server auf verschiedenen virtuellen Konsolen zu starten. Dies kann für den gleichzeitigen Betrieb von X mit verschiedenen Farbtiefen sinnvoll sein.

3.7 Virtuelle Konsolen

3.6 Benutzerverwaltung 3.8 Systemstart und -stop bei Debian

3.8 Systemstart und -stop bei Debian

3.7 Virtuelle Konsolen 3.9 Kommandozeile und Dokumentation

Der Start eines Debian Systems ähnelt zunächst sehr dem Systemstart anderer Linux-Versionen. Nach dem Einschalten des Systems lädt der Bootloader (meist GRUB oder LILO auf X86-Systemen) den gewünschten Kernel in den Speicher. Der Kernel wird gestartet und bindet das Wurzeldateisystem (Root) ein. Als erstes Programm wird /sbin/init aufgerufen und erhält somit immer die Prozessnummer 1, alle weiteren Prozesse werden direkt oder indirekt von init gestartet. Man bezeichnet init deshalb auch als „Vater aller Prozesse". init liest die Konfigurationsdatei /etc/inittab, in der festgelegt ist, dass zunächst das Skript /etc/init.d/rcS ausgeführt werden soll.

Dieses Skript führt in einer festgelegten Reihenfolge (aufsteigend in numerischer/alphabetischer Reihenfolge) alle Skripte im Verzeichnis /etc/rcS.d aus. Skripte, deren Dateiname mit einem „K" beginnt, werden mit der Option „stop" ausgeführt. Mit dem Buchstaben „S" beginnende Skripte dagegen werden mit der Option „start" ausgeführt. Mit diesen Skripten werden beispielsweise alle weiteren Partitionen eingebunden, Module geladen, das Netzwerk aktiviert und Dienste gestartet.

Debian verwendet das vom Unix-System V eingeführte Konzept verschiedener Zustände des Systems, die so genannten Runlevel. Ein Runlevel ist ein Software-Zustand des Systems, der es nur einer bestimmten Gruppe von Prozessen erlaubt, ausgeführt zu werden. So können Runlevel beispielsweise für unterschiedliche Betriebsmodi eines Systems verwendet werden. Es ist möglich, zwischen (fast allen) Runleveln im laufenden Betrieb zu wechseln, ohne das System neu starten zu müssen.

Jeder der verfügbaren Runlevel kann mit dem Kommando telinit n, wobei n die Zahl des gewünschten Runlevels ist, aktiviert werden. Zum Einsatz kommen die Runlevel 0-9 sowie S und s, wobei den Runleveln 0, 1 und 6 eine besondere Bedeutung zukommt.

0

halt, hält das System an, ohne es neu zu starten.

1

bringt das System in den Single-User-Modus. Es werden nur die minimal benötigten Dienste gestartet, Netzwerkschnittstellen sind nicht aktiviert. Dieser Modus ist für Wartungsarbeiten gedacht.

2 bis 5

Multiuser Modus, in diesen Runleveln wird üblicherweise gearbeitet. Ein unverändertes Debian System läuft im Normalbetrieb im Runlevel 2.

6

Dieser Runlevel dient zum Neustarten (Reboot) des Systems.

7 bis 9

Diese Runlevel werden nicht verwendet. `init` ist aber in der Lage, diese zu benutzen, wenn vom Administrator entsprechende Skripte und Verzeichnisse angelegt werden.

S und s

bringen das System in den Single-User-Modus. Siehe auch Runlevel 1.

a bis c

So genannte „ondemand"-Runlevel. Siehe inittab.

Schalten Sie Ihr Debian GNU/Linux-System niemals einfach aus! Sie riskieren in diesem Fall einen Verlust Ihrer Daten!

Wenn Sie Ihren Computer zu Hause benutzen, möchten Sie ihn vielleicht nachts abschalten (eigentlich schaltet kein richtiger Linux-Fan seinen Computer jemals ab, aber trotzdem wollen wir diesen Ausnahmefall kurz besprechen...).

Es ist eine sehr schlechte Idee, einen Linux-Computer nach der Arbeit einfach auszuschalten oder die RESET-Taste zu drücken. Der Linux-Kernel hat, um die Performance zu erhöhen, einen internen Festplatten-Cache. Das bedeutet, dass Informationen temporär im Arbeitsspeicher (RAM) des Computers abgelegt werden, bevor sie auf der Festplatte gespeichert werden. Dies beschleunigt viele Aktionen stark. Periodisch wird der Inhalt dieses Puffers auf die Festplatte übertragen. Dies können Sie auch selbst durch das Kommando `sync` erreichen.

Um Ihren Rechner ordnungsgemäß herunterzufahren, benutzen Sie bitte das Kommando `reboot` (als Administrator - „root") oder drücken die Tastenkombination **STRG+ALT+ENTF** gleichzeitig. Debian GNU/Linux wird nun unmittelbar alle Programme beenden und den Rechner neu starten.

Um den Rechner abzuschalten, müssen Sie Administrator (root) sein. Benutzen Sie das Kommando shutdown -h now. Wenn Sie die Zeile System halted, it's safe to turn off the computer. sehen, können Sie den Rechner ausschalten.

Bei neueren Computern mit Advanced Power Management-Unterstützung (APM) und einem Kernel, der dies unterstützt, schaltet sich der Rechner selbsttätig ab.

Sie können aber auch als normaler Benutzer den Rechner jederzeit mit der Tastenkombination **STRG+ALT+ENTF** herunterfahren, wenn der Administrator dies nicht deaktiviert hat (in /etc/inittab). Keine Angst, das System wird auch so korrekt heruntergefahren, es gehen keine Daten verloren.

Die Datei /etc/inittab beschreibt die Prozesse, die beim Systemstart und im normalen Betrieb ausgeführt werden. Ein Eintrag in dieser Datei ist wie folgt aufgebaut:

```
id:runlevels:action:process
```

Zeilen, die mit dem Zeichen # beginnen, sind Kommentare und werden nicht ausgewertet. Die Werte in den einzelnen Feldern einer Datei haben die folgende Bedeutung

id

Ist eine eindeutige Zeichenkette mit der maximalen Länge von vier Zeichen.

runlevels

Beschreibt, in welchen Runleveln der Prozess gestartet werden soll. Hier können mehrere Runlevel, beispielsweise 1234, angegeben werden.

action

Die auszuführende Aktion. Die Aktionen werden weiter unten im Detail beschrieben.

prozess

Das auszuführende Kommando.

Der Eintrag im Feld „action“ steuert, auf welche Weise ein Prozess gestartet wird. Das Feld kann mit den im folgenden beschriebenen Werten belegt werden.

respawn

Der Prozess wird neu gestartet, falls das Programm beendet wird.

wait

Der Prozess wird beim Wechsel in den angegebenen Runlevel gestartet, und init wartet, bis der Prozess beendet wird.

once

Der Prozess wird einmalig gestartet, wenn der Runlevel erreicht wird.

boot

Der Prozess wird beim Systemstart gestartet, das Feld „runlevels“ wird ignoriert.

bootwait

Der Prozess wird beim Systemstart gestartet, init wartet, bis der Prozess beendet wird. Das Feld „runlevels“ wird ignoriert.

off

Dieser Eintrag wird ignoriert.

ondemand

Wird bei dem entsprechenden „ondemand“-Runlevel ausgeführt. Der Runlevel wird nicht gewechselt. „ondemand“-Runlevel sind „a“, „b“ und „c“.

initdefault

Beschreibt den normalen Runlevel eines Systems, Debian verwendet hier den Runlevel 2. Ist kein Runlevel angegeben, so wird beim Systemstart nach dem gewünschten Runlevel gefragt.

sysinit

Dieser Eintrag wird beim Systemstart vor Einträgen mit der Kennung „boot“ oder „bootwait“ ausgeführt. Das Feld „runlevels“ wird nicht ausgewertet.

powerwait

Unterbrechungsfreie Stromversorgung. Der Prozess wird gestartet, wenn die Netzspannung unterbrochen wird (dies setzt natürlich eine USV - Unterbrechnungsfreie Stromversorgung - voraus). init wird über den Ausfall durch ein Programm informiert, das die USV überwacht, und wartet, bis der Prozess beendet ist.

powerfail

Wie „powerwait“, jedoch wird nicht auf die Beendigung des Prozesses gewartet.

powerokwait

Dieser Prozess wird ausgeführt, wenn die Netzspannung wieder verfügbar ist.

powerfailnow

Geht die Ladung der USV soweit zur Neige, dass nur noch für kurze Zeit der Betrieb aufrecht erhalten werden kann, so wird dieser Eintrag ausgeführt und das System heruntergefahren.

ctrlaltdel

Wird die Tastenkombination **STRG+ALT+ENTF** gedrückt, so führt dies zum Herunterfahren das Systems, ähnlich wie man es von anderen Betriebssystemen kennt. Für ein Serversystem sollte dieser Eintrag deaktiviert werden, um zu verhindern, dass das System „versehentlich“ heruntergefahren wird. Ansonsten bietet dieser Eintrag eine Möglichkeit, das System ohne Administrator- Passwort herunterzufahren.

kbrequest

Hierüber können spezielle Tastenkombinationen an init weitergereicht werden. Die Dokumentation hierzu findet sich im Paket kbd.

Hier einige Beispiele für Einträge in der Datei /etc/inittab. Zunächst der Eintrag für den Runlevel, in dem das System gestartet werden soll:

```
id:2:initdefault:
```

Debian verwendet hier den Runlevel 2, dies weicht von einigen anderen Distributionen ab. Mit folgendem Eintrag wird festgelegt, welche Aktion auszuführen ist, wenn das System im Single-User-Modus gestartet wird.

```
~~:S:wait:/sbin/sulogin
```

Die Runlevel 0 bis 6 werden über die folgenden Zeilen gesteuert, genau genommen wird in jedem Fall das Skript /etc/init.d/rc mit dem Runlevel als Option aufgerufen.

```
l0:0:wait:/etc/init.d/rc 0 l1:1:wait:/etc/init.d/rc 1 l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3 l4:4:wait:/etc/init.d/rc 4 l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6
```

Das nächste Beispiel aktiviert die virtuellen Konsolen. Debian bietet in den Runleveln 2 und 3 sechs virtuelle Konsolen, in den Runleveln 4 und 5 ist nur die erste Konsole aktiv.

```
1:2345:respawn:/sbin/getty 38400 tty1 2:23:respawn:/sbin/getty 38400
tty2 3:23:respawn:/sbin/getty 38400 tty3 4:23:respawn:/sbin/getty
38400 tty4 5:23:respawn:/sbin/getty 38400 tty5
6:23:respawn:/sbin/getty 38400 tty6
```

Die Datei /etc/inittab auf einem Debian System ist gut kommentiert, so dass mit Hilfe dieses Abschnittes und der Kommentare weitere Anpassungen leicht fallen werden. Aber Achtung: bitte jede Änderung gut prüfen, da Probleme zu einem Systemzustand führen können, in dem beispielsweise ein Login nicht möglich ist.

Mitunter kann es vorkommen, dass init nicht in der Lage ist, ein Programm zu starten. Die Ursache kann ein fehlerhafter Eintrag in der Datei /etc/inittab, oder auch ein fehlerhaftes Programm sein. Bemerkbar macht sich dies durch die Ausgabe der Zeile

```
init: Id "D1" respawning to fast: disabled for 5 Minutes
```

wobei „D1“ hier beispielhaft für den entsprechenden Eintrag in der Datei inittab steht. Da init nicht in der Lage ist, das Programm zu starten, wird der entsprechende Eintrag für die nächsten 5 Minuten ignoriert, um die Belastung des Systems nicht unnötig in die Höhe zu treiben.

3.8 Systemstart und -stop bei Debian

◁ 3.7 Virtuelle Konsolen △ 3.9 Kommandozeile und Dokumentation ▷

3.9 Kommandozeile und Dokumentation

3.8 Systemstart und -stop bei Debian 3.10 Befehle auf der Kommandozeile wiederholen und ändern

Nun beschäftigen wir uns mit etwas umfangreicheren Beispielen.

Eine minimale Kommandozeile enthält lediglich einen einzigen Befehl ohne Parameter, zum Beispiel whoami. Aber auch dies lässt sich noch ausbauen, geben Sie einfach mal man whoami ein.

Der Befehl man ruft die Benutzungsanleitung („manual page“) für das Programm whoami auf. Mit der Taste **SPACE** können Sie in der Anleitung weiterblättern, die Taste q beendet das Programm man.

Hier ein noch erweitertes Beispiel:

```
man -k Postscript
```

Dieses Kommando besteht aus 3 Teilen: Zuerst steht der eigentliche Name des Kommandos, man, gefolgt von einer so genannten Option, hier -k, abschließend dann das Argument Postscript.

Optionen verändern das Verhalten eines Programms. In den meisten Fällen werden Optionen mit dem Zeichen „ - “ eingeleitet. Die GNU-Programme kennen außerdem eine ausführliche Form der Optionen; für die Option -k des Befehls „man“ wäre dies dann --apropos.

Probieren Sie dies nun einmal mit dem Befehl man, gefolgt von der Option -h, und einmal in der ausführlichen Form mit der Option -help aus. Die Ergebnisse sind gleich.

Drucken von Manpages

Dem Kommando man kann die Option -t sowie die gewünschte Manpage übergeben werden. Dies führt zur Ausgabe einer Postscript-Datei, leider auf der Standard-Ausgabe. Wird die Ausgabe direkt an den Drucker geschickt, so werden die Seiten im A4-Format ausgegeben. Um 2 Manual-Seiten auf einer A4-Seite auszugeben, kann das Kommando psnup zwischengeschaltet werden. Die komplette Kommandozeile sieht dann wie folgt aus

```
man -t whoami | psnup -2 | lpr
```

Eine Vorschau des Ergebnisses kann unter einer grafischen Benutzeroberfläche erzeugt werden, wenn das Kommando lpr durch gv - ersetzt wird.

Jedes Kommando hat seine eigenen Optionen. Es wird versucht, diese so weit wie möglich zu vereinheitlichen, so dass Sie die Optionen -help und --version bei allen GNU-Programmen antreffen sollten. Probieren Sie das ruhig einmal mit einigen verschiedenen Kommandos aus.

Aus historischen Gründen gibt es einige - manchmal geradezu bizarre - Abweichungen, die sich bis heute erhalten haben. So ist es z.B. möglich, bei den Kommandos tar oder ps das Zeichen „ - “ vor den Optionen einfach wegzulassen.

Alle Zeichen, die nicht zu einer Option gehören und kein Kommandoname sind, nennt man Argumente. Argumente können verschiedene Zwecke erfüllen. Meistens handelt es sich um Namen von Dateien, die mit dem entsprechenden Kommando bearbeitet werden sollen. In dem oben genannten Beispiel (man -k Postscript) ist Postscript das Wort, das vom Kommando man gesucht werden soll. Es wird dann nicht die Anleitung zum Programm Postscript gesucht, sondern es wird in allen Anleitungen nach dem Wort Postscript gesucht, und es werden alle Namen der entsprechenden Anleitungen angezeigt. Im Beispiel von man whoami ist man das Kommando und whoami das zu suchende Argument.

Wenn Sie nur wenige Programme auf Ihrem System installiert haben, sehen Sie im obigen Beispiel wenige Suchergebnisse oder sogar nur die Meldung: Postscript: nothing appropriate.

3.9 Kommandozeile und Dokumentation

3.8 Systemstart und -stop bei Debian 3.10 Befehle auf der Kommandozeile wiederholen und ändern

3.10 Befehle auf der Kommandozeile wiederholen und ändern

3.9 Kommandozeile und Dokumentation 3.11 Dateien und Verzeichnisse

3.10.1 Beschreibung der Kommandozeile

Alles, was Sie hinter einem Shell-Prompt eingeben, ist in irgendeiner Form Teil eines Kommandos. Die bei Debian GNU/Linux standardmäßig genutzte Shell (bash) hat verschiedene Funktionen, um das Eingeben und nachträgliche Ändern von Befehlszeilen zu erleichtern.

Sie können bereits eingegebene Kommandos wiederholen oder leicht verändern, um sie dann auszuführen. Die Befehle werden in der so genannten „history“ gespeichert. Probieren Sie es aus: Führen Sie irgendein Kommando, zum Beispiel whoami aus; drücken Sie dann die Taste **PFEIL-OBEN**. Der letzte Befehl erscheint wieder am Shell-Prompt. Sie brauchen nun nur noch **RETURN** zu drücken, um ihn noch einmal auszuführen.

Doppelte Zeilen in „history“ vermeiden

Seit der Version 3.0 der Bash ist es möglich, doppelte Einträge in der Datei .bash_history zu vermeiden. Hier ist die Zeile

```
export HISTCONTROL=ignoredups:erasedups
```

in die datei .bashrc einzutragen.

Wenn Sie einige Befehle eingegeben haben, können Sie mit den Tasten **PFEIL-OBEN** und **PFEIL-UNTEN** diese Befehle noch einmal anzeigen lassen. Sie können so Befehle mehrfach ausführen oder Tippfehler nachträglich korrigieren, ohne die ganze Zeile nochmals eingeben zu müssen.

Am einfachsten können Sie den Cursor in der Befehlszeile mit den Tasten **PFEIL-LINKS** und **PFEIL-RECHTS** bewegen. Tippen Sie einfach mal einen Befehl, z.B. whoami, ein. Gehen Sie nun mit dem Cursor an die Stelle mit dem Tippfehler, und löschen Sie mit **BACKSPACE** oder **DELETE** die gewünschten Zeichen. Geben Sie das noch fehlende Zeichen ein und drücken Sie **RETURN**.

Es gibt noch viele leistungsstarke Funktionen in der Shell. Eine detaillierte Beschreibung bekommen Sie mit dem Befehl man bash. Die komplette Dokumentation finden Sie unter /usr/share/doc/bash/.

An dieser Stelle möchten wir Ihnen nur einige häufig benutzte Funktionen vorstellen. Drücken Sie **STRG+a**, der Cursor springt damit an den Anfang der Zeile. **STRG+k** löscht von der aktuellen Position bis zum Zeilenende die komplette Eingabe (**k** für „kill"). Probieren Sie dies mitten in einer längeren Zeile einmal aus. Die Kombination **STRG+a**, dann **STRG+k** löscht die komplette Zeile. Die gelöschten Zeichen werden, unabhängig von der Länge, „gespeichert" und können mit **STRG+y** wieder an jeder Stelle eingefügt werden. Und zu guter Letzt: **STRG+e** bringt den Cursor ans Zeilenende.

Spielen Sie einfach mal ein wenig mit diesen Funktionen herum: Sie werden schnell merken, dass sich mit ihnen sehr effizient arbeiten lässt!

Die Kommandos, die Sie an einem Shell-Prompt eingeben können, folgen einer bestimmten Syntax. Wenn Sie beispielsweise man man eingeben, erhalten Sie die Anleitung (Manpage) zu dem Kommando man. In dieser Anleitung finden Sie weitere Beschreibungen zum Kommando man und zu den Optionen dieses Kommandos. Beispielsweise finden Sie dort:

```
man -k [-M path] keyword ...
```

Optionen, die in den eckigen Klammern ([]) stehen, sind nicht obligatorisch und können ausgelassen werden. Sie können also den Befehl man auch ohne die Option -M benutzen. Wenn Sie diese aber doch einsetzen, müssen Sie als Parameter den passenden Pfad angeben. Für die Option -k müssen Sie einen Suchbegriff (keyword) angeben. Die drei Punkte (...) bedeuten, dass Sie nach mehreren Begriffen suchen können, die dann durch Leerzeichen getrennt sein müssen.

Hier noch ein Beispiel für eine etwas komplexere Beschreibung:

```
man [-c|-w|-tZT device] [-adhu7V] [-m system[,...]] [-L locale]
[-p string] [-M path] [-P pager] [-r prompt] [-S list] [-e extension]
[[section] page ...] ...
```

Sie müssen nicht jede kleine Option verstehen - wichtig ist das Prinzip.

Neu ist in dieser Beschreibung das Zeichen |, es steht für „oder" - nicht zu verwechseln mit der Verwendung dieses Zeichens in der Shell, hier ist nur die Manpage gemeint. Sie

können also eine der Optionen -c, -w oder -tZT benutzen, letztere zusammen mit einem Device als Argument.

Gruppen von Optionen wie -adhu7V bedeuten, dass Sie eine oder mehrere dieser Optionen gleichzeitig nutzen können. Dabei ist es nicht selbstverständlich, dass alle möglichen Kombinationen auch Sinn machen oder gar funktionieren. Lesen Sie dazu die komplette Anleitung zu man.

Beachten Sie weiterhin, wie die eckigen Klammern verschachtelt sind: [[section] page]. Dies bedeutet: Wenn Sie eine section angeben, müssen Sie auch eine page angeben.

3.10 Befehle auf der Kommandozeile wiederholen und ändern

3.9 Kommandozeile und Dokumentation 3.11 Dateien und Verzeichnisse

3.11 Dateien und Verzeichnisse

3.10 Befehle auf der Kommandozeile wiederholen und ändern
3.12 Gruppen und Zugriffsrechte

Dateien auf einer Festplatte dienen zur Organisation und Speicherung von Daten, ähnlich einem Blatt Papier. Dateien können in Verzeichnissen (oder „Schubladen“) geordnet sein. Im Folgenden finden Sie einige Informationen über die Organisation von Dateien und Verzeichnissen bei Debian GNU/Linux.

Das Zeichen / repräsentiert das so genannte „root“-Verzeichnis. Alle weiteren Dateien und Verzeichnisse sind hier angeordnet. Wenn Sie vorher schon mal mit einem DOS- oder Windows-System gearbeitet haben, entspricht dies „in etwa“ dem Laufwerk C:. Machen Sie sich aber ab sofort mit dem Gedanken vertraut, dass es unter Linux keine Laufwerksbuchstaben gibt! Unter Linux finden Sie alle Laufwerke (Festplatten, CD-ROMs, Disketten...) unterhalb des „root“-Verzeichnisses (/) in einem Verzeichnis.

Beispielsweise stellt /home/fr das User-Verzeichnis des Benutzers „fr“ dar. Unterhalb des „root“-Verzeichnisses findet sich auf jedem Debian GNU/Linux-System das Verzeichnis home. In diesem befinden sich weitere Unterverzeichnisse. Die Namensgebung dieser Verzeichnisse ist identisch mit den Login-Namen, die beim Einrichten neuer Benutzerkonten vergeben werden.

/etc/X11/XF86Config ist die Konfigurationsdatei für das X Window-System.

Wichtig: Linux unterscheidet Groß- und Kleinschreibung bei den Pfaden und Dateinamen.

Die Verzeichnisse sind in einer Struktur ähnlich einem Baum angeordnet. Vom „root“-Verzeichnis (/) verzweigt alles zu den weiteren Verzeichnissen. Unterhalb von / finden sich folgende Dateien und Verzeichnisse:

```
/ |-- System.map |-- bin |-- boot |-- cdrom |-- dev |-- etc |-- floppy |-- home |-- initrd |-- lib |-- lost+found |-- mnt |-- proc |-- root |-- sbin |-- tmp |-- usr |-- var |-- vmlinuz
```

Hier als Beispiel die weitere Verzweigung unterhalb von /usr (Sie finden hier hauptsächlich Programme, die von allen Benutzern ausgeführt werden können):

```
/usr/ |-- X11R6 |-- bin |-- dict |-- doc |-- games |-- i486-linuxlibc1 |-- include |-- info |-- lib |-- local |-- man |-- openwin -> X11R6 |-- sbin |-- share |-- src
```

Bis hierhin sollten Sie auf Ihrem System (je nachdem, welche Pakete Sie installiert haben) in etwa die gleiche Struktur vorfinden. Deutliche Unterschiede von System zu System

finden sich unterhalb von /home. Die dortige Struktur ist abhängig von den Benutzerkonten, die auf dem jeweiligen System eingerichtet wurden:

```
/home |-- fr |-- ftp |-- mw
```

Diese Verzeichnisse stellen die so genannten „Heimat-Verzeichnisse“ der Benutzer dar: Nach dem Anmelden am System „befindet“ sich jeder Benutzer in seinem privaten Verzeichnis unterhalb von /home. In diesem Beispiel existieren die User fr und mw. Das Verzeichnis ftp wurde im Laufe der Installation des FTP-Servers auf diesem Rechner eingerichtet. Dies stellt in diesem Sinne keinen eigentlichen Benutzer dar. Abweichend davon finden Sie das Heimat-Verzeichnis des Administrators (root) unterhalb von / als /root/. So können Sie für alle Benutzer des Systems die Heimat-Verzeichnisse auf einer eigenen Partition halten. Für den Fall, dass das System beim Starten in einen instabilen Zustand gerät und die Partition mit den Heimat-Verzeichnissen der Benutzer nicht in das Dateisystem einhängen kann, besteht trotzdem für den Administrator die Möglichkeit, auf sein Heimat-Verzeichnis zuzugreifen.

Weiterhin verwendet Linux zur Verwaltung dieser Geräte keine Buchstaben (wie unter DOS). Der Verzeichnisbaum stellt eine Abstraktion der vorhandenen Hardware dar. Sie können die Verzeichnisse benutzen, ohne Kenntnis von der eigentlichen Hardware zu haben: Alle Dateien Ihres Systems können auf einer einzigen Festplatte liegen oder auf vielen verschiedenen, einige davon in Ihrem Rechner, andere in anderen Rechnern irgendwo im Netzwerk.

Keine Panik, wenn Sie dies jetzt nicht völlig verstehen: Die Idee dahinter ist anders als bei anderen Betriebssystemen. Es reicht, wenn Sie sich merken, dass es keine Laufwerksbuchstaben gibt. Laufwerke - so wie Sie sie kennen - tauchen innerhalb des Verzeichnisbaumes auf, beispielsweise als /cdrom oder /floppy, wobei diese auch an jeder anderen Stelle liegen können (häufig unterhalb von /mnt).

3.11 Dateien und Verzeichnisse

3.10 Befehle auf der Kommandozeile wiederholen und ändern 3.12 Gruppen und Zugriffsrechte

3.12 Gruppen und Zugriffsrechte

3.11 Dateien und Verzeichnisse 3.13 Orientierung innerhalb von Debian

3.12.1 Gruppen

3.12.2 Zugriffsrechte

Unix-Betriebssysteme und damit auch Debian GNU/Linux sind dafür ausgelegt, dass mehrere Benutzer zur gleichen Zeit am System arbeiten können. Dabei müssen bestimmte private Dateien vor anderen Benutzern geschützt werden, aber auch Systemdateien müssen vor den Benutzern geschützt werden. Sie können dies sehr leicht selbst überprüfen:

Melden Sie sich mit Ihrem Benutzernamen am System an (benutzen Sie nicht den Zugang des Administrator (root)!) und geben Sie das Kommando `rm /etc/resolv.conf` ein:

```
bash-2.03$ rm /etc/resolv.conf rm: remove write-protected file
`/etc/resolv.conf'? y rm: cannot unlink `/etc/resolv.conf': Permission
denied
```

Das System schützt diese Datei vor Veränderungen durch andere Benutzer als dem Administrator. Wenn jeder Benutzer Veränderungen an wichtigen Systemdateien vornehmen dürfte, würde dies schnell zu Problemen führen. Sehen wir uns die Datei einmal etwas näher an:

Geben Sie nun das Kommando `ls -l /etc/resolv.conf` ein. Sie bekommen diese Ausgabe:

```
-rw-r--r-- 1 root root 119 Nov 02 1999 /etc/resolv.conf
```

Die Option `-l` des Kommandos `ls` gibt den Dateinamen sowie viele weitere Informationen zu der Datei aus. Diese Informationen sind ziemlich einfach zu verstehen: Die Größe der Datei ist 119 Byte, die Datei wurde zuletzt am 02. November 1999 geändert, und der Dateiname ist `/etc/resolv.conf`. Weiter links wird die Sache etwas komplizierter...

Kurz und knapp: `-rw-r--r--` steht für die eigentlichen Zugriffsrechte der Datei, die `1` steht für die Anzahl der (hard-)Links auf diese Datei (oder die Anzahl der Dateien in einem Verzeichnis), und `root root` bezeichnet den Besitzer sowie die Gruppe, zu der die Datei gehört.

Doch nun etwas ausführlicher...

Jede Datei auf Ihrem Debian GNU/Linux-System hat zwei Eigentümer: einen User und eine Gruppe. Das oben angeführte Beispiel ist da etwas verwirrend, es gibt sowohl einen User als auch eine Gruppe root. Gruppen sind, wie auch im echten Leben, Ansammlungen von Personen, sprich: Benutzern auf einem System. Diese Mitglieder einer Gruppe können gemeinsamen Zugriff auf bestimmte Dateien haben, beispielsweise auf alle Dateien unterhalb von /var/www/projekte/debian/, wenn sie gemeinsam an den Webseiten zu einem Debian-Projekt arbeiten sollen. Sie können auch beispielsweise bestimmte Benutzer der Gruppe dialout (bedeutet so viel wie „rauswählen“) zuordnen, damit diese per Modem eine Verbindung ins Netz herstellen können.

Das Kommando groups zeigt Ihnen an, zu welchen Gruppen Sie gehören. Dies ist abhängig von dem Benutzernamen, mit dem Sie sich am System angemeldet haben.

Sehen Sie sich nun die Datei /etc/group an; benutzen Sie hierzu beispielsweise das Kommando more (more /etc/group). Beachten Sie die Gruppe root (in dieser sollte als einziger der Benutzer root eingetragen sein) sowie Ihre eigene Gruppe (auch hier sollten nur Sie eingetragen sein). Es gibt einige weitere Gruppen in dieser Datei, beispielsweise dialout (siehe oben), floppy (diese Benutzer können auf das Diskettenlaufwerk zugreifen) und andere. Nach der Installation sind keine weiteren Benutzer in den verschiedenen Gruppen aufgeführt; diese hinzuzufügen ist die Aufgabe des Systemverwalters, also Ihre ;-). Hier sehen Sie ein Beispiel für eine veränderte Datei /etc/group aus einem laufenden System:

```
root:x:0: daemon:x:1:fr bin:x:2: sys:x:3: adm:x:4: tty:x:5: disk:x:6:
lp:x:7:lp mail:x:8:fr,mw news:x:9: uucp:x:10: proxy:x:13: kmem:x:15:
dialout:x:20:fr fax:x:21:fr voice:x:22:fr cdrom:x:24:fr floppy:x:25:fr
tape:x:26:fr sudo:x:27:fr audio:x:29:fr dip:x:30:
majordom:x:31:majordom postgres:x:32: www-data:x:33:fr
backup:x:34:fr msql:x:36:fr operator:x:37:fr list:x:38:fr irc:x:39:fr
src:x:40:fr gnats:x:41: shadow:x:42: utmp:x:43:telnetd video:*:44:
staff:x:50:fr games:x:60:fr qmail:x:70: users:x:100:fr telnetd:x:101:
fr:x:1000: mw:x:1001: fr2:x:1003: nogroup:x:65534: mysql:x:102:
```

Weitere Informationen zu dieser Datei bekommen Sie mit dem Kommando man group.

Mit dem Kommando ls -l /home können Sie sich einen Überblick über die Stammverzeichnisse aller Benutzer auf dem System verschaffen. Jedes Verzeichnis sollte auch dem dazugehörigen Benutzer gehören. Wenn Sie das System neu installiert haben, werden Sie der einzige Benutzer sein. Deshalb auch hier ein Beispiel aus der Praxis mit wenigen Benutzern:

```
bash-2.03$ ls -l /home/ total 8 drwxr-sr-x   65 fr       fr           5120 Jan  1
02:48 fr drwxr-sr-x    5 mw       mw           1024 Jan  4 08:27 mw
```

3.12.2.1 Einige Beispiele

Neben dem Besitzer und der Gruppe, zu denen eine Datei gehört, verfügt jede Datei auch über Zugriffsrechte, über die festgelegt wird, wer diese Datei lesen, schreiben oder ausführen darf. Es gibt noch weitere Details, die wir aber übergehen wollen.

Wie Sie schon vorher gesehen haben, werden die Zugriffsrechte bei dem Kommando ls -l ganz links an den ersten zehn Stellen angezeigt. Die erste Stelle hat nicht direkt etwas mit den Zugriffsrechten zu tun, sie zeigt vielmehr den Dateityp an. Ein „ - “ steht für eine normale Datei, ein d kennzeichnet ein Verzeichnis, und ein l steht für einen Link.

Die weiteren neun Stellen lassen sich in drei Gruppen teilen. Dies sind von links nach rechts: der Besitzer der Datei (owner), die Gruppe (group) und schließlich die Allgemeinheit (world). Jeder dieser drei Gruppen gehören drei dieser neun Stellen. Jede dieser drei Stellen steht für r lesen (Read), w schreiben (Write) und x ausführen (eXecute).

Im Detail bedeuten die drei Buchstaben r, w und x Folgendes:

r - lesen: Bei Dateien kann der Inhalt der Datei gelesen werden. Bei Verzeichnissen kann man den Inhalt des Verzeichnisses auflisten lassen.

w - schreiben: Bei Dateien kann diese Datei verändert und gespeichert werden. Bei Verzeichnissen können neue Dateien angelegt und bereits bestehende Dateien gelöscht werden.

x - ausführen: Dateien können als Kommando ausgeführt werden. Dies macht nur Sinn, wenn diese Datei wirklich ein Kommando darstellt. Sie können eine Grafik ausführbar machen, es ergibt aber keinen Sinn. Da Verzeichnisse nicht ausgeführt werden können, bedeutet hier ein gesetztes x, dass Sie in dieses Verzeichnis wechseln können. Um also in einem Verzeichnis mit Dateien arbeiten zu können, benötigen Sie die Kombination x und r sowie gegebenenfalls auch w.

Für Verzeichnisse ist dies alles ein wenig verwirrend, daher hier einige Beispiele:

r-- Eigentümer, Gruppe oder andere können den Inhalt dieses Verzeichnisses auflisten. Die Dateien selbst können in dem Verzeichnis gelesen, gelöscht oder verändert werden, abhängig von den eigenen Zugriffsrechten.

r-x Dieser Modus erlaubt das Auflisten der Dateien in dem Verzeichnis und gibt den Zugriff auf die Dateien frei. Sie können allerdings keine neuen Dateien anlegen oder bestehende Dateien löschen. Das Ansehen und Verändern von Dateien ist erlaubt; Programme können ausgeführt werden, wenn dies von den Rechten der Dateien selbst her erlaubt ist.

--x Sie können auf die Dateien in dem Verzeichnis zugreifen, diese aber nicht auflisten. Sie müssen also wissen, welche Dateien sich in dem Verzeichnis befinden, um auf diese zugreifen zu können. Wenn es sich um ein Verzeichnis handelt, so kann in dieses gewechselt werden.

rwx Sie können alles mit den Dateien anstellen, solange die Rechte der Dateien selbst dies zulassen.

Daraus folgen einige interessante Tatsachen, die Sie beachten sollten:

Schreibrechte auf einem Verzeichnis entscheiden darüber, ob Sie eine Datei in einem Verzeichnis löschen dürfen. Eine Datei, deren Rechte auf Nur-lesen gesetzt sind, kann gelöscht werden, wenn Sie die nötigen Rechte haben, um in diesem Verzeichnis zu schreiben! Weiterhin können Sie eine Datei in einem Nur-lesen-Verzeichnis nicht löschen, auch wenn Sie die nötigen Zugriffsrechte auf die Datei selbst haben.

Dies bedeutet auch, dass Sie, wenn Sie der Besitzer eines Verzeichnisses sind, auch die Dateien darin löschen können, sogar wenn diese dem Administrator (root) gehören.

Zugriffsrechte auf ein Verzeichnis haben also auch direkten Einfluss auf die Dateien in diesem Verzeichnis. An dieser Stelle kommen die Zugriffsrechte auf Dateien ins Spiel. Wenn Sie keinen Zugriff auf das Verzeichnis haben, spielen auch die Rechte an den Dateien für Sie keine Rolle, Sie kommen ja ohnehin nicht an die Dateien...

Um die Zugriffsrechte von Dateien und Verzeichnissen zu verändern, steht unter Debian GNU/Linux das Kommando chmod zur Verfügung. Spielen wir einmal ein wenig damit herum:

Erzeugen Sie zunächst eine neue Datei, beispielsweise mit dem Kommando touch testdatei. Das Kommando touch wird normalerweise dazu benutzt, die Datei mit einem aktuellen „Zeitstempel“ zu versehen. Wenn Sie jedoch einen Dateinamen angeben, der noch nicht existiert, so wird eine Datei mit diesem Namen neu angelegt Sie hat dann eine Größe von 0 Byte. Überprüfen Sie dies mit dem Kommando ls -l und werfen Sie einen Blick auf die Zugriffsrechte:

```
bash-2.03$ touch testdatei bash-2.03$ ls -l testdatei -rw-r--r--    1 fr
fr             0 Jan 19 18:15 testdatei
```

Bei Ihrem Versuch wird die Datei natürlich einen anderen Zeitstempel haben, und Benutzer- und Gruppenzugehörigkeit entsprechen Ihrem Loginnamen. Die Zugriffsrechte (-rw-r--r--) werden von Debian GNU/Linux automatisch für neue Dateien auf die gezeigten Werte gesetzt. Sie können diese Vorgabe mit dem Kommando umask ändern.

Sehen Sie sich zunächst die Manpage zu chmod mit dem Kommando man chmod an. Wir werden hier nicht auf jedes Detail eingehen, sondern an einigen Beispielen zeigen, wie sich chmod mit verschiedenen Parametern auswirkt.

Führen Sie das Kommando chmod u+x testdatei aus. Sehen Sie sich die Veränderung mit ls -l testdatei an. Es wurden Rechte zum Ausführen (x - execute) der Datei für den Besitzer (u - User) hinzugefügt (+ - Pluszeichen).

```
bash-2.03$ chmod u+x testdatei bash-2.03$ ls -l testdatei -rwxr--r--    1
fr       fr             0 Jan 19 18:15 testdatei
```

Ein solches Kommando können Sie beispielsweise auf ein selbst geschriebenes Shellskript oder Perl-Programm anwenden, damit es auch ausführbar ist.

Wenn Sie nun noch möchten, dass niemand außer Ihnen einen Blick in Ihre Arbeit werfen kann, so müssen Sie die Rechte zum Lesen der Datei für die Gruppe (g - Group) sowie alle anderen Benutzer (o - Other) entfernen (- - Minuszeichen). Sie können dies mit dem Kommando chmod go-r testdatei erreichen:

```
bash-2.03$ chmod go-r testdatei bash-2.03$ ls -l testdatei -rwx------    1
fr       fr             0 Jan 19 18:15 testdatei
```

Wie Sie gesehen haben, können Sie mit den Zeichen + (Plus) oder - (Minus) Rechte hinzufügen oder entfernen. Manchmal ist es damit etwas verwirrend, einen gewünschten Zustand herzustellen. Daher bietet chmod noch die Option = (Gleichheitszeichen), welche genau die angegebenen Rechte setzt und alle anderen löscht. Auch hier können Sie wieder die Buchstaben ugo (User, Group, Other) benutzen:

```
bash-2.03$ chmod ugo=rx testdatei bash-2.03$ ls -l testdatei -r-xr-xr-x
1 fr       fr             0 Jan 19 18:15 testdatei
```

Die Datei ist nun für jeden Benutzer lesbar und kann auch von jedem ausgeführt werden. Weiterhin kann keiner der Benutzer diese Datei schreiben.

Entfernen Sie nun die Rechte zum Ausführen der Datei für alle Benutzer (`chmod a-x testdatei`), bei einer Testdatei brauchen wir diese nicht.

Versuchen Sie einmal, die Datei zu löschen. Zur Erinnerung: Sie hatten vor kurzem die Datei mit dem Kommando `chmod ugo=rx testdatei` behandelt. Löschen Sie also die Datei mit dem Kommando `rm testdatei`:

```
bash-2.03$ rm testdatei
rm: schreibgeschützte Datei "testdatei" entfernen?
```

Wenn Sie die Umgebungsvariablen nicht passend gesetzt haben, wird Ihnen die Fehlermeldung in englischer Sprache präsentiert.

Da Sie (und auch alle anderen) keine Rechte haben, die Datei zu schreiben, fragt das Kommando `rm`, ob Sie diese Aktion wirklich durchführen wollen. Dies ist eine spezielle Funktion von `rm` und hat eigentlich wenig mit den Zugriffsrechten zu tun. Wenn Sie die Datei wirklich löschen möchten, können Sie die Frage bestätigen. Stören Sie sich nicht an der Ausgabe der Rechte an Zahlenform des Kommandos `rm`; Sie können die Bedeutung in der Manpage nachlesen.

3.12 Gruppen und Zugriffsrechte

3.11 Dateien und Verzeichnisse 3.13 Orientierung innerhalb von Debian

3.13 Orientierung innerhalb von Debian

3.12 Gruppen und Zugriffsrechte 3.14 Arbeiten mit Dateien - Mini-Workshop

3.13.1 Gerätedateien in /dev und ihre Bedeutung

Es gibt einige Unterschiede zwischen Debian GNU/Linux und anderen Distributionen. Selbst wenn Sie Unix und andere Linux-Distributionen bereits kennen, gibt es einige Dinge, die Sie wissen sollten, um Ihr System in einem gut zu wartenden Zustand zu halten. Dieser Abschnitt bietet einen Überblick über das System, um die Orientierung zu erleichtern.

Das wichtigste Konzept, das man verstehen muss, ist die Paketverwaltung von Debian. Im Wesentlichen müssen Sie akzeptieren, dass große Teile Ihres Systems unter der Kontrolle der Paketverwaltung stehen. Sie können nicht so ohne Weiteres „von Hand“ (zum Beispiel, wenn Sie eine eigene Version des Apache-Servers aus den Sourcen übersetzt haben) ein Paket aktualisieren. Diese Aufgabe nimmt Ihnen das Debian GNU/Linux-Paketsystem ab. Nutzen Sie dies, um Pakete zu aktualisieren. Folgende Bereiche stehen unter der Kontrolle dieses Paketsystems:

/usr (mit Ausnahme von /usr/local)

/var (Das Verzeichnis /var/local kann angelegt werden um eigene, variable Dateien abzulegen.)

/bin

/sbin

/lib

Wenn Sie zum Beispiel /usr/bin/perl ersetzen, weil Sie auf einer CD oder auf einem FTP-Server eine aktuellere Version als auf Ihren Debian GNU/Linux-CDs gefunden haben (kaum zu glauben, dass so etwas wirklich passiert...), wird das zunächst funktionieren.

Wenn Sie nun jedoch ein Paket mit dselect installieren, das ebenfalls irgendeine Perl-Version benötigt, so wird dselect immer die Version heranziehen, die auf Ihrem Installationspfad enthalten ist. Das heißt, dass dselect unter Umständen eine ältere Version installiert, obwohl Sie schon eine neuere Version übersetzt hatten. Bitte beachten Sie das!

Aktualisieren Sie jedoch Ihr Perl-Paket über dpkg/ dselect oder apt, dann wird die Datei durch die aus dem Debian Paket ersetzt. Erfahrene Anwender können dieses verhindern, indem sie das entsprechende Paket auf „hold“ (in der Paketauswahl von dselect mit der Taste =) setzen oder dpkg-divert benutzen.

Wenn Sie bestimmte Pakete durch eigene, modifizierte Versionen ersetzen wollen, sollten Sie sich intensiv mit dem Debian GNU/Linux-Paketmanagement befassen.

Eine der wichtigsten Vereinbarungen, an die Sie sich gewöhnen müssen, ist, dass sich alle Konfigurationsdateien unterhalb von /etc/ befinden. Hierbei ist es vielfach so, dass für einzelne Programme zusätzliche Verzeichnisse bei der Installation erzeugt werden. Meist geschieht dies für Programme, die über mehr als eine Konfigurationsdatei verfügen, beispielsweise für den Webserver apache.

Unter /usr/doc/ beziehungsweise /usr/share/doc/ finden Sie die Dokumentation zu den auf Ihrem System installierten Paketen. Zu vielen Paketen finden Sie in dem entsprechenden Verzeichnis eine Datei README.Debian (oder ähnlich), die die speziellen Anpassungen an Debian GNU/Linux beschreibt.

Weiterhin befinden sich in jedem Verzeichnis auch die jeweiligen Lizenzen zu den Paketen.

Unter Linux befinden sich im Verzeichnis /dev verschiedene besondere Dateien, die so genannten Gerätedateien (device files). Wie unter Unix üblich, wird auf Hardware- und System-Komponenten jeweils über Gerätedateien zugegriffen. Mit Hilfe einer solchen Datei kann vom jeweiligen Programm über einen speziellen internen Treiber das zugehörige Gerät beziehungsweise eine Systemkomponente angesprochen werden. Die Gerätedatei dient dafür als Schnittstelle. Aus Anwendersicht verhalten sich Gerätedateien oft anders als gewöhnliche Dateien. Die Bedeutung der wichtigsten Gerätedateien ist im Folgenden aufgeführt.

Es gibt sehr viele verschiedene CD-ROM-Laufwerke und Schnittstellen. Sollten Sie weder ein SCSI-CD-ROM-Laufwerk haben noch eines mit IDE-Schnittstelle, dann gibt es für Ihr CD-ROM-Laufwerk eine spezielle Gerätedatei, die bereits bei der Installation automatisch angelegt wurde. Der Name dieser Gerätedatei ist üblicherweise ein Kürzel des Namens Ihres CD-ROM-Laufwerks. Um den Zugriff auf das CD-ROM-Laufwerk zu vereinfachen, existiert der symbolische Link /dev/cdrom. Dieser Link wird bei der Installation ebenfalls angelegt und verweist auf die eigentliche Gerätedatei. Mit dem Kommando ls -l /dev/cdrom sehen Sie nach, auf welche Gerätedatei cdrom verweist.

```
fr@sushi:~$ ls -l /dev/cdrom  lrwxrwxrwx    1 root     root        3 Sep  1
16:06 /dev/cdrom -> hdc
```

Hier die wichtigsten Gerätedateien im Verzeichnis /dev/:

fd0 1. Diskettenlaufwerk fd1 2. Diskettenlaufwerk hda IDE-Festplatte / IDE-CD-ROM am 1. Anschluss Master hdb IDE-Festplatte / IDE-CD-ROM am 1. Anschluss Slave hdc IDE-Festplatte / IDE-CD-ROM am 2. Anschluss Master hdd IDE-Festplatte / IDE-CD-ROM am 2. Anschluss Slave hda1 1. Partition der ersten IDE-Festplatte hda15 15. Partition der ersten IDE-Festplatte sda SCSI-(Wechsel-)Festplatte, kleinste SCSI-ID sdb SCSI-(Wechsel-)Festplatte, nächstgrößere SCSI-ID sdc SCSI-(Wechsel-)Festplatte, nächstgrößere SCSI-ID sda1 1. Partition der ersten SCSI-(Wechsel-)Festplatte sda15 15. Partition der ersten SCSI-(Wechsel-)Festplatte scd0 1. SCSI-CD-ROM-Laufwerk scd1 2. SCSI-CD-ROM-Laufwerk cdrom Symbolischer Link auf das CD-ROM-Laufwerk mouse Symbolischer Link auf Maus-Gerätedatei ttyS0 1. Serielle Schnittstelle (COM1) ttyS1 2. Serielle Schnittstelle (COM2) lp0 1. Parallele Schnittstelle lp1 2. Parallele Schnittstelle null Hier können beliebig viel Daten hineinkopiert werden zero Hieraus können beliebig viel Nullen gelesen werden

Achtung: Die Daten bei DOS-formatierten Zip-Medien liegen auf Partition vier. Das heißt, dass Sie zum Mounten eines DOS-formatierten ZIP-Mediums Folgendes eingeben müssen:

```
mount -t vfat /dev/sdc4 /zip.
```

Dies gilt für den Fall, dass das ZIP-Laufwerk das dritte SCSI-Gerät für Wechselmedien ist und dass Sie das ZIP-Medium auf das Verzeichnis /zip mounten möchten.

3.13 Orientierung innerhalb von Debian

3.12 Gruppen und Zugriffsrechte 3.14 Arbeiten mit Dateien - Mini-Workshop

3.14 Arbeiten mit Dateien - Mini-Workshop

3.13 Orientierung innerhalb von Debian 3.15 Einige bash-Funktionen

Um mit Ihrem System arbeiten zu können, müssen Sie etwas über das Erzeugen, Verschieben, Umbenennen und Löschen von Dateien und Verzeichnissen erfahren.

Zunächst ist es jedoch wichtig zu wissen, an welcher Stelle des Verzeichnisbaumes man sich befindet. Wie schon beschrieben, „befinden“ Sie sich nach dem Anmelden am System in Ihrem Heimat-Verzeichnis. Sie können dies mit dem Kommando pwd überprüfen. Die Ausgabe sollte in etwa so aussehen:

```
bash-2.02$ pwd /home/fr
```

Wobei auf Ihrem System statt fr Ihr eigener Benutzername, mit dem Sie sich angemeldet haben, erscheint.

Dieses Kommando zeigt, wenn es ohne weitere Parameter verwendet wird, Dateien in dem aktuellen Verzeichnis an. Nach dem Anmelden an einem neu installierten Debian GNU/Linux-System befinden Sie sich in Ihrem Heimat-Verzeichnis, und dieses ist leer. Das Kommando ls wird also nichts anzeigen. (Das Verzeichnis ist nicht wirklich leer, Sie sehen lediglich mit dem Kommando ls ohne Parameter nicht die angelegten Dateien.)

Versteckte Dateien

Einige Dateien im Heimat-Verzeichnis eines jeden Benutzers werden beim Anlegen des Benutzerkontos durch den Administrator als versteckte Dateien angelegt. Bei diesen Dateien und Verzeichnissen handelt es sich meistens um Konfigurationsdateien, die nicht oder nur selten verändert werden und aus diesem Grund nicht ständig sichtbar sein müssen. Dies erhöht die Übersichtlichkeit im Heimat-Verzeichnis.

Um Dateien auf einem Unix-System zu „verstecken“, sind keine großartigen Tricks notwendig. Es reicht, die Datei umzubenennen, so dass diese mit einem Punkt beginnt. Dateien mit einem führenden Punkt im Dateinamen werden von den Kommandos nicht ohne Weiteres angezeigt. Wird dem Kommando ls die Option -a übergeben, so werden auch versteckte Dateien angezeigt.

Sie können aber auch das Kommando ls mit einem Pfad als Option aufrufen. Beispielsweise zeigt ls / das „root“-Verzeichnis des Systems an.

Mit diesem Kommando können Sie in ein anderes Verzeichnis (directory) wechseln. cd /tmp wechselt beispielsweise in das Verzeichnis für temporäre Dateien unterhalb von /.

Auf zwei Besonderheiten möchte ich an dieser Stelle eingehen, die nicht nur mit dem Kommando cd funktionieren, aber zum besseren Verständnis hier gut untergebracht sind.

Das Zeichen ~ steht als Abkürzung für den kompletten Pfad zu Ihrem privaten Heimat-Verzeichnis. cd mit der Option ~, also cd ~, wechselt ins Heimat-Verzeichnis.

Weiterhin möchte man häufig in ein in der Struktur höher gelegenes Verzeichnis wechseln. Sicher könnte man mit pwd nachsehen, wo man sich gerade befindet und dann dem Kommando cd den passenden Pfad übergeben. Als einfache Alternative steht aber das Kürzel .. für das übergeordnete Verzeichnis zur Verfügung. cd .. wechselt also in das unmittelbar übergeordnete Verzeichnis (beachten Sie das Leerzeichen).

Wurde in ein anderes Verzeichnis durch Angeben eines Pfadnamen gewechselt, so kann mit dem Kommando cd - in das letzte Verzeichnis gewechselt werden.

Eine weitere Abkürzung stellt . dar. Diese steht für das aktuelle Verzeichnis, in dem Sie sich gerade befinden, doch dazu kommen wir gleich in einem anderen Beispiel.

Mit diesem Kommando können Sie weitere Verzeichnisse anlegen. Wechseln Sie mit cd ~ in Ihr Heimat-Verzeichnis, und erzeugen Sie ein Verzeichnis test (mit mkdir test). Überprüfen Sie mit ls, ob es funktioniert hat. Erzeugen Sie nach Belieben einige weitere Verzeichnisse, auch unterhalb von test.

Mit diesem Kommando können Sie Kopien von Dateien erzeugen. Kopieren Sie die Datei /etc/profile mit cp /etc/profile . in Ihr Heimat-Verzeichnis (hier also das versprochene Beispiel mit nur einem Punkt). Prüfen Sie mit ls, ob sich in Ihrem Verzeichnis nun eine Datei namens profile befindet.

more profile zeigt Ihnen den Inhalt der Datei profile seitenweise auf der Konsole an. Sie können mit der **SPACE**-Taste seitenweise weiterblättern und mit der Taste q das Programm more wieder verlassen. more zeigt am unteren Bildschirmrand die aktuelle Position in der Datei in Prozent an. Am Ende einer Datei wird more automatisch beendet. more kann, wie die meisten anderen Programme auch, über Optionen auf der Kommandozeile gesteuert werden. Sie können diese Optionen auch in die Umgebungsvariable MORE schreiben; die Optionen werden dann bei jedem Aufruf von more benutzt.

Lesen Sie die Manual-Seite zu more, das Programm hat noch einige andere interessante Möglichkeiten.

Ansehen von gepackten Dateien

Sollten Sie auf eine gepackte Textdatei stoßen, so können Sie diese mit dem Kommando zmore ansehen.

Beispielsweise ist ein großer Teil der Dokumentation, die mit fast jedem Debian Paket installiert wird, komprimiert. In einem der Verzeichnisse unterhalb von /usr/share/doc/ wird man sicherlich fündig.

Wenn Ihnen die Optionen von more nicht ausreichen, sollten Sie einen Blick auf das Programm less werfen. Es verfügt über die gleichen Funktionen, wurde aber noch um einige nützliche Funktionen erweitert.

Zum Verschieben von Dateien benötigt das Kommando mv mindestens zwei Parameter: der erste Parameter ist die Quelldatei, der zweite die Zieldatei. Verschieben Sie die Datei profile mit mv profile test in das erstellte Verzeichnis test. mv dient aber auch zum Umbenennen von Dateien. Wechseln Sie in das Verzeichnis test, und benennen Sie die Datei profile in testdatei um (mv profile testdatei).

Weiterhin können Sie mit mv auch Verzeichnisse verschieben oder umbenennen, die Syntax unterscheidet sich dabei nicht, egal ob Sie mit Dateien oder Verzeichnissen arbeiten.

Beachten Sie, dass es mit mv nicht möglich ist, mehrere Dateien auf einmal umzubenennen. Hierzu bedarf es eines kleinen Skriptes oder anderer entsprechender Hilfsmittel.

Mit dem Kommando rm können Sie eine oder mehrere Dateien löschen. Im einfachsten Fall geben Sie den Namen der zu löschenden Datei an, also: rm ichwillweg.txt (Sollte diese nicht existieren, so können Sie sie mit dem Kommando touch ichwillweg.txt anlegen). Als Parameter kann dem Kommando rm eine Reihe von Dateinamen mitgegeben werden, diese werden durch Leerzeichen getrennt: rm ichwillweg.txt ichauch.txt metoo.asc removeme.txt done.sh; dieses Kommando löscht fünf

Dateien von der Festplatte. An dieser Stelle ein wichtiger Hinweis: Es gibt kein undelete unter Linux. Dateien, die Sie mit rm gelöscht haben, können Sie nicht zurückholen. (Es gibt Programme, die auch mit dem ext3-Dateisystem gelöschte Dateien zurückholen können. Dies ist aber momentan noch in der Entwicklung.)

Wenn Sie ein Verzeichnis inklusive aller darin enthaltenen Dateien löschen möchten, können Sie dies auch mit dem Kommando rm erledigen. Hierzu dient die Option -rf. Ein Beispiel: rm -rf /home/fr/test/ löscht im Home-Verzeichnis das Verzeichnis test mit allen Dateien und Unterverzeichnissen. Etwas, was Sie nicht ausprobieren sollten (man findet das manchmal, weil sich Leute einen Spaß daraus machen...). Ein weiteres Beispiel wäre Folgendes (nicht abtippen!!!): rm -rf / (nicht abtippen!!!). Wie schon beschrieben, stellt / das Startverzeichnis des gesamten Verzeichnisbaums dar. Sie würden also Ihr komplettes System von der Festplatte verbannen; Fallen Sie also nicht auf diesen kleinen Spaß herein!

Nun fehlt uns noch ein Kommando, um lediglich ein Verzeichnis, in dem sich keine Dateien befinden, zu entfernen. rmdir mit dem Verzeichnisnamen erledigt dies für uns. Natürlich lässt sich auch rm mit den entsprechenden Optionen dafür nutzen. Sie können selbst entscheiden, welches Kommando Sie benutzen wollen... Noch schnell ein Beispiel: rmdir test entfernt das Verzeichnis test im aktuellen Verzeichnis.

Namen von versteckten Dateien oder Verzeichnissen beginnen mit einem Punkt (.). Sie können das Kommando ls mit der Option -a dazu bringen, auch diese versteckten Dateien anzuzeigen. Sie können das sehr einfach in Ihrem Heimat-Verzeichnis ausprobieren; dort werden bei der Einrichtung eines neuen Benutzers und später durch verschiedene Programme diverse versteckte Dateien und Verzeichnisse angelegt.

Wechseln Sie in Ihr Heimat-Verzeichnis (mit dem Kommando cd) und sehen Sie sich den Inhalt des Verzeichnisses einmal an, inklusive der versteckten Dateien (mit dem Kommando ls -la). Sie sehen nun die „normalen“ Dateien sowie auch die versteckten Dateien. Dabei wird Ihnen vielleicht auffallen, dass es zwei etwas außergewöhnliche Dateien, nämlich . und .., gibt. Diese stellen das aktuelle Verzeichnis „.“, in dem Sie sich befinden, sowie das übergeordnete Verzeichnis „ .. “ dar. Wenn Sie das Kommando ls mit der Option -lA benutzen, werden diese beiden Dateien nicht mit angezeigt.

Der Grund für versteckte Dateien liegt nicht in der Geheimhaltung von Daten. Vielmehr ist es im täglichen Umgang mit dem System nicht sinnvoll, alle möglichen Dateien anzuzeigen, die sich in Ihrem Heimat-Verzeichnis befinden. Viele Programme legen dort auch individuelle Konfigurationsdateien ab. Es hat sich eingebürgert, diese Dateien oder

Verzeichnisse mit einem Punkt beginnen zu lassen, so dass diese nicht bei der normalen Arbeit mit Dateien stören.

Über diese Konfigurationsdateien in Ihrem Heimat-Verzeichnis können Sie das Verhalten oder Aussehen von Programmen verändern. Diese Änderungen sind nur wirksam, wenn Sie sich mit Ihrem Benutzernamen am System angemeldet haben. Systemweite Konfigurationsdateien finden Sie im Verzeichnis /etc/.

Um Dateien in Ihrem System zu finden, stehen Ihnen auf der Kommandozeile zwei Programme zur Verfügung: find und locate. Mit dem Programm find können Sie die Festplatte nach Dateien durchsuchen; dies kann je nach Größe der Festplatten einige Zeit dauern. find verfügt über sehr viele Parameter, die Sie in der Manpage nachlesen können.

```
bash-2.03$ find --help Usage: find [path...] [expression] default path is
the current directory; default expression is -print expression may consist
of: operators (decreasing precedence; -and is implicit where no others
are given):  ( EXPR ) ! EXPR -not EXPR EXPR1 -a EXPR2 EXPR1 -
and EXPR2        EXPR1 -o EXPR2 EXPR1 -or EXPR2 EXPR1 ,
EXPR2 options (always true): -daystart -depth -follow --help        -
maxdepth LEVELS -mindepth LEVELS -mount -noleaf --version -xdev
tests (N can be +N or -N or N): -amin N -anewer FILE -atime N -cmin
N        -cnewer FILE -ctime N -empty -false -fstype TYPE -gid N -
group NAME        -ilname PATTERN -iname PATTERN -inum N -
ipath PATTERN -iregex PATTERN        -links N -lname PATTERN -
mmin N -mtime N -name PATTERN -newer FILE        -nouser -
nogroup -path PATTERN -perm [+-]MODE -regex PATTERN        -
size N[bckw] -true -type [bcdpfls] -uid N -used N -user NAME        -
xtype [bcdpfls] actions: -exec COMMAND ; -fprint FILE -fprint0 FILE
-fprintf FILE FORMAT        -ok COMMAND ; -print -print0 -printf
FORMAT -prune -ls
```

Für den normalen Einsatz ist es allerdings ausreichend, wenn Sie sich folgendes Beispiel einprägen:

```
bash-2.02$ find / -name resolv.conf /etc/resolv.conf find:
/var/spool/cron/atjobs: Permission denied find: /var/spool/cron/atspool:
Permission denied find: /var/lib/xdm/authdir: Permission denied
```

Nach einiger Zeit hat find die Datei im Verzeichnis /etc/ gefunden. Für Verzeichnisse, auf die Sie keinen Zugriff haben, gibt find eine entsprechende Fehlermeldung aus. Direkt hinter dem Kommando find können Sie das Verzeichnis angeben, in dem mit der Suche begonnen werden soll. Im Beispiel wird das gesamte Dateisystem (/) durchsucht. Wenn Sie den Namen einer Datei nicht genau kennen, können Sie auch nur einen Teil des Namens angeben und den Rest mit den üblichen Wildcards ersetzen. Beachten Sie, dass jedes Kommando zuerst von der Shell interpretiert und dann ausgeführt wird. Sie müssen also beispielsweise den * vor der Shell „verstecken“, indem Sie das Zeichen \ voranstellen. Die Shell wird nun das folgende Zeichen ignorieren und direkt an das Kommando weiterreichen:

```
bash-2.02$ find / -name resol\* /etc/resolv.conf find:
/var/spool/cron/atjobs: Permission denied find: /var/spool/cron/atspool:
Permission denied find: /var/lib/xdm/authdir: Permission denied
```

Doppelte Dateien finden

find ist auch in der Lage, zusammen mit einigen anderen Werkzeugen aus dem Unix-Werkzeugkasten, doppelte Dateien auf der Festplatte zu ermitteln. Bei folgendem Beispiel wird ab der aktuellen Position im Dateisystem nach identischen Dateien gesucht. Hierzu wird von jeder Datei eine MD5-Checksumme erzeugt, die Liste wird sortiert, und es werden doppelte Einträge einmalig ausgegeben. Das Ergebnis wird in die Datei doppelte.asc geschrieben.

```
find . -exec md5sum {} 2>/dev/null \; | sort | uniq -w 32 -D > doppelte.asc
```

Soll dieser Vergleich über eine sehr große Anzahl von Dateien ausgeführt werden, so sollte statt dem Programm md5sum das schnellere cfv (versatile file checksum creator and verifier) eingesetzt werden. Dieses liegt als Debian Paket vor und muss gesondert installiert werden.

Der zweite Weg, um Dateien zu finden, bietet sich über das Programm locate. Dieses ist um einiges schneller beim Finden von Dateien, da es eine Datenbank benutzt, die einmal am Tag aktualisiert wird. Es ist also nicht notwendig, jedes Mal die komplette Festplatte zu durchsuchen. Allerdings funktioniert das Aktualisieren der Datenbank nur, wenn Ihr Rechner zu der Zeit in Betrieb ist, zu der auch diese Aktualisierung stattfinden soll. Da hierbei die komplette Festplatte durchsucht wird, kann der Vorgang einige Zeit dauern.

Sollten Sie Ihren Rechner nicht rund um die Uhr laufen lassen, so können Sie auch (als Administrator) die Datenbank von locate mit dem Kommando updatedb zu jeder anderen Zeit aktualisieren. Sie können aber auch die Zeit, zu der updatedb gestartet wird, in der Datei /etc/crontab Ihren Bedürfnissen anpassen.

Wenn Ihnen diese Änderungen zu kompliziert erscheinen, können Sie auch das Paket anacron installieren. Dieses sorgt dafür, dass Cronjobs, die eigentlich während der Zeit ausgeführt werden sollten, zu der Sie Ihren Rechner ausgeschaltet hatten, nachträglich ausgeführt werden. Das Programm cron erlaubt es, zu bestimmten Zeitpunkten Programme auszuführen. Meistens sind dies administrative Aufgaben wie der Aufruf von updatedb.

Dadurch, dass die Datenbank einmal täglich aktualisiert wird, kann locate natürlich auch nur Dateien finden, die zu diesem Zeitpunkt bereits vorhanden waren. Später erzeugte Dateien werden von locate nicht angezeigt. Das klang jetzt etwas aufwändig; locate ist aber sehr leicht zu benutzen, wie folgendes Beispiel beweist:

```
bash-2.02$ locate XF86Config /etc/X11/XF86Config
/usr/X11R6/lib/X11/XF86Config /usr/X11R6/lib/X11/XF86Config.eg
/usr/X11R6/man/man5/XF86Config.5x.gz
```

Wie Sie sehen, findet locate alle Dateien, die den Suchbegriff beinhalten, ohne dass Sie mit Wildcards arbeiten müssen.

Wenn Sie find und locate vergleichen, werden Sie feststellen, dass find leistungsfähiger ist, locate dagegen im täglichen Gebrauch Dateien wesentlich schneller finden kann.

Wenn Sie häufiger mit locate arbeiten oder auch die Datenbank nicht als Administrator aktualisieren lassen, werden Sie merken, dass unter Umständen nicht alle Dateien angezeigt werden. Dies liegt daran, dass nur Dateien in die Datenbank wandern, auf die das Programm updatedb Zugriff hat.

Weiterhin zeigt locate aber auch Dateien an, die ein normaler User nicht sehen sollte; dies ist vielleicht nicht gewünscht.

Debian GNU/Linux beinhaltet noch das Paket slocate. Diese spezielle Version zeigt nur die Dateien an, auf die der jeweilige User auch Zugriff hat. Sie können zusätzlich das Paket suidmanager installieren. Damit kann slocate über das dazugehörige Programm suidregister Ihnen auch die Dateien anzeigen, auf die Sie normalerweise keinen Zugriff haben; natürlich erst nach Angabe des entsprechenden Passwortes.

Beachten Sie bitte auch den Hinweis am Ende der Installation von slocate:

```
sushi:/root# apt-get install slocate Reading Package Lists... Done
Building Dependency Tree... Done The following NEW packages will
be installed:   slocate 0 packages upgraded, 1 newly installed, 0 to
remove and 27 not upgraded. Need to get 0B/23.2kB of archives. After
unpacking 143kB will be used. Selecting previously deselected package
slocate. (Reading database ... 67742 files and directories currently
installed.) Unpacking slocate (from .../utils/slocate_2.1-5.1.deb) ...
Adding `diversion of /usr/bin/locate to /usr/bin/locate.notslocate by
slocate' Adding `diversion of /usr/bin/updatedb to
/usr/bin/updatedb.notslocate by slocate' Adding `diversion of
/usr/share/man/man1/locate.1.gz to /usr/share/man/
man1/locate.notslocate.1.gz by slocate' Adding `diversion of
/usr/share/man/man1/updatedb.1.gz to /usr/share/
man/man1/updatedb.notslocate.1.gz by slocate' Adding `diversion of
/etc/cron.daily/find to /etc/cron.daily/find.notslocate by slocate' Setting
up slocate (2.1-5.1) ... Adding group slocate (104)... Done. Changing
permissions on: /usr/bin/slocate Changing permissions on:
/var/lib/slocate to: 0750  WARNING: You should run
'/etc/cron.daily/slocate' as root. locate will  not work properly until you
do or until it is run by cron (it is daily).
```

Manchmal ist es sinnvoll, große Dateien zu komprimieren, sei es, um Festplattenplatz zu sparen, sei es, um Downloadzeiten zu verkürzen. Das Programm der Wahl unter Debian GNU/Linux ist gzip (GNU Zip).

Erstellen Sie zuerst eine Testdatei, um mit gzip experimentieren zu können, und sehen Sie sich die Größe dieser Datei an.

```
bash-2.03$ cd bash-2.03$ cp /etc/profile ./testdatei bash-2.03$ ls -l
testdatei -rw-r--r--    1 fr       fr                  359 Jan 20 20:10 testdatei
```

Komprimieren Sie nun die Datei testdatei mit gzip und sehen Sie sich wieder das Ergebnis an:

```
bash-2.03$ gzip testdatei bash-2.03$ ls -l testdatei.gz -rw-r--r--    1 fr
fr                275 Jan 20 20:10 testdatei.gz
```

Beachten Sie, dass die Datei nun die Endung .gz bekommen hat. Somit ist klar zu erkennen, mit welchem Programm die Datei gepackt wurde und dass diese Datei überhaupt gepackt ist.

Um nun diese Datei wieder in den ursprünglichen Zustand zu versetzen, können Sie gzip mit der Option -d aufrufen:

```
bash-2.03$ gzip -d testdatei.gz bash-2.03$ ls -l testdatei -rw-r--r--    1 fr
fr                359 Jan 20 20:10 testdatei
```

Somit ist der alte Zustand wieder hergestellt. Der Komprimierungserfolg ist bei so kleinen Dateien nicht sehr groß, Sie können das gleiche Experiment aber auch mit anderen, größeren Dateien probieren.

Sicher standen Sie schon einmal vor dem Problem, dass eine Datei zu groß war. Sei es, um diese auf einem Medium zu transportieren oder um diese übers Netz zu verschicken. Unter Debian GNU/Linux ist es mit den Programmen split und cat möglich, Dateien zu zerteilen und wieder zusammenzufügen.

Kopieren Sie die Datei /bin/bash in Ihr Heimat-Verzeichnis. Diese Datei hat eine Größe von etwas mehr als 450 Kilobyte.

```
bash-2.03$ cd bash-2.03$ cp /bin/bash . bash-2.03$ ls -l bash -rwxr-xr-x
1 fr        fr                461720 Jan 22 15:42 bash
```

Sie können nun diese Datei mit dem Kommando split in kleinere Stücke teilen. Hierzu benötigt split Angaben über die maximale Größe der einzelnen Fragmente sowie die Erweiterung des Dateinamens, die an jede der erzeugten Teil-Dateien angehängt werden soll.

Mit der Option -b teilen Sie split die Größe mit. Ohne weitere Angaben geht split davon aus, dass der Wert in Byte angegeben wurde. Das ist natürlich nicht sehr praktikabel. Deshalb können Sie hinter dem Zahlenwert die Buchstaben k für Kilobyte oder m für Megabyte angeben.

Als Kennung für die einzelnen Dateien erweitert split den Dateinamen der ersten Datei mit aa, den zweiten mit ab und so weiter. Wenn Sie eine eigene Erweiterung (Präfix) zu jeder Datei erzeugen wollen, so können Sie diese mit angeben. Hier im Beispiel wird „einzel“ angegeben:

```
bash-2.03$ split -b100k bash einzel bash-2.03$ ls -l einzel* -rw-r--r--
1 fr        fr         102400 Jan 22 15:42 einzelaa -rw-r--r--    1 fr        fr
102400 Jan 22 15:42 einzelab -rw-r--r--    1 fr        fr         102400 Jan 22
15:42 einzelac -rw-r--r--    1 fr        fr         102400 Jan 22 15:42 einzelad
-rw-r--r--    1 fr        fr          52120 Jan 22 15:42 einzelae
```

Das Zusammenfügen der Dateien ist ebenfalls sehr einfach.

```
bash-2.03$ cat einzel* > bash-neu bash-2.03$ ls -l bash* -rwxr-xr-x    1
fr        fr         461720 Jan 22 15:42 bash -rw-r--r--    1 fr        fr
461720 Jan 22 15:43 bash-neu
```

Anhand der Länge sehen wir, dass die Datei wiederhergestellt wurde.

3.14.14.1 tar - Packen von Dateien

Häufig bekommt man Archive aus dem Netz in gepackter Form. Das Packen von Archiven beinhaltet zwei Dinge: erstens die Zusammenfassung von mehreren Dateien zu einer einzigen und zweitens das Komprimieren der Daten, um Festplattenplatz zu sparen oder auch die Übertragungszeit zu verringern. Historisch gesehen, verteilen sich diese beiden Funktionen unter Unix auch auf zwei Programme. Üblicherweise benutzt man zum Zusammenfassen der Dateien das Programm tar (Tape Archive) und zum Komprimieren das Programm gzip. Etwas bessere Ergebnisse beim Komprimieren von Daten erreicht das (neuere) Programm bzip2. Die GNU-Version des Programms tar, die auch bei Debian GNU/Linux verwendet wird, kann beide Komprimierungsverfahren benutzen, so dass man nicht mit verschiedenen Programmen hantieren muss.

Altgediente Programme, wie zum Beispiel tar, verfügen häufig über eine Vielzahl von Funktionen. Einige davon werden heute kaum noch verwendet, sind aber trotzdem noch aus Kompatibilitätsgründen verfügbar. Beispielsweise können Sie die Blockgröße bestimmen, mit der die Daten auf das Medium geschrieben werden. Eine solche Funktion werden Sie heute nur noch in sehr seltenen Fällen benötigen.

Einen Überblick über die Optionen von tar bekommen Sie wie üblich mit der Option --help:

```
bash-2.03$ tar --help  GNU "tar" schreibt mehrere Dateien in ein
Archiv auf Band oder Festplatte  und kann einzelne Dateien aus diesem
Archiv herausholen.  Verwendung: tar [OPTION]... [Datei]...
Beispiele:   tar -cf archiv.tar foo bar  # archiv.tar mit den Dateien foo
und bar                    # erzeugen.   tar -tvf archiv.tar        # Inhalt
von archiv.tar ausführlich anzeigen.   tar -xf archiv.tar          # Alle
Dateien aus archiv.tar extrahieren  Wenn eine lange Option ein
Argument erfordert, ist es für die entsprechende kurze Option auch
erforderlich. Das Gleiche gilt für optionale Argumente.  Aktionen:   -t, -
-list             Inhalt eines Archivs anzeigen   -x, --extract, --get    Dateien
aus Archiv holen   -c, --create             neues Archiv erzeugen   -d, --diff,
--compare   Dateien im Archiv und im Dateisystem vergleichen   -r, --
append            Dateien an das Archiv anhängen   -u, --update            nur
Dateien anhängen, die jünger sind als ihre                  Archiv-
Version   -A, --catenate          "tar"-Dateien an ein Archiv anhängen
--concatenate       wie '-A'       --delete            aus Archiv löschen (nicht
```

auf Magnetbändern!) Operation modifiers: -W, --verify attempt to verify the archive after writing it --remove-files remove files after adding them to the archive -k, --keep-old-files don't replace existing files when extracting --overwrite overwrite existing files when extracting -U, --unlink-first remove each file prior to extracting over it --recursive-unlink empty hierarchies prior to extracting directory -S, --sparse handle sparse files efficiently -O, --to-stdout extract files to standard output -G, --incremental handle old GNU-format incremental backup -g, --listed-incremental=FILE handle new GNU-format incremental backup --ignore-failed-read do not exit with nonzero on unreadable files Datei-Eigenschaften: --owner=NAME Eigentümer für hinzugefügte Dateien auf NAME setzen --group=NAME Gruppe für hinzugefügte Dateien auf NAME setzen --mode=RECHTE Zugriffsrechte für hinzugefügte Dateien auf RECHTE setzen --atime-preserve Zugriffszeit beim Auspacken erhalten -m, --modification-time Änderungszeit beim Auspacken erhalten --same-owner Eigentümer beim Auspacken erhalten --no-same-owner Eigentümer beim Auspacken auf Ausführenden setzen --numeric-owner Zahlen für Benutzer bzw. Gruppen benutzen -p, --same-permissions Zugriffsrechte beim Auspacken erhalten --no-same-permissions Keine Zugriffsrechte erhalten --preserve-permissions wie '-p' -s, --same-order zu entpackende Dateinamen wie im Archiv sortieren --preserve-order wie '-s' --preserve wie '-p' und '-s' zusammen Geräteauswahl und -einstellung: -f, --file=ARCHIV Gerät oder Datei ARCHIV benutzen --force-local Archiv-Datei ist lokal, auch wenn der Name einen Doppelpunkt enthält --rsh-command=BEFEHL statt "rsh" den BEFEHL benutzen -[0-7][lmh] Laufwerk und Schreibdichte angeben -M, --multi-volume mehrteiliges Archiv anlegen/listen/ herausholen -L, --tape-length=ZAHL Medium wechseln, wenn ZAHL KBytes geschrieben sind -F, --info-script=DATEI am Ende jedes Mediums das Skript DATEI ausführen (impliziert '-M') --new-volume-script=DATEI wie '-F DATEI' --volno-

file=DATEI Teil-Nummer in DATEI benutzen/aktualisieren Blockung des Gerätes: -b, --block-size=BLÖCKE BLÖCKE à 512 Bytes pro Record --record-size=GRÖSSE GRÖSSE Bytes pro Record, Vielfaches von 512 -i, --ignore-zeros Blöcke mit Nullen im Archiv ignorieren (heißt EOF) -B, --read-full-blocks Blockung beim Lesen ändern (für 4.2BSD "Pipes") Archive format selection: -V, --label=NAME create archive with volume name NAME PATTERN at list/extract time, a globbing PATTERN -o, --old-archive, --portability write a V7 format archive --posix write a POSIX format archive -j, --bzip2 filter the archive through bzip2 -z, --gzip, --ungzip filter the archive through gzip -Z, --compress, --uncompress filter the archive through compress --use-compress-program=PROG filter through PROG (must accept -d) Local file selection: -C, --directory=DIR change to directory DIR -T, --files-from=NAME get names to extract or create from file NAME --null -T reads null-terminated names, disable -C --exclude=PATTERN exclude files, given as a globbing PATTERN -X, --exclude-from=FILE exclude globbing patterns listed in FILE -P, --absolute-names don't strip leading `/'s from file names -h, --dereference dump instead the files symlinks point to --no-recursion avoid descending automatically in directories -l, --one-file-system stay in local file system when creating archive -K, --starting-file=NAME begin at file NAME in the archive -N, --newer=DATUM nur Dateien jünger als DATUM sichern --newer-mtime Datum und Zeit nur vergleichen, wenn sich der Datei-Inhalt geändert hat --after-date=DATUM wie '-N' --backup[=CONTROL] Sicherheitskopie vor dem Löschen, wählt Versionskontrolle --suffix=SUFFIX Sicherheitskopie vor dem Löschen, Namenserweiterung SUFFIX Informationen: --help Hilfe anzeigen und "tar" beenden --version Versionsnummer anzeigen und "tar" benden -v, --verbose zu bearbeitende Dateien ausführlich listen --checkpoint Verzeichnisnamen beim Lesen des Archivs zeigen --totals geschriebene Bytes beim Schreiben des Archivs zeigen -R, --block-number Blocknummer innerhalb des Archivs mit jeder Meldung zeigen -w, --interactive Bestätigung für jede Aktion

verlangen --confirmation wie '-w' Die Namenserweiterung für Sicherheitskopien ist '~', wenn nicht mit --suffix oder der Umgebungsvariablen SIMPLE_BACKUP_SUFFIX etwas anderes eingestellt ist. Die Versionskontrolle kann mit --backup oder der Umgebungsvariablen VERSION_CONTROL gesetzt werden. Mögliche Werte sind: t, numbered nummerierte Sicherheitskopien nil, existing nummerierte Sicherheitskopien, wenn schon nummerierte vorhanden sind, sonst einfache never, simple immer einfache Sicherheitskopien GNU "tar" kann keine '--posix'-Archive lesen. Wenn die Umgebungsvariable POSIXLY_CORRECT gesetzt ist, sind GNU-Erweiterungen mit '--posix' nicht zulässig. Unterstützung für POSIX ist nur teilweise implementiert, rechne derzeit noch nicht damit! ARCHIV kann DATEI, RECHNER:DATEI oder BENUTZER@RECHNER:DATEI sein; DATEI kann eine Datei oder ein Gerät (z.B. ein Streamer) sein. Die Voreinstellung für _dieses_ "tar" ist '-f- -b20'. Fehlermeldungen an <bug-tar@gnu.org>.

Lange Texte ansehen

Wenn die Anzeige eines längeren Textes nicht komplett auf dem Bildschirm erfolgen kann, können Sie den Text seitenweise ausgeben lassen, indem Sie die Ausgabe über eine „Pipe“ an das Programm more weiterreichen: tar --help | more

Für den täglichen Gebrauch kommen Sie aber mit maximal zehn von diesen vielen Optionen aus. Auch in diesem Abschnitt zeigen wir einige Beispiele aus der Praxis auf.

Um mehrere Dateien in einem Archiv zusammenzufassen, benutzen Sie die Optionen -cf (create - erzeugen und file - Datei):

```
linux:/home/fr# tar -cf /tmp/test.tar /etc/ tar: Removing leading `/' from member names
```

Dies erzeugt eine neue Datei (/tmp/test.tar) mit allen Dateien und Unterverzeichnissen aus dem Verzeichnis /etc/.

tar entfernt automatisch das jedem Pfad vorangestellte /, bevor die Dateien in das Archiv aufgenommen werden. Dies verhindert, dass beim späteren Entpacken versehentlich Daten überschrieben werden.

Um die Daten wieder zu entpacken, benutzen Sie die Option -x (extract - entpacken). Beachten Sie, dass die Daten per Voreinstellung immer an der aktuellen Position im Dateisystem entpackt werden. Wenn Sie sich nicht sicher sind, erstellen Sie ein temporäres Arbeitsverzeichnis und verschieben Sie danach die Daten an die gewünschte Stelle:

```
linux:/home/fr# mkdir bla linux:/home/fr# cd bla linux:/home/fr/bla# tar -xf /tmp/test.tar linux:/home/fr/bla# ls etc
```

tar zum Plaudern bringen...

Benutzen Sie die zusätzliche Option -v, um den Vorgang des Packens oder Entpackens der Daten verfolgen zu können: tar -xvf /tmp/test.tar.

Bisher haben wir tar zum Packen von Dateien benutzt. Nun werden wir das Archiv zusätzlich noch komprimieren. Hierzu stehen bei GNU-Tar die Optionen -z für das

Komprimieren mit gzip und -j zum Komprimieren mit bzip2 zur Verfügung. Die Benutzung ist ganz einfach: Verwenden Sie tar wie oben gezeigt, und fügen Sie beispielsweise die Option -z hinzu.

Jedes Zeichen zählt...

Zur Vereinfachung können Sie den Strich - vor den Optionen einfach weglassen, tar versucht, die erste Zeichenkette hinter dem Kommando als Optionen zu interpretieren: tar xvf /tmp/test.tar.

Hier ein Beispiel, wie Sie ein gepacktes Archiv erzeugen können:

```
linux:/home/fr# tar cvfz /tmp/test.tar.gz /etc/
```

Beachten Sie, dass es üblich ist, entweder (wie hier gezeigt) die Endung .gz anzuhängen oder aber die Kurzform .tgz zu verwenden.

tar kann mit beiden Endungen „umgehen“, genau genommen ist der Dateiname völlig egal. Die Endungen dienen nur der besseren Übersicht für den Benutzer.

Wenn Sie die Daten auf einem Tape-Streamer speichern wollen, können Sie die Daten auch direkt auf das Gerät speichern. Geben Sie dazu statt des Dateinamens des Archivs einfach das entsprechende Device an:

```
linux:/home/fr# tar cvfz /dev/st0 /etc/
```

Das Device /dev/st0 wird von Streamern benutzt, die über einen SCSI-Anschluss verfügen. Wenn Sie die Daten von diesem Gerät wieder einlesen wollen, benutzen Sie dazu ebenfalls das entsprechende Device.

```
linux:/home/fr# tar xvfz /dev/st0
```

Es ist üblich, Dateien so zu benennen, dass der Typ der Datei am Namen zu erkennen ist. Programme sind nicht zwingend auf eine solche Endung angewiesen, diese dient nur der besseren Übersicht für den Benutzer (mit Ausnahme von Dateinamen, die fest in das Programm einkompiliert sind, wie zum Beispiel Namen von Konfigurationsdateien). Textdateien bekommen die Endung .txt, Perl-Programme die Endung .pl und Bilder die Endung .jpeg oder .tiff und so weiter. Trotzdem kann es passieren, dass Dateien entweder keine oder eine falsche Endung haben. Debian GNU/Linux stellt Ihnen das Programm file zur Verfügung, das die meisten Dateitypen ermitteln kann.

Das Programm file ist denkbar einfach zu benutzen: Rufen Sie das Programm einfach zusammen mit dem Namen der zu bestimmenden Datei(en) auf. Sie können hier einen oder mehrere Dateinamen, auch mit Wildcards, angeben:

```
bash-2.03$ file /bin/bash /etc/passwd /etc/init.d/lpd /bin/bash:       ELF
32-bit LSB executable, Intel 80386, version 1, dynamically linked (uses
shared libs), stripped /etc/passwd:     ASCII text /etc/init.d/lpd: Bourne
shell script text
```

file listet nun die verschiedenen Dateinamen mit den Dateitypen auf.

sed steht für Stream EDitor und ist ein Unix-Werkzeug, mit dem Texte bearbeitet werden können. Hierbei wird das Ergebniss auf der Standard-Ausgabe angezeigt - die Datei aus der gelesen wird wird nicht verändert. Dies kann mit einer speziellen Option -i erzwungen werden. Alternativ kann man mit der Ausgabeumleitung > die veränderte Daten in eine neue Datei schreiben.

In den meisten Fällen wird das Programm für das suchen und ersetzen von Texten in Dateien verwendet, hier ein kurzes Beispiel:

```
sed 's/Linux/Debian/g' unix.txt
```

Ergebniss: in der Datei unix.txt wird jedes Vorkommen des Wortes „Linux“ durch das Wort „Debian“ ersetzt.

sed Beispiele

Eine sehr schöne und umfangreiche Sammlung von Beispielen zu sed finden Sie hier: sed.sourceforge.net/sed1line_de.html.

Auf der Homepage finden sich weitere umfassende Links zu diesem mächtigen Programm: sed.sourceforge.net/.

3.14 Arbeiten mit Dateien - Mini-Workshop

3.13 Orientierung innerhalb von Debian 3.15 Einige bash-Funktionen

3.15 Einige bash-Funktionen

3.14 Arbeiten mit Dateien - Mini-Workshop 3.16 Pipes

3.15.1 help

Wir möchten hier nur kurz auf die Debian GNU/Linux-Standard-Shell, die bash, eingehen. Weitere Informationen finden Sie in der Manpage zur bash.

Eine der nützlichsten Funktionen der bash ist die Möglichkeit, Programm- und Dateinamen zu vervollständigen. Sie können dies zu jeder Zeit am Shell-Prompt ausprobieren: Tippen Sie ein paar Zeichen eines Befehls, und drücken Sie die **TAB**-Taste. Wenn die Eingabe bis zu der Stelle, an der Sie die **TAB**-Taste gedrückt haben, eindeutig war, es also kein zweites Programm gibt, das mit der gleichen Buchstabenfolge beginnt, wird die Eingabe automatisch vervollständigt. Sollte es zwei oder mehr Möglichkeiten geben, so werden Sie lediglich einen „Piepton“ hören. Ein nochmaliges Drücken der **TAB**-Taste zeigt Ihnen alle möglichen Alternativen an. Probieren Sie es an einem Beispiel einmal aus: Nehmen wir an, Sie möchten sich die Datei /var/log/syslog ansehen. Dazu benutzen Sie das Programm more. Geben Sie also mo ein, und drücken Sie die **TAB**-Taste zweimal. Nun sollten Sie zumindest die Programme mount und more angezeigt bekommen.

Geben Sie nun so lange weitere Zeichen ein, bis die Eingabe eindeutig ist, in diesem Beispiel sollte also ein r ausreichen. Drücken Sie nun die **TAB**-Taste, und der Befehl wird vervollständigt. Probieren Sie dies noch einmal (als Administrator) mit dem Dateinamen /var/log/syslog aus. Versuchen Sie immer nach einigen Zeichen, mit der **TAB**-Taste die Eingabe zu vervollständigen.

Eine weitere recht nützliche Funktion ist das Wiederholen von bereits eingegebenen Kommandos. Sie können mit den Pfeiltasten (**PFEIL-OBEN** und **PFEIL-UNTEN**) durch die so genannte „History“ der bash blättern und bereits ausgeführte Kommandos noch einmal ausführen oder auch gleich auf der Kommandozeile ändern.

Die bash verfügt neben der obligatorischen Manpage auch über eine eingebaute Hilfe-Funktion. Mit dem Kommando help wird eine Übersicht der in der bash enthaltenen Kommandos angezeigt.

```
GNU bash, version 2.05a.0(1)-release (i386-pc-linux-gnu) These shell
commands are defined internally.  Type 'help' to see this list. Type 'help
name' to find out more about the function 'name'. Use 'info bash' to find
```

```
out more about the shell in general.  A star (*) next to a name means
that the command is disabled.   %[DIGITS | WORD] [&]              .
filename  :                              [ arg... ]  alias [-p] [name[=value] ... ]
bg [job_spec]  bind [-lpvsPVS] [-m keymap] [-f fi break [n]  builtin
[shell-builtin [arg ...]]  case WORD in [PATTERN [| PATTERN].  cd [-
PL] [dir]                 command [-pVv] command [arg ...]  compgen [-
abcdefgjkvu] [-o option] complete [-abcdefgjkvu] [-pr] [-o   continue
[n]                  declare [-afFrxi] [-p] name[=value  dirs [-clpv] [+N] [-
N]           disown [-h] [-ar] [jobspec ...]  echo [-neE] [arg ...]
enable [-pnds] [-a] [-f filename]   eval [arg ...]                exec [-cl] [-a
name] file [redirec  exit [n]                    export [-nf] [name ...] or
export   false                        fc [-e ename] [-nlr] [first] [last  fg
[job_spec]                for NAME [in WORDS ... ;] do COMMA
function NAME { COMMANDS ; } or NA getopts optstring name
[arg]  hash [-r] [-p pathname] [-t] [name help [-s] [pattern ...]  history [-
c] [-d offset] [n] or hi if COMMANDS; then COMMANDS; [ elif  jobs
[-lnprs] [jobspec ...] or job kill [-s sigspec | -n signum | -si  let arg [arg
...]              local name[=value] ...  logout                    popd [+N
| -N] [-n]  printf format [arguments]         pushd [dir | +N | -N] [-n]  pwd
[-PL]                  read [-ers] [-t timeout] [-p promp  readonly [-anf]
[name ...] or read return [n]  select NAME [in WORDS ... ;] do CO set
[--abefhkmnptuvxBCHP] [-o opti  shift [n]                  shopt [-pqsu]
[-o long-option] opt  source filename              suspend [-f]  test
[expr]                 time [-p] PIPELINE  times                  trap
[arg] [signal_spec ...] or tr  true                      type [-apt] name
[name ...]  typeset [-afFrxi] [-p] name[=value ulimit [-SHacdflmnpstuv]
[limit]  umask [-p] [-S] [mode]          unalias [-a] [name ...]  unset [-f]
[-v] [name ...]        until COMMANDS; do COMMANDS; done
variables - Some variable names an wait [n]  while COMMANDS; do
COMMANDS; done  { COMMANDS ;  }
```

Zu jedem der aufgelisteten Kommandos sind, ebenfalls mit dem Kommando help, genauere Informationen zu bekommen. Beispielsweise zu shift: Der Befehl help shift ergibt:

```
shift: shift [n]    The positional parameters from $N+1 ... are renamed
to $1 ...  If N is    not given, it is assumed to be 1.
```

Wenn ausführlichere Informationen benötigt werden, hilft ein Blick in die Manpage von bash weiter.

3.15 Einige bash-Funktionen

3.14 Arbeiten mit Dateien - Mini-Workshop 3.16 Pipes

3.16 Pipes

3.15 Einige bash-Funktionen 3.17 ps

Bereits sehr früh wurde das Prinzip der „Pipes“ („Röhren“ ist keine schlechte Übersetzung) integriert. Sie können etwas „hineinschieben“, und am anderen Ende kommt es wieder heraus. Wie bereits beschrieben, gibt es sehr viele kleine, spezialisierte Programme unter Unix, die mit speziellen Parametern aufgerufen werden können. Sinnvoll wäre eine Schnittstelle zwischen diesen Programmen, um Daten auszutauschen oder auch das Ergebnis eines Programmlaufs in einem weiteren Programm aufzubereiten. Diese Schnittstelle ist in Form von „ Pipes“ realisiert. Sicher haben Sie schon das Kommando `ls` benutzt, um sich die Dateien in einem Verzeichnis anzeigen zu lassen. Wenn Sie aber in einem Verzeichnis mit sehr vielen Dateien die Anzahl der Dateien ermitteln möchten, kann das Zählen leicht etwas umständlich werden. Um Zeichen, Wörter oder Zeilen in einer Datei zu zählen, gibt es aber das Kommando `wc` (Word Count). Ein Weg wäre also, die Ausgabe von `ls -l` in eine Datei zu schreiben und mittels `wc -l` die Zeilen zählen zu lassen. Der Umweg über eine Datei lässt sich aber mit einer Pipe umgehen.

Unix benutzt hierfür das Zeichen | (Pipe). Verknüpfen Sie einfach die beiden Kommandos mittels dieses Zeichens zu einer Zeile: `ls -l | wc -l` gibt Ihnen die Anzahl der Dateien im aktuellen Verzeichnis aus. An dieser Stelle gibt es aber einen kleinen Haken: `ls` gibt als erste Zeile keinen Dateinamen aus, sondern eine Zeile, in der Informationen über das Verzeichnis aufgezeigt werden; Sie müssen also vom Ergebnis eine Zeile subtrahieren, um auf das genaue Ergebnis zu kommen.

Wenn Sie sich die Rechenarbeit ersparen wollen, können Sie auch die Option `-l` statt `-l` beim Kommando `ls` benutzen, doch das führt jetzt zu weit...

Ein weiteres Beispiel für die Benutzung von Pipes werden wir im folgenden Abschnitt zu `ps` aufzeigen. Sie werden im Laufe der Zeit an vielen Stellen auf weitere Anwendungsfälle stoßen.

3.16 Pipes

3.15 Einige bash-Funktionen 3.17 ps

3.17 ps

3.16 Pipes 3.18 Links

Um auf einem Debian GNU/Linux-System die vielen gleichzeitig laufenden Programme im Zaum halten zu können, muss man sich natürlich auch einen Überblick über diese verschaffen. Dazu dient unter anderem das Programm ps. ps liest die benötigten Informationen unter Linux aus dem Verzeichnis /proc des Dateisystems. /proc ist nicht tatsächlich auf einer Festplattenpartition abgelegt, sondern wird vom Kernel ständig aktualisiert und in den Verzeichnisbaum eingeblendet. Somit hat der Benutzer auf einfachste Art und Weise Zugriff auf die Informationen.

Rufen Sie ps einfach einmal ohne weitere Parameter in einer Shell auf. Dies sollte Ihnen in etwa folgendes Ergebnis anzeigen:

```
bash-2.02$ ps PID TTY          TIME CMD 3522 pts/5    00:00:00 bash
3523 pts/5    00:00:00 ps
```

Ohne weitere Optionen zeigt ps die Prozesse der aktuellen Shell an. Zu diesem Zeitpunkt sind das die Shell selbst sowie das Programm ps, das ja gerade gestartet wurde. In der ersten Spalte sehen Sie die Prozess-ID (PID); diese dient dazu, ein Programm im System eindeutig zu identifizieren. ps verfügt über eine Vielzahl von Optionen, mit denen Sie sich detailliertere Informationen zu den laufenden Programmen ansehen können.

Eine Kurzübersicht über die verfügbaren Optionen erhalten Sie, wie bei allen GNU-Programmen, mit der Option --help.

```
bash-2.02$ ps --help  ********* simple selection *********
********* selection by list ********* -A all processes                -C
by command name -N negate selection               -G by real group ID
(supports names) -a all w/ tty except session leaders  -U by real user ID
(supports names) -d all except session leaders        -g by session leader
OR by group name -e all processes                 -p by process ID T  all
processes on this terminal     -s processes in the sessions given a  all w/
tty, including other users  -t by tty g  all, even group leaders!         -u
by effective user ID (supports                          names) r  only
running processes           U  processes for specified users x  processes
w/o controlling ttys     t  by tty ********** output format **********
```

********** long options ********** -o,o user-defined -f full --Group --User --pid --cols -j,j job control s signal --group --user --sid --rows -O,O preloaded -o v virtual memory --cumulative --format --deselect -l,l long u user-oriented --sort --tty --forest --version X registers --heading --no-heading ********* misc options ********* -V,V show version L list format codes f ASCII art forest -m,m show threads S children in sum -y change -l format -n,N set namelist file c true command name n numeric WCHAN,UID -w,w wide output e show environment -H process hierarchy

Am häufigsten werden Sie sicher Optionen wie a, u, x und w benutzen. Diese werden nach dem Kommando einfach zusammengefasst, also beispielsweise ps auxw.

Nun folgt das versprochene Beispiel zum Thema Pipe: Nehmen wir an, Sie benötigen die Prozess-ID eines Programms, um es mit kill zu beenden. Auf einem System mit vielen Prozessen kann dies ein Problem sein. Die Lösung ist eine Kombination aus den Programmen ps und grep, die mittels einer Pipe verkettet werden. Das benötigte Kommando würde wie folgt aussehen: ps aux|grep netscape. ps mit den Optionen aux gibt alle laufenden Prozesse in einer ausführlichen Form aus. grep filtert aus der Ausgabe des Programms ps alle Zeilen heraus, in denen die Zeichenfolge netscape vorkommt. Sie sollten nun eine recht knappe Ausgabe bekommen und die gewünschte Prozess-ID leicht finden können.

Weitere Informationen zu ps finden Sie in der Manpage zu ps (man ps).

3.17 ps

3.16 Pipes 3.18 Links

3.18 Links

3.17 ps 3.19 vi

Unter Unix werden so genannte Links bereits seit vielen Jahren eingesetzt. Sie kennen diese vielleicht schon von anderen Betriebssystemen unter dem Namen „Verknüpfungen“. Um einen Link zu erzeugen, bedient man sich des Kommandos ln. Auch dieses Kommando verfügt über die Option --help, die Ihnen eine kurze Information zu den verfügbaren Optionen gibt:

bash-2.02$ ln --help Benutzung: ln [OPTION]... ZIEL [VERKNÜPFUNGSNAME] oder: ln [OPTION]... ZIEL... VERZEICHNIS oder: ln [OPTION]... --target-directory=VERZEICHNIS ZIEL... Erzeugen einer Verknüpfung des angegebenen ZIELES mit optionaler VERKNÜPFUNG. Wenn mehr als ein ZIEL angegeben wird, muss das letzte Argument ein Verzeichnis sein. Erzeugen von Verknüpfungen für jedes ZIEL in VERZEICHNIS. Als Standardvorgabe werden harte Verknüpfungen erstellt, symbolische Verknüpfungen mit --symbolic. Beim Erzeugen von harten Verknüpfungen muss jedes ZIEL existieren. --backup=[KONTROLLE] Erzeugen von Sicherungen für vorhandene Zieldateien. -b Wie --backup, akzeptiert aber kein Argument. -d, -F, --directory Verzeichnisse hart verknüpfen. (Nur Super-User) -f, --force Vorhandene Ziele entfernen. -n, --no-dereference Behandeln eines Zieles, das eine symbolische Verknüpfung auf ein Verzeichnis ist, wie normale Datei. -i, --interactive Nachfrage vor Entfernen vorhandener Ziele. -s, --symbolic Symbolische statt harter Verknüpfung erzeugen -S, --suffix=SUFFIX Überschreiben der normalen Anhänge für Sicherungen. --target-directory=VERZ Angabe des VERZeichnisses, in dem die Verknüpfungen erstellt werden sollen. -v, --verbose Ausgabe des Namens jeder Datei vor dem Verknüpfen. --help Anzeige dieser Hilfe und beenden. --version Ausgabe der Versionsinformation und beenden. Der Anhang für Sicherheitskopien ist ~, außer wenn er --suffix oder SIMPLE_BACKUP_SUFFIX gesetzt wurde. Die Versionskontrolle kann mit --backup oder VERSION_CONTROL gesetzt werden. Mögliche Werte sind: none,

```
off      Niemals Sicherung erzeugen (selbst wenn --backup
angegeben wurde)   numbered, t     Erzeugen von nummerierten
Sicherheitskopien   existing, nil   Nummeriert wenn nummerierte
Backups existieren,                 sonst einfach.   simple, never   Immer
einfache Sicherheitskopien erzeugen  Berichten Sie Fehler an <bug-
fileutils@gnu.org>.
```

Aber das sieht komplizierter aus als es ist. In 98% aller Fälle wird Ihnen `ln` in der „Sparversion“ als `ln -s originaldatei link-zur-datei` ausreichen. Weitere Informationen finden Sie wie immer auch in der Manpage.

3.18 Links

3.17 ps 3.19 vi

3.19 vi

3.18 Links 3.20 Dateisysteme

Der Unix-Standard-Editor vi ist nach der Installation des Basissystems auf jedem Debian GNU/Linux-System verfügbar. vi wird schon seit vielen Jahren auf Unix-Systemen eingesetzt. Seine für Anfänger kryptische Bedienung rührt aus der langen Geschichte dieses Editors her. In der Urzeit der Computertechnik standen keine aufwändigen grafischen Arbeitsplätze zur Verfügung. Textdrucker mit Tastatur oder - etwas moderner - Text-Terminals (VT100, ein Modell der Firma DEC, ist noch heute ein Begriff), die seriell an den Rechner angeschlossen wurden, waren der Stand der Technik.

Debian verwendet bereits von Haus aus eine verbesserte Version des „vi“, den „nvi“.

In jedem Fall ist es sinnvoll, ein paar wenige Grundlagen über den vi zu erfahren. Dieser Editor ist einfach immer verfügbar, auch auf einem minimalem System. Unabhängig von Ihren Vorlieben (oder denen Ihrer Freunde) sind Grundkenntnisse in der Bedienung des vi für jeden Benutzer eines Unix-Systems notwendig, spätestens wenn der Bereich der Systemadministration gestreift wird.

Wenn Sie sich etwas in den vi eingearbeitet haben, werfen Sie mal einen Blick auf Vim, „vi improved“, der eine erweiterte Version des vi ist.

Beim vi wird zwischen einem Kommando- und einem Eingabemodus unterschieden. Durch Drücken der Taste **i** für „Input“ kommen Sie in den Eingabemodus. Ein Druck auf die Taste **ESC** beendet den Eingabemodus, und man befindet sich wieder im Kommandomodus. Zum Eingabemodus gibt es nicht viel zu sagen, es können damit weitere Zeichen in die Datei eingegeben werden.

Interessanter ist der Kommandomodus des vi. Mit einzelnen Tasten können Sie im Text navigieren oder auch Zeichen bzw. Zeilen löschen. Laden Sie einfach eine Datei, beispielsweise mit den Kommandos: cp /etc/hosts . (legt eine Kopie der Datei im aktuellen Verzeichnis ab) und vi ./hosts (startet vi und lädt die Datei). Nach dem Start des vi befinden Sie sich im Kommandomodus. Sie können nun ein einzelnes Zeichen löschen, indem Sie die Taste **x** drücken. Ein solcher Befehl kann mit der Taste **u** rückgängig gemacht werden.

Um den Cursor im Text zu bewegen, können Sie die Pfeiltasten benutzen. Sollte dies aufgrund fehlerhafter Einstellungen (zum Beispiel über eine telnet-Verbindung) einmal nicht funktionieren, können Sie in jedem Fall mit den Tasten **h** (bewegt den Cursor ein Zeichen nach links), **j** (springt zur nächsten Zeile), **k** (springt eine Zeile nach oben) und **l** (bewegt den Cursor ein Zeichen nach rechts) navigieren. Seitenweises Blättern kann mit den Tasten **STRG+U** (Up/Aufwärts) und **STRG+D** (Down/Abwärts) erreicht werden.

Viele der vi-Kommandos lassen sich auch „vervielfältigen". Beispielsweise löscht **9x** neun Zeichen ab der aktuellen Position. Dies funktioniert mit den meisten anderen Kommandos ebenso.

Wenn Sie die Änderungen in einer Datei speichern wollen, können Sie dies mit **:w** tun. Dateien unter einem anderen Namen speichern Sie mit **:w neu.txt**. Sie können den vi beenden, indem Sie **:q** eingeben. Auch hier sind Kombinationen möglich, so können Sie eine Datei mit **:wq** speichern und den Editor verlassen.

Häufig möchte man Zeilen kopieren: Hierzu dient der Befehl **yy**. Dieser speichert die aktuelle Zeile in einem Puffer. Die gespeicherten Daten lassen sich mit **p** wieder an einer anderen Stelle einfügen. Analog dazu lassen sich mit **7yy** sieben Zeilen kopieren und so weiter.

So viel zu den Grundzügen des Editors vi. Mit diesen wenigen Kommandos sind Sie in der Lage, Anpassungen an den Konfigurationsdateien Ihres Debian GNU/Linux-Systems vorzunehmen.

Wenn Sie sich eine Zeit lang mit diesem Editor beschäftigt haben, können Sie folgende Übersicht verwenden, um Ihr vi-Know-how zu vertiefen.

PDF-Dateien bearbeiten mit Vim

Das Paket pdftk (http://www.pdfhacks.com/pdftk/) erlaubt es, PDF-Dateien zu

bearbeiten. Über ein Makro (http://www.pdfhacks.com/pdftk/#vim_plugin) kann diese Funktionalität auch in Vim eingebunden werden.

Vim Tipps

Im „Vim Tips Wiki“ finden sich seh viele und nützliche Tipps zum Editor Vim: vim.wikia.com/wiki/Vim_Tips_Wiki.

Bereits beim Programmstart können einige Optionen angegeben werden. Diese sind unter anderem:

vi name - Startet den Editor und lädt die Datei name.

vi name1 name2 name3 - Startet den Editor und lädt die Dateien name1, name2 und name3.

vi -R name - Startet den Editor und lädt die Datei name im Nur-lesen-Modus.

vi -r name - Startet den Editor und restauriert die Datei name nach einem Absturz.

Verschiedene Dateien gleichzeitig bearbeiten

Vim beherrscht einen Modus, in dem es möglich ist, an unterschiedlichen Dateien zur gleichen Zeit zu arbeiten. Dazu wird mit :vsplit der Editierbereich vertikal in zwei so genannte „Fenster“ geteilt und mit :bn im aktuellen Fenster der nächste Buffer ausgewählt bzw. das nächste Fenster aktiviert. Die Bewegung der Schreibmarke (Cursor) kann in beiden Fenstern durch den Befehl :set scrollbind gekoppelt werden. Der Befehl muss dazu auch in beiden Fenstern ausgeführt werden!

Sollen bereits beim Aufruf von Vim zwei Dateien geladen werden, so kann eine Datei zwei.scr mit folgendem Inhalt angelegt werden:

```
:vsplit^M:bn^M:set scrollbind^M^W^W:set scrollbind^M^W^W
```

^M wird durch gleichzeitiges Drücken von Ctrl+M erreicht, ^W durch Drücken von Ctrl+W. Aktiviert wird dieses Skript nun durch:

```
vim -s zwei.scr dateia.txt dateib.txt
```

Wenn der Editor gestartet ist, können einige Einstellungen verändert werden. Sollen diese Änderungen dauerhaft gespeichert werden, so können diese in die Datei ~/.vimrc eingetragen werden.

:set - Zeigt die aktuellen Benutzereinstellungen.

:set all - Zeigt alle Einstellungen.

:set <option> - Aktiviert eine Option. Zum Beispiel aktiviert set number die Anzeige der Zeilennummern.

:set no<option> - Deaktiviert eine Option. Zum Beispiel deaktiviert set nonumber die Anzeige der Zeilennummern.

:set option? - Zeigt die möglichen Werte dieser Option.

ZZ - Speichert die Datei und beendet das Programm.

:wq - Speichert die Datei und beendet das Programm.

:w - Speichert die Datei.

:w! - Speichert die Datei, auch wenn die Zugriffsrechte auf Nur-lesen gesetzt sind.

:w name - Speichert die Datei unter dem Namen name.

:q - Beendet das Programm.

:q! - Beendet das Programm, Änderungen werden verworfen.

:e name - Lädt die Datei name.

:e! name - Lädt die Datei name erneut und verwirft alle bisherigen Änderungen.

:e + name - Lädt die Datei name und springt ans Ende der Datei.

:e +<n> name - Lädt die Datei name und springt in die Zeile < n>.

:n - Geht zur nächsten der geladenen Dateien.

:args - Zeigt die aktuelle Liste der Dateien.

:rew - Springt zur ersten Datei in der Dateiliste.

:f - zeigt den Namen der aktuellen Datei und die aktuelle Zeilennummer an.

:q - Beendet das Programm.

Pfeiltasten - Mit den Pfeiltasten kann der Cursor wie in anderen Anwendungen bewegt werden.

STRG+d> - Springt eine halbe Seite nach unten.

STRG+u> - Springt eine halbe Seite nach oben.

STRG+f - Springt eine Seite nach unten.

STRG+b - Springt eine Seite nach oben.

:0 - Springt zum Anfang der Datei.

:<n> - Springt zur Zeile < n> der Datei.

:$ - Springt zum Ende der Datei.

0 - Springt zum Anfang der Zeile.

^ - Springt zum ersten Zeichen, das kein Leerzeichen ist.

$ - Springt zum Ende der Zeile.

RETURN - Springt zum Anfang der nächsten Zeile.

% - Zeigt die zugehörige Klammer.

G - Springt zur letzten Zeile.

H - Springt zur ersten Zeile im aktuellen Fenster.

L - Springt zur letzten Zeile im aktuellen Fenster.

M - Springt in die Mitte des aktuellen Fensters.

- - Springt zum ersten Nicht-Leerzeichen der vorhergehenden Zeile.

+ - Springt zum ersten Nicht-Leerzeichen der nächsten Zeile.

j - Springt zur nächsten Zeile in der gleichen Spalte.

k - Springt zur vorhergehenden Zeile in der gleichen Spalte.

h - Springt ein Zeichen nach links.

l - Springt ein Zeichen nach rechts.

w - Springt ein Wort vorwärts.

b - Springt ein Wort rückwärts.

e - Springt zum Ende des Wortes.

) - Springt zum nächsten Satz.

(- Springt zum vorhergehenden Satz.

} - Springt zum nächsten Absatz.

{ - Springt zum vorhergehenden Absatz.

x - Löscht das Zeichen unter dem Cursor.

X - Löscht das Zeichen vor dem Cursor.

D - Löscht alles bis zum Ende der Zeile.

d^ - Löscht alles bis zum Anfang der Zeile.

dd - Löscht die gesamte Zeile.

<n>dd - Löscht <n> Zeilen.

d<n>w - Löscht <n> Wörter ab der Cursorposition.

i - Aktiviert den Eingabemodus vor dem Cursor.

I - Aktiviert den Eingabemodus vor dem ersten Nicht-Leerzeichen in der aktuellen Zeile.

a - Aktiviert den Eingabemodus nach dem Cursor.

A - Aktiviert den Eingabemodus nach dem Ende der Zeile.

o - Beginnt eine neue Zeile nach der aktuellen Zeile und aktiviert den Eingabemodus.

O - Beginnt eine neue Zeile über der aktuellen Zeile und aktiviert den Eingabemodus.

r<n> - Ersetzt das Zeichen unter dem Cursor durch das Zeichen < n>, der Eingabemodus wird nicht aktiviert.

R - Aktiviert den Eingabemodus, Zeichen werden überschrieben.

C - Ändert den Text bis zum Zeilenende.

D - Löscht den Text bis zum Zeilenende.

s - Ersetzt Zeichen.

S - Ersetzt Zeilen.

J - Entfernt den Zeilenumbruch am Ende einer Zeile, fügt also die aktuelle und die folgende Zeile zusammen.

yy - Kopiert die aktuelle Zeile.

<n>yy - kopiert <n> Zeilen.

p - Fügt die kopierten Zeilen nach der aktuellen Zeile ein.

P - Fügt die kopierten Zeilen vor der aktuellen Zeile ein.

Auch gelöschte Zeilen können mit den aufgeführten Kommandos eingefügt werden.

Absätze neu formatieren

Um mehrere Zeilen oder komplette Absätze, beispielsweise innerhalb einer zitierten E-Mail, neu umzubrechen, ist der gewünschte Abschnitt zunächst mittels „v“ (Visual-Mode) und den Pfeiltasten zu markieren. Danach formatiert die Tastenkombination „gqa“ den Absatz mit den aktuellen Einstellungen neu.

/text - Sucht vorwärts nach text.

?text - Sucht rückwärts nach text.

n - Sucht noch einmal in der gleichen Richtung.

N - Sucht noch einmal in der umgekehrten Richtung.

[addr] s/from/to/ [g] - Ersetzt einmalig from durch to. Mit addr kann ein Bereich angegeben werden, in dem die Aktion durchgeführt werden soll. Die Zeilennummern sind

durch ein Komma zu trennen. g führt die Aktion an jeder gefundenen Stelle durch. Beispiel: :2,10s/a/b/g ersetzt in den Zeilen von 2 bis 10 alle Buchstaben a durch b.

u - Undo, widerruft die letzte Änderung.

U - Stellt die aktuelle Zeile wieder her.

~ - Ändert Groß-/Kleinschreibung.

. - Wiederholt das letzte Kommando.

Zeilenumbruch entfernen

Unter DOS/Windows erstellte Dateien erscheinen unter GNU/Linux häufig mit doppelten Zeilenendezeichen, da auf diesen Systemen eine Kombination aus CR+LF verwendet wird. CR (Carriage Return) hat den ASCII-Code 13, LF (Line Feed) den ASCII-Code 10.

Um diese Dateien zu konvertieren, gibt es verschiedene Lösungsansätze. Im vi benutzt man das Kommando :se ff=unix. Sollten noch überflüssige **RETURN**-Zeichen (^M) am Zeilenende auftauchen, so lassen sich diese mittels :1,$s/^M//g entfernen. Dabei muss man das ^M durch Drücken von **STRG+V**, gefolgt von **STRG+M**, eingeben. **STRG+V** dient dazu, die nachfolgend gedrückte Tastenkombination direkt in den Text zu übernehmen.

3.20 Dateisysteme

3.19 vi 3.21 hdparm

Als Dateisystem bezeichnet man auf einem Debian GNU/Linux-System (und auf allen anderen Unix-Systemen) den kompletten Verzeichnisbaum ab dem „Wurzel“-Verzeichnis / (auch „root“-Verzeichnis genannt). Als Dateisystem wird aber auch die Organisationsform der Daten auf einem Medium (Festplatte, Diskette) bezeichnet, die von Betriebssystem zu Betriebssystem unterschiedlich ist.

Jedes physikalische Gerät, auf dem Sie Daten speichern wollen, müssen Sie zunächst mit einem Dateisystem versehen. Wenn Sie verschiedene Partitionen auf einem Medium erstellen, kann jede dieser Partitionen mit einem anderen Dateisystemtyp versehen werden. Jedes Betriebssystem verwendet mindestens einen eigenen Dateisystemtyp, viele verwenden auch mehrere oder können mit verschiedenen Typen von Dateisystemen umgehen.

Häufig sind im Debian-Umfeld Kombinationen aus Linux- und Windows-Dateisystemen anzutreffen. Debian GNU/Linux kann mit einer großen Zahl von Dateisystemen umgehen. Sie können somit sehr einfach Ihre Daten von anderen Betriebssystemen auf Ihr Debian GNU/Linux-System übernehmen.

Unter Debian GNU/Linux gibt es nur einen einzigen Verzeichnisbaum (beginnend mit /). In diesem sind als Unterverzeichnisse alle physikalischen Geräte zu finden. Es werden keine Buchstaben zur Identifikation der Geräte benutzt.

Mit Ausnahme des Root-Dateisystems (/ welches beim Systemstart automatisch angemeldet wird) müssen Sie alle weiteren Dateisysteme erst einmal in das System einbinden. Dabei kann, wie bereits beschrieben, jedes physikalische Gerät über mehrere Partitionen verfügen. Jedes dieser Dateisysteme wird im System in einem Verzeichnis (dem so genannten „Mount-Point“)„ eingehängt“.

Das ist so einfach, wie es sich anhört: Sie können ein Verzeichnis, welches als Mount-Point für eine Partition genutzt werden kann, mit dem Kommando mkdir erzeugen. Danach kann mit dem Kommando mount die Partition an die gewünschte Stelle im Dateisystem eingehängt werden. Bei der Installation von Debian GNU/Linux von CD-ROM wurde das Dateisystem (/cdrom) bereits benutzt und vom Installationsprogramm ins System eingebunden. Hierbei wurde ein Link von dem entsprechenden Device auf das neue Device /dev/cdrom angelegt. Sie müssen sich somit nicht das Device Ihres CD-ROM-Laufwerks merken (oder herausfinden), sondern können diesen Link benutzen. Weiterhin wurde bei der Installation das Verzeichnis /cdrom angelegt. Sie können nun eine

eingelegte CD leicht mit dem Kommando mount /dev/cdrom /cdrom in das Dateisystem einhängen.

Der Mount-Point kann anstelle von /cdrom jedes andere, beliebige Verzeichnis sein.

Beachten Sie jedoch, dass Verzeichnisse, an deren Stelle Sie ein Dateisystem mounten möchten, keine weiteren Dateien enthalten sollten. Das Mounten eines Dateisystems funktioniert auch, wenn sich bereits Dateien in dem Verzeichnis befinden, Sie können lediglich nicht mehr auf diese Dateien zugreifen. Die Dateien werden nicht gelöscht, sie werden praktisch von dem gemounteten Dateisystem „überlagert".

In der Praxis reicht dieses Wissen jedoch nicht lange aus. Sicher werden Sie irgendwann den Wunsch haben, den Festplattenplatz Ihres Systems zu erweitern. Der erste Schritt ist natürlich der mechanische Einbau der Festplatte. Schon hierbei (eigentlich schon beim Kauf der Festplatte) müssen Sie sich für den Anschluss an einem der beiden IDE-Busse oder am SCSI-Hostadapter entscheiden.

Bei einem IDE-System notieren Sie sich, ob Sie die Festplatte am ersten (primary) oder am zweiten (secondary) Bus anschließen und ob die Festplatte als erste (Master) oder zweite (Slave) am jeweiligen Bus betrieben wird.

Am SCSI-Bus notieren Sie sich die ID der Festplatte und kontrollieren, ob (und wenn ja, mit welcher ID) noch weitere Geräte angeschlossen sind. Beachten Sie hierbei auch externe Geräte!

Im nächsten Schritt müssen Sie mindestens eine Partition auf der neuen Festplatte anlegen. Diese kann den gesamten Festplattenplatz in Anspruch nehmen; Sie können aber auch mehrere, kleine Partitionen anlegen. Unter Debian GNU/Linux haben Sie die Auswahl zwischen zwei Programmen: fdisk und cfdisk. Wir werden im Folgenden cfdisk vorstellen, da seine Oberfläche ansprechender ist.

Ermitteln Sie zunächst mit Hilfe der zuvor notierten Daten das entsprechende Device für die neue Festplatte. Für IDE-Laufwerke ist die Bezeichnung folgende:

/dev/ hda - Master am primären Bus

/dev/ hdb - Slave am primären Bus

/dev/ hdc - Master am sekundären Bus

/dev/ hdd - Slave am sekundären Bus

Dabei ist unerheblich, ob es sich um eine Festplatte oder um ein CD-ROM-Laufwerk handelt.

Bei SCSI-Laufwerken ist die Bestimmung etwas anders. Zunächst ist zu beachten, dass zwischen Festplatten, CD-ROMs/CD-Brennern und anderen Geräten unterschieden wird. Die Gerätedateien für SCSI-Festplatten werden mit /dev/sdX bezeichnet, wobei X für

einen Buchstaben steht, angefangen bei a und dann aufsteigend nach SCSI-ID zugeordnet. Hier ein denkbares Beispiel:

/dev/sda - SCSI-Festplatte mit der kleinsten ID (z.B.: 0)

/dev/sdb - SCSI-Festplatte mit der mittleren ID (z.B.: 2)

/dev/sdc - SCSI-Festplatte mit der größten ID (z.B.: 3)

SCSI-CD-ROMs oder -CD-Brenner werden ähnlich bezeichnet. Die Gerätedatei wird als /dev/scdX bezeichnet, hier steht X für eine Zahl, beginnend bei 0. Beispielsweise:

/dev/ scd0 - SCSI-CD-ROM-Brenner (z.B. ID: 3)

/dev/ scd1 - SCSI-CD-ROM-Brenner (z.B. ID: 5)

Natürlich müssen/können Sie auf einer CD-ROM kein Dateisystem anlegen, dies sollte hier nur der Anschauung dienen.

Nachdem Sie nun das Ihrer Festplatte entsprechende Device bestimmt haben, können Sie cfdisk mit dem entsprechenden Device als Option starten, beispielsweise: cfdisk /dev/hdb. Bei einer neuen Festplatte werden nach dem Programmstart keinerlei Partitionen angezeigt. In diesem Beispiel, mit einer ca. 25 Gbyte großen Festplatte, wurde eine einzige Partition mit einem Linux Extended2-Dateisystem (ext2) angelegt.

Die erste Partition wird als /dev/hdb1 angelegt, eine zweite würde /dev/hdb2 genannt werden und so weiter.

Sie können innerhalb von cfdisk mit den Cursortasten navigieren und mit der **RETURN**-Taste den ausgewählten Menüpunkt ansteuern.

Wenn Sie alle gewünschten Partitionen (oder auch nur eine einzige) angelegt haben, können Sie nun das eigentliche Dateisystem auf der Partition erzeugen. Hierzu steht Ihnen unter Debian GNU/Linux das Kommando mke2fs zur Verfügung. Auch diesem Kommando müssen Sie natürlich wieder angeben, welche Festplatte und vor allem auch welche Partition Sie mit dem Dateisystem beschreiben wollen. Für unser Beispiel beginnen wir mit der ersten Partition auf unserer Festplatte, also dem Device /dev/hdb1:

```
sushi:~# mke2fs /dev/hdb1 mke2fs 1.15, 18-Jul-1999 for EXT2 FS
0.5b, 95/08/09 Filesystem label= OS type: Linux Block size=4096
(log=2) Fragment size=4096 (log=2) 128256 inodes, 256032 blocks
12801 blocks (5.00%) reserved for the super user First data block=0 8
block groups 32768 blocks per group, 32768 fragments per group
16032 inodes per group Superblock backups stored on blocks:
        32768, 98304, 163840, 229376  Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

Die bei Ihnen angezeigten Werte werden, je nach verwendeter Festplatte, von den hier gezeigten abweichen.

Nun können Sie die frisch formatierte Partition zu Ihrem Dateisystem hinzufügen: mount /dev/hdb1 /mnt und mit dem Kommando df überprüfen, ob nichts schief gelaufen ist.

Die Ausgabe könnte auf einem System mit mehreren Festplatten wie folgt aussehen:

```
sushi:~# df Filesystem          1k-blocks      Used Available Use%
Mounted on /dev/hda1              5767132   4352356   1121804  80% /
/dev/hdb1            16247612  14355348   1066928  94% /home/ftp
/dev/hdc1            24597980  21574360   1774080  93%
/home/ftp/debian /dev/hda2           18263244  15157492   2178016
88% /home/ftp/images
```

In dieser Auflistung sollten Sie dann auch die neu eingebundene Festplattenpartition finden. Die Option -h zum Kommando df erzeugt eine etwas lesbarere Ausgabe, indem der Speicherplatz in Mega- bzw. Gigabyte-Größen angegeben wird.

Plattenplatzbelegung in Farbe

Mit dem Programm pydf (dieses muss gesondert installiert werden) kann die Belegung des Plattenplatzes farbig und in grafischer Form erfolgen.

```
wasabi:~# pydf Filesystem            Size      Used     Avail  Use% [        ] Mounted on
/dev/hdc3             758M       86M      671M  11.0 [#       ] /  tmpfs                 58M         0
58M   0.0 [        ] /dev/shm /dev/hdc1              90M     7604k       83M   8.0 [#       ] /boot
/dev/hdc5            1902M      913M      988M  48.0 [####    ] /usr /dev/hdc6            1902M
1579M      323M  83.0 [######## ] /var /dev/hdc7             949M      916k      948M   0.0 [
] /tmp /dev/hdc8              49G       36G       13G  73.0 [######  ] /home
```

3.20.2.1 /etc/fstab - Im Detail

Vielleicht werden Sie nach einem Neustart des Systems bemerkt haben, dass die neu eingebundene Festplatte nicht automatisch ins System eingebunden wird. Wenn Sie möchten, dass bestimmte andere Partitionen zusätzlich zu / (root) ins System eingebunden werden, so müssen Sie diese in die Datei /etc/ fstab (für: „file system table“) aufnehmen. Weiterhin ist es sinnvoll, dort auch Dateisysteme einzutragen, die zwar nicht automatisch gemountet werden sollen, auf die Sie aber trotzdem schnellen Zugriff haben möchten, beispielsweise CD-ROMs, die öfter gewechselt werden.

Nach der Basisinstallation von Debian GNU/Linux sind bereits einige Einträge in der Datei /etc/fstab vorhanden:

```
# /etc/fstab: static file system information. # # <file system>     <mount
point>   <type>  <options>   <dump >  <pass> /dev/hda1            /
```

```
ext2    defaults    0       1 /dev/hda2             none            swap    sw          0
0 proc              /proc           proc    defaults    0       0
```

Bei der Installation wurden (mindestens) das Root-Dateisystem (/) sowie eine Swap-Partition angelegt. Der dritte Eintrag dient dem „virtuellen“ Verzeichnis /proc, das zur Laufzeit des Systems diverse Informationen zum System und zur Hardware enthält. Dieses verbraucht keinen Festplattenplatz.

Die erste Spalte beschreibt das Device und die zu mountende Partition. Die zweite Spalte verweist auf das Verzeichnis im Dateisystem, an der die Partition eingebunden werden soll. Beachten Sie, dass dieses Feld bei einer Swap-Partition mit dem Text „none“ anstatt eines Verzeichnisses gefüllt wird. Die dritte Spalte beschreibt den Typ des Dateisystems. Eine Beschreibung der weiteren Spalten finden Sie weiter unten, übernehmen Sie die Werte erst einmal wie gezeigt.

Um die im vorigen Abschnitt beschriebene Festplatte /dev/hdb automatisch ins System einzubinden, erweitern Sie die Datei um folgenden Eintrag:

```
/dev/hdb1           /mnt            ext2    defaults    0       2
```

Weitere nützliche Einträge sind:

```
/dev/hdd            /cdrom          iso9660 ro          0       0 /dev/fd0
/floppy         auto    noauto,sync 0       0
```

Der erste Eintrag ermöglicht es Ihnen, eine CD-ROM einfach mit dem Kommando mount /cdrom anstatt mount /dev/cdrom /cdrom einzubinden. Gleiches gilt für die zweite Zeile, diesmal aber für das Diskettenlaufwerk. Um mit DOS-formatierten Disketten umzugehen, sehen Sie sich das Paket mtools an.

Wie Sie schon gesehen haben, sind die Einträge in der Datei /etc/fstab in tabellarischer Form angeordnet:

```
# /etc/fstab: static file system information. # # <file system>     <mount
point>   <type>  <options>   <dump >  <pass> /dev/hda1           /
```

```
ext2    defaults    0       1 /dev/hda2             none            swap    sw      0
0 proc              /proc           proc    defaults    0       0
```

Zeilen, die mit einem Kommentarzeichen (# - „Gartenzaun“, „Bahnübergang“ oder im englischen „Hashmark“ genannt) beginnen, können Sie ignorieren, das System tut dies auch.

Mit den ersten drei Spalten haben wir uns ja schon kurz beschäftigt, diese enthalten die Einträge für die Partition, den Mount-Punkt und den Dateisystemtyp.

Die fünfte Spalte wird von dem Programm dump benutzt, um festzustellen, wann diese Partition gesichert werden soll. dump und restore dienen zur Sicherung von Daten.

Die sechste Spalte wird beim Systemstart von dem Programm fsck ausgewertet. Damit wird festgestellt, in welcher Reihenfolge die Dateisysteme beim Systemstart geprüft werden sollen. Die Root-Partition (/) sollte hier den Wert 1 erhalten. Dateisysteme, die nicht überprüft werden sollen, wie zum Beispiel swap oder CD-ROMs, bekommen den Wert 0, alle anderen Dateisysteme bekommen eine 2.

Nein, das ist kein Fehler im Text... Zur vierten Spalte kommen wir jetzt. Diese bedarf einiger Erklärungen. Die Einträge in der vierten Spalte werden beim Mounten des Dateisystems benutzt. Sie können hier einen (im einfachsten Fall den Text: default) oder mehrere Werte angeben.

async oder sync - Stellt die Datenübertragung (I/O) auf den synchronen oder asynchronen Modus. Im synchronen Modus werden alle veränderten Daten sofort auf das Medium geschrieben; der asynchrone Modus speichert diese zwischen und schreibt später auf das Medium.

ro oder rw - Mountet das Dateisystem zum „Nur-lesen“ (ro - read-only) oder zum Lesen und Schreiben (rw - read-write). Wenn Sie keine Änderungen an einem Dateisystem vornehmen wollen, so können Sie dieses „ro“ mounten, um versehentliche Änderungen zu verhindern. Ebenso ist dieser Modus für CD-ROMs geeignet.

auto oder noauto - Beim Systemstart oder wenn Sie das Kommando mount -a benutzen, werden alle Dateisysteme gemountet, die Sie mit dem Eintrag auto versehen haben. Dateisysteme, die nicht dauerhaft zur Verfügung stehen, wie zum Beispiel Disketten oder CD-ROMs, sollten den Eintrag noauto bekommen. Sie verhindern so eine Fehlermeldung beim Systemstart, müssen diese Dateisysteme allerdings dann von Hand einbinden.

dev oder nodev - nodev ignoriert die Gerätedateien auf dem gemounteten Dateisystem.

user oder nouser - Normalerweise können Dateisysteme nur vom Administrator (root) in das System eingebunden werden. Mit dem Wert user erlauben Sie auch normalen Benutzern das Mounten von Dateisystemen. Sie können so beispielsweise allen Benutzern den Zugriff auf das Diskettenlaufwerk oder das CD-ROM-Laufwerk erlauben.

exec oder noexec - Erlaubt oder verbietet das Ausführen von Programmen, die sich auf diesem Dateisystem befinden.

suid oder nosuid - Wertet das Suid-Bit aus oder nicht.

defaults - Der eigentlich wichtigste Wert, den dieses Feld annehmen kann. Aktiviert die Optionen: rw, dev, suid, exec, auto, nouser, async. Sie können einzelne Werte mit weiteren Parametern überschreiben.

3.20 Dateisysteme

3.19 vi 3.21 hdparm

3.21 hdparm

3.20 Dateisysteme 3.22 Internationalisierung und Lokalisierung

3.21.1 Optionen

3.21.2 Einbinden von hdparm

Mit dem Programm hdparm lassen sich Festplattenparameter auslesen und verändern.

Die Syntax zu diesem Befehl lautet:

hdparm - get/set hard disk parameters - version v5.2 Usage: hdparm [options] [device] .. Options: -a get/set fs readahead -A set drive read-lookahead flag (0/1) -b get/set bus state (0 == off, 1 == on, 2 == tristate) -B set Advanced Power Management setting (1-255) -c get/set IDE 32-bit IO setting -C check IDE power mode status -d get/set using_dma flag -D enable/disable drive defect-mgmt -E set cd-rom drive speed -f flush buffer cache for device on exit -g display drive geometry -h display terse usage information -i display drive identification -I detailed/current information directly from drive -Istdin similar to -I, but wants /proc/ide/*/hd?/identify as input -k get/set keep_settings_over_reset flag (0/1) -K set drive keep_features_over_reset flag (0/1) -L set drive doorlock (0/1) (removable harddisks only) -M get/set acoustic management (0-254, 128: quiet, 254: fast) (EXPERIMENTAL) -m get/set multiple sector count -n get/set ignore-write-errors flag (0/1) -p set PIO mode on IDE interface chipset (0,1,2,3,4,...) -P set drive prefetch count -q change next setting quietly -Q get/set DMA tagged-queuing depth (if supported) -r get/set readonly flag (DANGEROUS to set) -R register an IDE interface (DANGEROUS) -S set standby (spindown) timeout -t perform device read timings -T perform cache read timings -u get/set unmaskirq flag (0/1) -U un-register an IDE interface (DANGEROUS) -v defaults; same as -mcudkrag for IDE drives -V display program version and exit immediately -w perform device reset (DANGEROUS) -W set drive write-caching flag (0/1) (DANGEROUS) -x tristate device for hotswap (0/1) (DANGEROUS) -X set IDE xfer mode (DANGEROUS) -y put IDE drive in standby

mode -Y put IDE drive to sleep -Z disable Seagate auto-powersaving
mode -z re-read partition table

hdparm ist ein Kommandozeilen Programm, mit dessen Hilfe verschiedene Festplattenparameter im Linux-Kernel verändert werden können. Dieses Programm benötigt mindestens einen Kernel ab der Version 1.2.13; dies sollte aber auf einem aktuellen Debian System kein Problem sein.

Wenn keine weiteren Optionen angegeben werden, so werden bei einem IDE-Gerät automatisch die Optionen -acdgkmnru und bei einem SCSI-Gerät die Optionen -gr benutzt.

Im Einzelnen bewirken die Optionen Folgendes:

-A schaltet die „read-lookahead"-Funktion einer IDE-Festplatte ein oder aus. Normalerweise ist diese Funktion bereits aktiviert.

-c zeigt den Status oder aktiviert den (E)IDE-32-Bit-I/O-Support. Die Parameter hier für sind: 0, um den Support zu deaktivieren, und 1, um den Support zu aktivieren. Der Wert 3 aktiviert den 32-Bit-Support mit einer speziellen Sync-Sequenz, die von einigen Chipsets benötigt wird. Diese Option funktioniert mit fast allen Chipsets, bringt jedoch etwas mehr Overhead mit sich. Wird kein Parameter angegeben, so wird die aktuelle Einstellung angezeigt.

-C zeigt den aktuellen „Power-Mode-Status" einer IDE-Festplatte an. Dieser kann folgende Werte annehmen: „unknown" - Das Laufwerk unterstützt dieses Kommando nicht; „active"/„ idle" - Normalbetrieb; „standby" - Low Power Modus, die Festplatte dreht sich nicht oder „schläft" sogar komplett. Die Optionen -S, -y, -Y und -Z können zur Veränderung der IDE-Power-Modes verwendet werden.

-d zeigt, ob für das genannte Gerät der DMA-Modus benutzt wird, oder aktiviert/deaktiviert den Modus. Das Flag „using_dma" funktioniert nur mit wenigen Kombinationen aus Festplatte und Controllern. Beispielsweise wird beim Intel Triton Chipset der Bus-Master-DMA-Modus in Verbindung mit vielen Festplatten unterstützt. Wenn möglich, sollte die Option -X34 zusammen mit -d1 benutzt werden, um sicherzustellen, dass das Laufwerk selbst den Multiword DMA Modus 2 unterstützt.

-f synchronisiert und speichert den Cache-Puffer der Festplatte beim Beenden des Programms. Dieser Vorgang wird auch bei den Optionen -t und -T durchgeführt.

-g zeigt die Festplattengeometrie, also Angaben zu Zylinder, Köpfe, (Sektoren), die Größe (in Sektoren) des Laufwerkes und den Offset (in Sektoren) des Devices vom Anfang der Festplatte.

-h zeigt die Hilfe-Informationen.

-i zeigt die Daten zur Identifikation der Festplatte, die beim Booten ausgelesen wurden, falls diese verfügbar sind. Diese Funktion wird nur von neueren Festplatten unterstützt. Die Informationen können veraltet sein, wenn das System seit dem Start viel beschäftigt war.

-I zeigt die aktuelle Identifikation der Festplatte an. Wie die Option -i; die Informationen werden allerdings neu ausgelesen.

-k liest oder setzt das Flag keep_settings_over_reset für das Laufwerk. Wenn diese Option gesetzt ist, werden die Optionen -dmu über einen (Soft-) Reset hinaus gespeichert.

-K setzt das Flag keep_features_over_reset. Das Setzen dieser Option rettet die Einstellungen -APSWXZ über einen (Soft-) Reset hinaus. Diese Option wird nicht von allen Laufwerken unterstützt.

-m zeigt den aktuellen Zustand oder setzt den IDE-Block-Modus. Ein Wert von 0 deaktiviert diese Funktion.

-p versucht, das IDE-Interface-Chipset auf den gewünschten PIO-Modus zu setzen oder den bestmöglichen PIO-Modus einzustellen.

-q unterdrückt die Ausgabe von Meldungen für die folgenden Optionen. Dies kann eingesetzt werden, wenn hdparm aus einer Datei beim Systemstart heraus aufgerufen wird. Diese Option hat keine Auswirkungen auf die Optionen -i, -v, -t oder -T.

-r liest oder setzt das „Read-Only-Flag“.

-S setzt die Standby-Zeit des Gerätes. Nach der eingestellten Zeit in Sekunden fährt die Festplatte in den Ruhezustand. Ein Wert von 0 schaltet diese Funktion aus.

-T führt einen Benchmark mit Lesezugriffen auf den Cache durch. Um sinnvolle Ergebnisse zu bekommen, sollte ein Test zwei- bis dreimal auf einem System mit möglichst wenig anderen laufenden Prozessen durchgeführt werden.

-t führt einen Benchmark mit Lesezugriffen auf das Gerät durch. Um sinnvolle Ergebnisse zu bekommen, sollte ein Test zwei- bis dreimal auf einem System mit möglichst wenig anderen Prozessen durchgeführt werden.

-u liest oder setzt das „interrupt-unmask“-Flag. Dies kann auf vielen Systemen die Reaktionszeit des Systems verbessern, ist aber mit Vorsicht einzusetzen, da nicht alle Chipsets diese Funktion richtig umsetzen können. Ein Wert von 1 aktiviert diese Funktion.

-v zeigt die aktuellen Einstellungen an, mit Ausnahme von -i. Wenn keinerlei Optionen auf der Kommandozeile angegeben werden, wird ebenfalls -v angenommen. Bei einem SCSI-Gerät entspricht dies den Optionen -gr, bei einem IDE-Gerät den Optionen -acdgkmnru.

-W aktiviert oder deaktiviert bei einem IDE-Laufwerk die Funktion write-caching Dies ist Standardmäßig deaktiviert.

-X setzt den IDE-Übertragungsmodus bei neueren (E)IDE/ATA2-Laufwerken. Diese Option wird meistens zusammen mit -d1 benutzt, um den DMA-Modus von/zu einem Laufwerk mit Chipsets, die dies unterstützen (beispielsweise Intel 430FX Triton), zu aktivieren. Dort können dann auch mit -X34 „ Multiword-DMA-mode2“-Übertragungen aktiviert werden. Die Option -X66 aktiviert UltraDMA-Mode2-Übertragungen.

-y aktiviert sofort den Energiesparmodus bei einem IDE-Laufwerk.

-Y aktiviert sofort den Energiesparmodus mit dem geringsten Stromverbrauch bei einem IDE-Laufwerk.

-Z deaktiviert die Stromsparfunktionen bei einigen Seagate-Laufwerken, die diese Funktion nur fehlerhaft implementiert haben.

hdparm wird über das Init-Skript /etc/init.d/hdparm bei jedem Systemstart aktiviert. Anpassungen der Parameter werden in der Datei /etc/hdparm.conf vorgenommen.

3.21 hdparm

3.20 Dateisysteme 3.22 Internationalisierung und Lokalisierung

3.22 Internationalisierung und Lokalisierung

3.21 hdparm 3.23 Tastaturbelegung

Die im Debian-Projekt eingesetzte Software ist in weiten Bereichen bereits an die Belange der nicht-englischsprachigen Benutzer angepasst. Diese Aufgabe ist für die Entwickler und Übersetzer nicht leicht zu bewerkstelligen, da in den seltensten Fällen eine wörtliche Übersetzung ausreichen würde. In den Programm-Menüs könnte man die meisten Ergebnisse noch akzeptieren, die Übersetzung der Online-Hilfe, der Manpages und der Dokumentation stellt aber eine echte Herausforderung dar.

Ein Vorteil, den die freie Software in diesem Bereich ausspielen kann, ist aber sicher die weltweit verteilte Entwicklung. Hierdurch wird es jedem Benutzer, auch Nicht-Programmierern, ermöglicht, an Entwicklungsprojekten teilzunehmen.

Doch wie benutzt man nun die bereits übersetzten Programme? Die Steuerung der gewünschten Sprache erfolgt über eine Umgebungsvariable. Wenn Sie nach der Installation keine Veränderungen an Ihrem System vorgenommen haben, so sollte die Variable LANG den Wert C haben: Damit wird die Sprache Englisch verwendet. Sie können den Wert der Variablen mit dem Kommando echo auf der Kommandozeile prüfen.

```
fr@sushi:~$ echo $LANG C
```

Das Verhalten lässt sich relativ gut an weit verbreiteten Kommandos prüfen, bei denen man davon ausgehen kann, dass bereits eine Übersetzung in viele Sprachen existiert. Das Kommando ls ist recht gut geeignet.

```
fr@sushi:~$ ls --help Usage: ls [OPTION]... [FILE]... List information about the FILEs (the current directory by default). Sort entries alphabetically if none of -cftuSUX nor --sort.  -a, --all  do not hide entries starting with . ...
```

Das war zu erwarten und sollte so weit bekannt sein... Setzen Sie nun die Variable auf einen anderen Wert. Da Sie dieses Buch lesen, wird Sie vielleicht die deutsche Sprache interessieren:

```
fr@sushi:~$ export LANG=de_DE fr@sushi:~$ ls --help Benutzung: ls [OPTION]... [DATEI]... Auflistung von Informationen der DATEIen (Standardvorgabe ist das momentane Verzeichnis). Alphabetisches
```

Sortieren der Einträge, falls weder -cftuSUX noch --sort angegeben. -a, --all Einträge, die mit . beginnen, nicht verstecken. ...

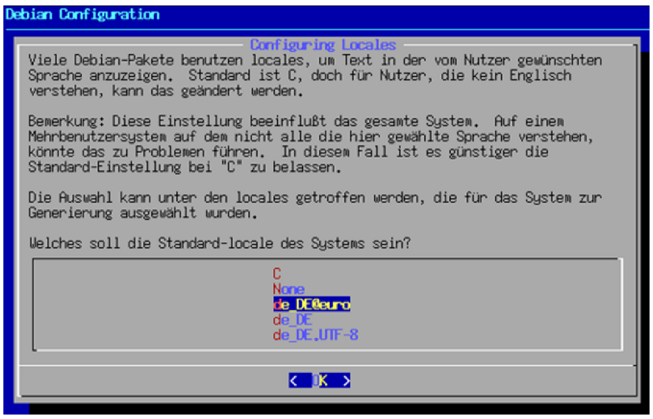
Diese Einstellungen sind in dieser Form lediglich für die aktuelle Shell gültig. Wenn die Spracheinstellungen für das gesamte System geändert werden sollen, so können die Veränderungen in den Konfigurationsdateien im Verzeichnis /etc vorgenommen werden. Eleganter ist es aber, die Debian Werkzeuge für diese Aktionen zu benutzen.

Das Paket locales enthält zunächst die notwendigen Dateien, um verschiedene Sprachen auf einem Debian System nutzen zu können. Diese nehmen relativ viel Platz auf der Festplatte ein. Normalerweise werden auf einem System nur die wenigsten der enthaltenen Sprachen benötigt. Wenn das Paket localepurge installiert wird, so wird regelmäßig geprüft, welche dieser Dateien wirklich benötigt werden; alle anderen werden gelöscht.

Während der Installation des Pakets locales wird die Konfiguration des Pakets vorgenommen. Nachträglich kann dies mittels dpkg-reconfigure locales geschehen.

Zunächst sind die auf dem System gewünschten Sprachen und Zeichensätze zu wählen. Die hier gemachten Angaben werden in der Datei /etc/locale.gen gespeichert.

Einträge in dieser Datei können auch von Hand vorgenommen werden; in diesem Fall ist nach der Änderung das Kommando locale-gen auszuführen. Im hier gezeigten Beispiel werden alle mit de_DE beginnenden Werte ausgewählt, insbesondere auch die Euro-Variante. Die Benutzung des Euro-Zeichens ist unter Debian GNU sehr einfach möglich; Informationen hierzu finden Sie im Abschnitt Euro-Symbol .



Im nächsten Schritt kann nun aus den eben gewählten Sprachen die für das gesamte System gültige Sprache gewählt werden. Dieser Wert sollte nur dann auf einen anderen Wert als „C“ gestellt werden, wenn Sie sicher sind, dass dies keinen Einfluss auf wichtige Funktionen hat. Sind beispielsweise Skripts zur Systemadministration geschrieben worden, die Ausgaben von Programmen auswerten, so kann es nun passieren, dass diese Ausgaben in einer anderen Sprache erscheinen. Das Skript kann diese Ausgaben nun nicht mehr korrekt auswerten. In diesem Fall wird die Einstellung besser auf dem Wert „C“ belassen.

Setzen von Umgebungsvariablen

Skripts sollten immer selbst dafür sorgen, dass alle Umgebungsvariablen richtig gesetzt sind. Um die Spracheinstellungen zu setzen, kann in jedes Skript die Zeile export

LANG=C eingefügt werden.

3.22 Internationalisierung und Lokalisierung

3.21 hdparm 3.23 Tastaturbelegung

3.23 Tastaturbelegung

3.22 Internationalisierung und Lokalisierung Kapitel 4. Paketmanagement

Neben der verwendeten Sprache ist es natürlich für den Benutzer vor dem Rechner auch von entscheidender Wichtigkeit, wie die Tasten auf der Tastatur tatsächlich belegt sind. Bei der Installation von Debian wird schon sehr früh das gewünschte Tastatur-Layout abgefragt. Es wird davon ausgegangen, dass die zur Installation verwendete Belegung auch im späteren Betrieb des Systems verwendet werden soll.

Sicher ist dies in den meisten Fällen richtig. Jedoch kann es wüschenswert sein, die Belegung temporär oder dauerhaft umzustellen. Die Gründe hierfür können vielfältig sein. Sei es, dass Sie ein System an einen Kunden im Ausland liefern, oder es wurde eine falsche Tastatur bei der Installation verwendet...

Die Belegung der Tastatur ist - anders als die Einstellung der verwendeten Sprache für die Anwendungen - eine systemweite Einstellung für die Konsole (grafische Oberflächen bieten einfache Möglichkeiten, zwischen verschiendenen Tastaturbelegungen zu wechseln).

Um für die Dauer einer Anmeldung am System die Belegung der Tastatur zu wechseln, kann das Kommando loadkeys <SPRACHE> verwendet werden. Um eine deutschsprachige Tastaturbelegung zu wählen, ist dazu „de“ als Sprache anzugeben. Mittels „en“ wechselt man zu einer englischsprachigen Belegung. Dies wird nicht als Einstellung gespeichert, sondern gilt nur für diese eine Sitzung, in der das Kommando aufgerufen wurde.

Um eine dauerhafte Änderung der Tastaturbelegung zu bewirken, muss das Kommando dpkg-reconfigure console-data verwendet werden (auf einigen Systemen funktioniert auch dpkg-reconfigure console-common). Es kann dann aus verschiedenen Tastaturbelegungen gewählt werden. Die Angaben werden dauerhaft für das gesamte System gespeichert. Änderungen werden aber erst nach der nächsten Anmeldung am System für andere Benutzer wirksam.

3.23 Tastaturbelegung

3.22 Internationalisierung und Lokalisierung | Kapitel 4. Paketmanagement

Kapitel 4. Paketmanagement

3.23 Tastaturbelegung 4.2 Release

Inhaltsverzeichnis

Um weitere Pakete auf einem Debian GNU/Linux-System zu installieren, stehen mehrere Programme zur Verfügung. Neben dpkg, das auf der Kommandozeile ausgeführt wird, ist dselect das älteste von diesen mit einer Benutzeroberfläche. dselect stellt ein überaus mächtiges Programm mit vielen Möglichkeiten und allen Freiheiten dar. Dies hat aber leider nicht bei allen Anwendern (gerade bei Einsteigern) zur Beliebtheit beigetragen. dselect bietet eine sehr ausführliche Online-Hilfe, aber mal ehrlich: Wer liest so etwas schon? Aber wer über einige wenige Kenntnisse von dselect verfügt, wird sehen, dass alles ganz einfach ist, und die wird Mächtigkeit dieses Werkzeugs schnell zu schätzen wissen.

apt stellt die nächste Generation der Debian GNU/Linux-Installationsprograme dar. Mit der Version 3.0 von Debian GNU/Linux ist die komplette Umstellung des Paketmanagements auf apt erfolgt.

aptitude ist nun die bevorzugte Methode zur Paketverwaltung auf der Konsole. aptitude unterstützt die meisten Kommandozeilenoperationen von apt-get und hat bessere Fähigkeiten bei der Auflösung von Abhängigkeiten bewiesen.

„Secure APT“ ist seit Debian 4.0 Bestandteil der Distribution. Secure APT sorgt für zusätzliche Sicherheit von Debian GNU/Linux-Systemen, indem starke Kryptografie und digitale Signaturen zum Verifizieren heruntergeladener Pakete verwendet werden. Das Programm apt-key wird genutzt, um dem Schlüsselring von apt neue Schlüssel hinzuzufügen. Standardmäßig wird nur der Schlüssel zum Signieren des Debian Archives aus dem Paket debian-archive-keyring berücksichtigt. In der Standardkonfiguration wird apt nun eine Warnung ausgeben, falls Pakete aus Quellen heruntergeladen werden, die nicht authentifiziert werden konnten.

Unabhängig davon, welches Programm Sie zur Installation benutzen - Sie müssen in jedem Fall auswählen, welche Pakete Sie installieren möchten. Dies ist bei Debian GNU/Linux keine leichte Aufgabe, denn mittlerweile umfasst die Distribution einige tausend Pakete! Um Ihnen die Auswahl zu erleichtern, wurden von den Entwicklern so genannte „ task“-Pakete zusammengestellt. Diese „Gruppierung“ von Paketen vereinfacht Ihnen die Selektion. Wählen Sie beispielsweise das Paket task-gnome-desktop aus, so werden alle nötigen Pakete für einen GNOME-Desktop ausgewählt und installiert. Weitere Informationen zu diesen Paketen finden Sie in „Taskpakete“.

apt-get oder gnome-apt oder auch das bereits erwähnte dselect können zum Auswählen und Installieren einzelner Pakete benutzt werden.

Die Paketverwaltung unter Debian GNU/Linux stellt eines der Highlights dieser Distribution dar. Wir wollen hier nicht auf jedes Detail eingehen, aber trotzdem ein paar Informationen zu den grundsätzlichen Vorgängen bei der Installation eines Pakets festhalten.

Jedes Programmpaket enthält in der Regel einige verschiedene Dateien, die zur Funktion dieses Programms benötigt werden. Hierzu gehören natürlich das Programm selbst, die Bibliotheken, Dateien mit Daten wie zum Beispiel Grafiken für Spiele, die Anleitung

(Manpages) usw. Bei der Installation eines Programms muss sichergestellt sein, dass alle Dateien am richtigen Platz im Verzeichnisbaum installiert werden.

Wenn Sie später ein Programm wieder von Ihrem System entfernen wollen, ist es ebenfalls wichtig zu wissen, welche Dateien zu einem Programm gehören und wo sie sich befinden. Weiterhin ist festzustellen, ob Dateien installiert wurden, die mittlerweile von anderen Programmen benutzt werden. In diesem Fall darf das Paket unter Umständen nicht entfernt werden.

Ebenso kann der Fall eintreten, dass ein Programm auf den neuesten Stand gebracht werden soll. Bei diesem Vorgang werden Dateien aktualisiert und eventuell auch einige Dateien gelöscht, die von der neuen Version nicht mehr benötigt werden. Auch dies muss so durchgeführt werden, dass nicht andere Programme danach nicht mehr funktionsfähig sind.

Das Debian GNU/Linux-Paketsystem wacht über alle diese Vorgänge und hält Ihr System immer in einem benutzbaren Zustand.

Debian GNU/Linux organisiert die Pakete in einer festgelegten Hierarchie. Unter anderem werden Bereiche mit allgemeiner, freier Software (main), die den DFSG (Debian Free Software Guidelines - „Debian Free Software Guidelines“) entspricht, und Bereiche mit Software, die nicht unter solchen freien Lizenzen steht (non-free), unterschieden. Als weiterer Bereich steht contrib zur Verfügung, hier finden sich Pakete mit freier Software, die aber auf nicht-freien Paketen aufbauen. Die Verantwortlichen für das Debian-Projekt haben sich entschieden, den Bereich „non-free“ zukünftig nicht mehr zu unterstützen. Es wird versucht, die Autoren der Software-Pakete zu überzeugen, eine freie Lizenz zu verwenden. Gelingt dies nicht, so wird das entsprechende Paket in absehbarer Zeit nicht mehr auf den Debian Servern verfügbar sein.

Die nächste Ebene beschreibt die Architektur, also die Prozessorfamilie, auf der die Binärpakete laufen. Auch die Sourcen (Quellcodes) zu den Paketen sind in dieser Ebene angesiedelt. In der letzten Ebene finden sich diverse Verzeichnisse, in denen die eigentlichen Pakete zu Gruppen, wie zum Beispiel „Games“ oder „X11“, zusammengefasst werden.

Im Folgenden finden Sie eine Übersicht der verschiedenen Ebenen.

Kapitel 4. Paketmanagement

◁ 3.23 Tastaturbelegung △ 4.2 Release ▷

4.2 Release

Kapitel 4. Paketmanagement 4.3 Distribution

Von Debian werden zu jedem Zeitpunkt vier Release-Varianten angeboten: Die zuletzt veröffentlichte Version heißt „stable“. Dort sind alle Pakete sehr gut getestet, und die jeweils aktuellsten Sicherheitsupdates sind sehr schnell verfügbar. Mit einer solchen „stable“-Version ist in etwa alle zwei Jahre zu rechnen. Aktualisiert wird diese Version durch „Release“-Versionen (Unterversionen wie beispielsweise 3.1 R2), die in unregelmäßigen Abständen herausgegeben werden. Sicherheitsupdates sind für diese Version sehr zeitnah verfügbar. Dieses stabile Release von Debian sollte als Basis für alle produktiven Systeme eingesetzt werden.

Darüber hinaus gibt es zwei weitere Zweige, die zwar verfügbar, aber nicht vom Debian Team für den produktiven Betrieb freigegeben sind: „unstable“ (an diesem Release wird laufend weiterentwickelt) und „testing“, das der Sammelplatz für das nächste offizielle Release ist. „testing“ setzt sich aus Paketen zusammen, die nach 10 bis 14 Tagen im Zweig „unstable“ keine für das Release entscheidenden Fehler gezeigt haben und dann in den „testing“-Zweig übernommen werden. Die Pakete in „testing“ haben eine gute Stabilität und können für den laufenden Betrieb auf nicht geschäftskritischen Systemen eingesetzt werden. Hier sollte man aber auf kleinere Probleme und Ausfallzeiten gefasst sein. Sehr aktuelle Pakete finden Sie aber eher im „unstable“-Bereich. Sicherheitsupdates werden für diese Releases nicht garantiert. Häufig sind entsprechende Pakete verfügbar; verlassen Sie sich aber nicht darauf.

Für einige Pakete wird darüber hinaus manchmal „Experimental“ benutzt. Dieser Zweig enthält neuere Software-Versionen, die jedoch sehr große Änderungen mit sich bringen. Beispielsweise werden dort sehr neue und aktuelle Software-Pakete in die Debian Distribution integriert, oder es werden größere Debian Pakete komplett neu strukturiert. Sie sollten diesen Zweig nur einsetzen, wenn Sie gute Kenntnisse über das Debian Paketmanagement haben (etwa nach der Lektüre dieses Buches ;-)). Für diese Pakete gibt es keinen festgelegten Wert, wie sie in die Distribution aufgenommen werden.

Die vier Release-Varianten sind auf den Servern durch „Links“ mit den jeweiligen Release-Namen verbunden. So ist beispielsweise momentan, kurz nach der Release von „Sarge“, „stable“ ein Link auf „Sarge“, „testing“ ein Link auf „Etch“ und „unstable“ ist immer ein Link auf „sid“. „sid“ steht für „still in development“.

```
ftp> ls -l 227 Entering Passive Mode (128,101,80,133,219,159) 150
Here comes the directory listing. lrwxrwxrwx    1 1176     1176           5
Jun 01 19:38 Debian3.0r6 -> woody lrwxrwxrwx    1 1176     1176
5 Jun 06 18:33 Debian3.1r0 -> sarge -rw-rw-r--    1 1176     1176
449 Jun 06 17:41 README drwxrwsr-x    5 1176     1176         4096
Jul 30 19:34 etch drwxrwsr-x    5 1176     1176         4096 Jul 30 19:34
```

```
etch-proposed-updates lrwxrwxrwx    1 1176     1176          23 Jan 13
2005 experimental -> ../project/experimental lrwxrwxrwx    1 1176
1176           5 Jun 06 18:33 oldstable -> woody lrwxrwxrwx    1 1176
1176          22 Jun 06 18:33 proposed-updates -> sarge-proposed-
updates drwxrwsr-x    5 1176     1176        4096 Jun 16 19:34 sarge
drwxrwsr-x    5 1176     1176       12288 Jul 30 19:33 sarge-proposed-
updates drwxrwsr-x    5 1176     1176        4096 Jul 30 19:35 sid
lrwxrwxrwx    1 1176     1176           5 Jun 06 18:33 stable -> sarge
lrwxrwxrwx    1 1176     1176          22 Jun 06 18:33 stable-proposed-
updates -> sarge-proposed-updates lrwxrwxrwx    1 1176     1176
4 Jun 06 18:33 testing -> etch lrwxrwxrwx    1 1176     1176          21
Jun 06 18:33 testing-proposed-updates -> etch-proposed-updates
lrwxrwxrwx    1 1176     1176           3 Jan 13  2005 unstable -> sid
drwxrwsr-x    5 1176     1176        4096 May 31 20:55 woody
drwxrwsr-x    5 1176     1176       57344 Jun 06 17:42 woody-proposed-
updates
```

Kurz vor einem neuen Release gibt es für kurze Zeit noch ein weiteres Release: „frozen“. Es stellt den eingefrorenen Zustand einer „testing“-Distribution dar, kurz bevor diese zu „stable“ wird. In einem „frozen“-Status, kurz vor der Freigabe eines neuen Release, wird kein neuer Code mehr akzeptiert. Es werden lediglich bei Bedarf Bugfixes eingepflegt. Es wird ein neuer Verzeichnisbaum im dists/-Verzeichnis angelegt und einem neuen Kodenamen zugeordnet. Die eingefrorene Distribution durchläuft nun einige Monate lang Tests mit zwischenzeitlichen Updates und Zwischenversionen, die „Test-Zyklen“ genannt werden.

4.2 Release

Kapitel 4. Paketmanagement 4.3 Distribution

4.3 Distribution

4.2 Release 4.4 Architektur

Die so genannten „Distributionen“ innerhalb der gesamten Debian GNU/Linux-Distribution bezeichnen verschiedene Bereiche, die sich durch die Lizenzen unterscheiden, unter denen die Pakete in diesen Bereichen stehen.

Main - Dies ist der Hauptbestandteil von Debian. Diese Pakete sind unter einem Copyright veröffentlicht, das eine freie Weiterverteilung ermöglicht, und enthalten den vollständigen Quellcode.

Contrib - Die Pakete in diesem Verzeichnis sind für sich frei, jedoch benötigen sie nicht-freie (Non-Free) Software oder Bibliotheken, um zu funktionieren. Sie können daher nicht in den Bereich „main“ von Debian GNU/Linux einfließen.

Non-Free - Pakete in diesem Verzeichnis müssen nicht unbedingt Geld kosten, jedoch haben sie einige Bedingungen, die eine Weiterverteilung der Software einschränken. Beispielsweise kann ein Autor die Verbreitung der Software auf CD-ROM untersagen.

Non-US - Diese Pakete dürfen nicht aus den USA exportiert werden; es handelt sich meist um Verschlüsselungssoftware. Einige sind aufgrund der Lizenzen ebenfalls nicht-freie Software. Dieser Bereich wurde mit der Release von „Sarge“ (Debian Version 3.1) bereinigt und ist entfallen. Alle Lizenzen der Pakete wurden in Zusammenarbeit mit den Entwicklern angepasst, und die Pakete sind in den Bereich „main“ eingeflossen. Pakete, bei denen das nicht möglich war, wurden aus der Debian Distribution entfernt.

4.3 Distribution

4.2 Release 4.4 Architektur

4.4 Architektur

4.3 Distribution 4.5 Gruppen

Hier finden Sie eine Übersicht einiger von Debian GNU/Linux unterstützten beziehungsweise in der Entwicklung befindlichen Architekturen.

Bitte beachten Sie auch, dass nicht alle Architekturen auf CD-ROM im Handel erhältlich sind. Am einfachsten sind CDs für die i386-Architektur zu bekommen. Dies ist sowohl im Heimbereich als auch bei professionellen Anwendern die am weitesten verbreitete Hardware-Plattform.

Eine komplette Übersicht finden Sie unter http://www.debian.org/ports/.

Intel x86 / IA-32 (i386)

Die erste Architektur und nicht direkt eine Portierung. Mit dieser Architektur begann alles. Linus Torvalds entwickelte auf Basis von Minix die ersten Zeilen Sourcecode zu Linux auf einem i386-Prozessor. Debian unterstützt alle IA-32-Prozessoren, hergestellt von Intel, AMD, Cyrix und weiteren Herstellern.

Intel IA-64 (ia64)

Zum ersten Mal offiziell mit Debian 3.0 freigegeben. Dieses ist die Portierung auf Intels 64-Bit-Architektur.

S/390 (s390)

Zum ersten Mal offiziell mit Debian 3.0 freigegeben. Dieses ist die Portierung auf IBM S/390-Server.

DEC Alpha (alpha)

Das erste Release erfolgte mit Debian 2.1. Eine der länger bestehenden Portierungen und ziemlich stabil.

ARM (arm)

Eine neue Portierungsbestrebung, motiviert durch Rebels (ehemals Corel) interessante „NetWinder“-Maschine.

Motorola 68k (m68k)

Das erste Release dieser Portierung erfolgte mit Debian 2.0. Der Debian m68k-Port läuft auf einer großen Bandbreite von Computern, die auf der Motorola 68k-Prozessorfamilie basieren - im Besonderen die Sun3-Workstation-Familie, die Apple Macintosh Personal-Computer und die Atari und Amiga Personal-Computer, aber auch einige aus dem Industriebereich stammenden VME-Bus-Boards.

MIPS (mips)

Zum ersten Mal offiziell mit Debian 3.0 freigegeben. Debian wird auf die MIPS-Architektur portiert, die in SGI-Computern (debian-mips - big-endian) und Digital Decstations (debian-mipsel - little-endian) verwendet wird.

Motorola/IBM PowerPC (powerpc)

Diese Portierung läuft auf vielen Apple Macintosh PowerMac-Rechnern und kann auch auf den meisten Motorola-Computern laufen. Ältere Macintosh-Rechner benutzen die m68k-Prozessoren; diese werden nicht als PowerMac bezeichnet. Debian/PowerPC wurde mit Debian 2.2 (Potato) das erste Mal offiziell veröffentlicht. Die Portierung läuft auf vielen der Apple Macintosh PowerMac-Modelle, den CHRP- und den PReP-Rechnern.

HP PA-RISC (hppa)

Zum ersten Mal offiziell mit Debian 3.0 freigegeben. Dieses ist eine Portierung auf die PA-RISC-Architektur von Hewlett-Packard, die sich in einem weit entwickelten Stadium befindet.

Sun SPARC (sparc)

Zum ersten Mal mit Debian 2.1 veröffentlicht. Diese Portierung läuft sowohl auf der SPARCstation-Familie von Workstations als auch auf einem Teil ihrer Nachfolger in der Sun4-Architektur.

Sun UltraSPARC (sparc64)

Dies ist auch der Beginn einer Portierung auf die Sun UltraSPARC-(sun4u-) Workstation-Familie. Dieser 64-Bit-Prozessor hat den Vorteil der Rückwärtskompatibilität zu seinem Vorgänger, so dass die UltraSPARC-Portierung in der Lage sein wird, Sparc-Binärfiles auszuführen. Dieses Projekt braucht zurzeit Entwickler. Wenn Sie einen UltraSPARC-basierenden Computer haben und helfen möchten, senden Sie bitte eine E-Mail an die Liste unter debian-ultralinux@lists.debian.org, um Ihre Mitarbeit anzubieten.

Debian GNU/Hurd (hurd-i386)

GNU/Hurd ist ein völlig neues Betriebssystem, das von der GNU-Gruppe aufgebaut wird. In der Tat ist GNU/Hurd die letzte Komponente, die es möglich macht, ein komplettes GNU-Betriebssystem aufzubauen, und Debian GNU/Hurd wird ein solches (vielleicht sogar das erste) GNU-Betriebssystem sein. Das gegenwärtige Projekt ist auf der i386-Architektur aufgebaut, aber Sie können erwarten, dass die anderen wenig später folgen werden.

Debian GNU/NetBSD (netbsd-i386 und netbsd-alpha)

Dies ist eine Portierung des Debian Betriebssystems inklusive apt, dpkg und GNU-Software auf den NetBSD-Kernel. Sie befindet sich im Augenblick in einem sehr vorläufigem Stadium, aber da NetBSD ein Produktionslevel-Kernel ist, sollte sich die Verwendbarkeit von Debian GNU/NetBSD sehr rasch entwickeln. Im Augenblick ist Debian

GNU/NetBSD für Intel x86 am weitesten fortgeschritten, aber die Arbeit an der Unterstützung für Alpha-basierende Computer hat bereits begonnen.

Debian GNU/FreeBSD (freebsd-i386)

Dies ist eine Portierung des Debian Betriebssystems auf den FreeBSD-Kernel. Sie befindet sich im Augenblick in einer Vorentwicklungsphase.

Debian Beowulf

Obwohl nicht wirklich eine Portierung, so wird Beowulf doch ein Ersatz für viele Großrechner in Forschung und Entwicklung sein. Deshalb scheint hier der richtige Platz zu sein, um es zu erwähnen. Dieses Projekt arbeitet daran, Beowulf-Cluster unter Debian laufen zu lassen.

4.4 Architektur

4.3 Distribution 4.5 Gruppen

4.5 Gruppen

4.4 Architektur 4.6 Backports

Jedes Paket der Debian Distribution kann vom Betreuer (Maintainer) einem bestimmten Bereich zugeordnet werden. Diese Bereiche beschreiben den hauptsächlichen Einsatzort bzw. den Einsatzbereich. Innerhalb eines Frontends zur Paketinstallation (beispielsweise dselect) können dann die Pakete entsprechend dieser Zuordnung ausgewählt werden.

Diese Struktur lässt sich auf den FTP-Servern oder den Debian CD-ROMs nicht ohne Weiteres zurückverfolgen, wenn nach einem Paket gesucht wird. Die Bereiche sind in den Dateien Packages.gz vermerkt und werden vom Frontend auf die tatsächlichen Pfade umgesetzt. Die Pakete selbst befinden sich in einer einfachen Dateisystemstruktur, die wie folgt aufgebaut ist: /pool/main/a/paketname/paket.deb.

Doch nun zu den Bereichen:

Administration - Programme zur Systemadministration

Base - Teile des Basissystems, wie zum Beispiel verschiedene Kernel

Communication - Terminalprogramme usw.

Development - Diverse Programmiersprachen, Compiler, Interpreter, Header-Dateien (C und C++) usw.

Documentation - HOWTOs, FAQs und andere Dokumentation sowie Programme, um diese zu lesen

Editors - Textverarbeitungsprogramme, Editoren für Programmierer

Electronics - Simulatoren für elektronische Schaltungen usw.

Games - Spiele

Graphics - Grafikprogramme

Ham Radio - Programme für Amateurfunker

Interpreters - Interpreter wie Perl, Python und Tcl/Tk

Libraries - Bibliotheken, die von verschiedenen Programmen genutzt werden

Mail - Alles rund um E-Mail. Mailserver, Mailprogramme usw.

Mathematics - Mathematische und wissenschaftliche Programme

Miscellaneous - Diverses, was sonst nirgends hineinpasst

Network - Netzwerkserver und Clientprogramme

Newsgroups - Software für öffentliche Diskussionsforen

Old Libraries - Ältere Versionen von Bibliotheken

Other Operating Systems and File Systems - Zugriff auf andere Betriebs- oder Dateisysteme

Shells - Verschiedene Shells

Sound - Alles für den guten Ton

TeX - Donald Knuths Schriftsatzprogramm

Text Processing - Werkzeuge zum Umgang mit Textdateien

Utilities - Verschiedene Werkzeuge

Web - Programme für das WWW, Server und Clients

X Window - X-Server, Window-Manager und anderes

4.5 Gruppen

4.4 Architektur 4.6 Backports

4.6 Backports

4.5 Gruppen 4.7 Das Debian Paketformat - .deb

Backporting (Rückportierungen) bezeichnet den Prozess der Rückportierung von Software-Paketen aus einem neueren Release in ein älteres Release. Es gibt zwei wesentliche Gründe dafür, Pakete aus anderen Releases zu portieren:

Durch den langen Release-Zyklus der stabilen Debian Version ist es manchmal wünschenswert, auf Software-Pakete aus „testing“ oder „unstable“ zuzugreifen. Dies ist leider nicht direkt ohne Weiteres möglich, da Abhängigkeiten zwischen den Paketen nicht erfüllt werden. Mitunter sind beispielsweise Bibliotheken nicht mehr in den benötigten Versionen in einer neueren Release vorhanden.

Auch kann eine individuelle Konfiguration bei der Übersetzung bestimmter Software-Pakete gewünscht sein. Hierfür ist aber nur in wenigen Fällen ein wirklicher Backport notwendig, es kann auch auf die Version aus dem verwendeten Release zurückgegriffen werden.

Detaillierte Informationen darüber, welche Schritte für das Backporting notwendig sind, finden Sie auf der Seite Debian Backports.

4.6 Backports

4.5 Gruppen 4.7 Das Debian Paketformat - .deb

4.7 Das Debian Paketformat - .deb

4.6 Backports 4.8 debconf

Das Debian Paketformat beinhaltet eine Vielzahl von Informationen zu jedem Paket, um sicherzustellen, dass sich jedes einzelne perfekt in das System integriert. Detailierte technische Informationen zum Aufbau des Paketformates finden sich im Abschnitt Debian Pakete im Detail. Debian Paketnamen enden immer mit .deb; somit können sie leicht von anderen Dateien oder von Paketen aus anderen Distributionen unterschieden werden. Das bekannteste Merkmal des Debian Paketformats sind die Abhängigkeiten (dependency) zwischen den Paketen.

Abhängigkeiten zwischen den Paketen erlauben es zum Beispiel einzelnen Programmen, auf gemeinsame Bestandteile anderer Pakete zuzugreifen; meist sind dies Libraries (Systembibliotheken). Dies verhindert ein unnötiges, doppeltes Installieren von Dateien. Auf einem durchschnittlichen System kann so die Zahl der installierten Dateien deutlich reduziert werden.

Betrachten wir zunächst einmal den einfachsten Fall: Ein Paket benötigt zwingend ein zweites Paket, um zu funktionieren. Das Paket mail-crypt ist eine Erweiterung zu Emacs, um E-Mail mit PGP zu verschlüsseln. Wenn PGP nicht installiert ist, wird auch mail-crypt nicht funktionieren. Somit wurde vom Debian Paket-Betreuer (Maintainer) dem Paket die Abhängigkeit zu PGP mitgegeben. Ebenso bedingt mail-crypt die Installation von emacs: Da es eine Erweiterung dazu ist, macht es keinen Sinn, dieses Paket allein zu installieren.

Weiterhin sind in den Paketbeschreibungen auch Konflikte zwischen den Paketen festgelegt. So ist es unter Debian GNU/Linux nicht möglich, zwei oder mehrere der Programme exim, smail, sendmail, postfix oder qmail zu installieren, da diese alle das virtuelle Paket mail-transport-agent zur Verfügung stellen. Die Abhängigkeiten erlauben es, dass genau ein Programm, das für den Mail Transport zuständig ist, installiert werden kann. Dann haben Sie aber die freie Auswahl zwischen den verfügbaren Paketen.

Virtuelle Pakete ermitteln

Das Kommando apt-cache showpkg exim zeigt unter „Provides“ an, welches virtuelle Paket von exim zur Verfügung gestellt wird.

Nun wäre es denkbar, dass ein Programm zum Erstellen von E-Mail (beispielsweise mutt), da es ja eine Möglichkeit benötigt, auch Mails auszuliefern, beispielsweise von smail abhängt.

Damit würde man dem Benutzer von mutt vorschreiben, welchen MTA („Mail Transfer Agent“) er zu benutzen hat. Debian GNU/Linux-Pakete gehen auch hier einen besonderen Weg. Von den Betreuern (Maintainer) der Pakete wird zusätzlich ein „virtueller“ Name festgelegt. Die Programme exim, smail, sendmail und qmail beispielsweise verfügen noch über die Information, dass Sie einen „mail-transport-agent“ zur Verfügung stellen. Auf diese Weise ist das Paket mutt von einem virtuellen Namen und nicht von einem konkreten Paket abhängig. Somit bleibt Ihnen die freie Auswahl zwischen einem dieser Programme.

Das Debian Paketsystem überwacht zu jeder Zeit alle diese Abhängigkeiten und sorgt dafür, dass Ihr System in einem sicheren, lauffähigen Zustand bleibt.

4.7 Das Debian Paketformat - .deb

◁ 4.6 Backports △ 4.8 debconf ▷

4.8 debconf

4.7 Das Debian Paketformat - .deb 4.9 dselect

debconf ist ein zentrales System zur Verwaltung von Debian Paketen. Es gibt auch ein selten verwendetes Programm namens debconf, dies soll jedoch an dieser Stelle nicht weiter betrachtet werden (siehe hierzu debconf). debconf bietet eine einheitliche Schnittstelle zur Konfiguration von Debian Paketen, der Benutzer kann dabei zwischen verschiedenen Frontends wählen. debconf kann vor der Installation von Paketen die notwendigen Parameter erfragen, so dass die weitere Installation einer großen Anzahl von Paketen unbeaufsichtigt erfolgen kann. Hierzu ist es notwendig, dass APT in der Version 0.5 oder höher installiert ist, weiterhin wird das Paket apt-utils benötigt.

Sollen Pakete nicht vor der Installation konfiguriert werden, so kann dies in der Datei /etc/apt/apt.conf.d/70debconf angepasst werden. Wie dort beschrieben, ist einfach eine Zeile mit einem Kommentarzeichen zu versehen. Ein einzelnes Paket kann vor der Installation gezielt vorkonfiguriert werden, hierzu dient das Kommando dpkg-preconfigure. Dieses findet sich im Paket apt-utils.

Ist nun ein Paket installiert und wurden alle debconf-Fragen während der Installation beantwortet, so kann es nach einiger Zeit wünschenswert sein, Anpassungen an der Konfiguration vorzunehmen. Ein einfaches Neuinstallieren des Pakets führt dabei nicht zum Erfolg, da die Antworten in der debconf-Datenbank gespeichert wurden. Bei einer neu Installation des Pakets versucht das System zunächst, diese Antworten in der Datenbank zu finden. Ist dies erfolgreich, so werden keine Fragen zur Konfiguration gestellt. Um nun ein Paket, beispielsweise debconf selbst, neu zu konfigurieren, kann das Kommando dpkg-reconfigure eingesetzt werden.

```
dpkg-reconfigure debconf
```

Bei diesem Vorgang werden alle Fragen erneut angezeigt, die auch ursprünglich bei der ersten Installation von debconf gestellt wurden. Dies muss natürlich mit Administratorrechten ausgeführt werden.

debconf kann unterschiedliche Frontends zur Konfiguration von Paketen verwenden, diese können vom Benutzer ausgewählt werden. Folgende Frontends stehen zur Auswahl:

dialog

Das standardmäßig verwendete Frontend auf Basis von whiptail oder dialog. Eine einfache grafische Oberfläche auf Textbasis.

readline

Einfache, rein textbasierte Oberfläche. Es wird eine Frage gestellt, und es müssen Werte eingegeben werden. Sehr gut geeignet, um remote auf einem System zu arbeiten.

noninteractive

Dies ist eigentlich gar kein Frontend, denn es werden keinerlei Fragen angezeigt. Alle Werte werden mit den Vorgabewerten aus dem Paket belegt. Es werden allerdings E-Mails an den Administrator gesendet, mit dem Hinweis, dass Pakete konfiguriert wurden bzw. dass noch Einstellungen vorzunehmen sind. Dies ist das beste Frontend, um automatische Installationen durchzuführen.

gnome

Ein modernes Frontend auf Basis der GNOME- und GTK-Bibliotheken. Dieses Frontend lässt sich nur sinnvoll verwenden, wenn lokal ein X-Server installiert ist oder wenn eine entsprechende DISPLAY Variable gesetzt wurde, um den Konfigurationsdialog auf einem anderen System anzuzeigen. Ist das Anzeigen dieses Frontends nicht möglich, so wird auf ein einfacheres Frontend zurückgegriffen.

kde

Ein modernes Frontend auf Basis der QT-Bibliotheken. Dieses Frontend lässt sich nur sinnvoll verwenden, wenn lokal ein X-Server installiert ist oder wenn eine entsprechende DISPLAY-Variable gesetzt wurde, um den Konfigurationsdialog auf einem anderen System anzuzeigen. Ist das Anzeigen dieses Frontends nicht möglich, so wird auf ein einfacheres Frontend zurückgegriffen.

editor

Das Frontend für Unix-Freaks, die alles in einem Texteditor bearbeiten möchten. Hierbei werden nicht die Konfigurationsdateien zu einem Paket aufgerufen, vielmehr können debconf-Parameter in dem Editor verändert werden.

web

Dieses Frontend verhält sich wie ein Webserver; alle Einstellungen können über einen beliebigen Browser vorgenommen werden. Dieses Frontend befindet sich noch in der Entwicklung.

Das von debconf verwendete Frontend kann bei einer Rekonfiguration von debconf gesetzt werden. Soll dagegen das Frontend nur für eine einzelne Aktion geändert werden, so kann auch die Umgebungsvariable DEBIAN_FRONTEND auf den gewünschten Wert gesetzt werden.

```
DEBIAN_FRONTEND=readline apt-get install slrn
```

Es ist auch möglich, den Kommandos dpkg-reconfigure und dpkg-preconfigure die Option --frontend=readline zu übergeben und auf diesem Wege das Frontend zu bestimmen.

Über die debconf-Prioritäten kann die „Tiefe“ der Fragen bestimmt werden. Jede Frage zu einer Konfigurationsoption ist mit einer Priorität versehen, über die der Betreuer des Pakets steuern kann, wie wichtig die Frage für das Funktionieren des Pakets ist. Wenn es Ihnen nicht so darauf ankommt, ein Paket bis ins letzte Detail anzupassen, so kann debconf dazu veranlasst werden, lediglich die essenziell wichtigen Fragen zu stellen.

Die Prioritäten sind dabei (in aufsteigender Priorität):

low

Sehr einfache Fragen, die mit Voreinstellungen belegt sind, die in den meisten Fällen sinnvoll sind.

medium

Normale Fragen mit sinnvollen Vorgaben.

high

Fragen, die keine Vorgaben haben.

critical

Fragen, die unbedingt beantwortet werden müssen.

Es werden immer nur Fragen mit einer Priorität angezeigt, die der gewünschten Tiefe entspricht, oder die in der Wertigkeit höher liegen. Die Priorität kann gesetzt werden, indem debconf neu konfiguriert wird oder indem die Option --priority=medium den Kommandos dpkg-reconfigure und dpkg-preconfigure übergeben wird. Alternativ kann auch die Umgebungsvariable DEBIAN_PRIORITY auf die gewünschte Tiefe gesetzt werden.

debconf benutzt eine sehr flexible Struktur, um die Ergebnisse der Fragen in einer Datenbankstruktur abzulegen. In der Datei /etc/debconf.conf wird diese Anbindung an die Datenbank konfiguriert. Im Normalfall liegt diese Datenbank in Form von ASCII-Dateien auf dem lokalen Dateisystem im Verzeichnis /var/cache/debconf/. In der Konfigurationsdatei finden Sie Beispiele, wie eine Datenbank via NFS oder LDAP genutzt werden kann.

debconf im Backup

Bei der Planung eines Backups für das neue System sollte auch das Verzeichnis /var/cache/debconf/ gesichert werden. Obwohl im Verzeichnis /var/cache/ lediglich Daten liegen, die laufend verändert werden, so erspart das Sichern dieses Verzeichnisses die Fragen bei der erneuten Installation des Systems.

Bei der automatischen, unbeaufsichtigten Installation oder Aktualisierung von vielen Systemen kann es wünschenswert sein, nicht die vorgegebenen Antworten zu nutzen. Vielmehr ist eine individuelle Anpassung, meist von nur sehr wenigen Werten, wünschenswert.

Es gibt verschiedene Möglichkeiten, um dies zu erreichen. So kann beispielsweise eine debconf-Datenbank mit den gewünschten Informationen aufgesetzt werden, die bei der Installation verwendet wird. Der einfachste Weg zu einer solchen, mit sinnvollen Werten gefüllten Datenbank ist es, auf einem anderen System eine komplette Installation mit den entsprechenden Einstellungen durchzuführen und dann die Datenbank zu kopieren. Alternativ kann auch dpkg-preconfigure eingesetzt werden, um eine Auswahl von Paketen zu konfigurieren, ohne diese zu installieren. Natürlich kann die komplette debconf-Datenbank auch mit einem Texteditor der Wahl erstellt werden.

Ist die Datenbank mit den benötigten Werten vorbereitet, so ist zu überlegen, wie die zu installierenden Systeme auf diese Datenbank zugreifen können. Soll ein zentraler LDAP-Server verwendet werden, so kann bei der Installation von Paketen auf diesen zugegriffen werden, um die dort hinterlegten Werte auszulesen. debconf benutzt die beiden Umgebungsvariablen DEBCONF_DB_FALLBACK und DEBCONF_DB_OVERRIDE, um den Zugriff auf eine andere Datenbank zu steuern. Ein Beispiel:

```
cat /var/cache/debconf/config.dat | ssh root@target
"DEBCONF_FRONTEND=noninteractive
DEBCONF_DB_FALLBACK=Pipe apt-get upgrade"
```

Dieses Kommando benutzt die lokale Datenbank, die via ssh auf das System gebracht wird. Dort wird die Datenbank als Fallback ausgelesen, falls lokal keine entsprechenden Antworten verfügbar sind. Dabei wird nur in die lokale Datenbank geschrieben; die via ssh nur temporär verwendete Datenbank wird ausschließlich gelesen.

```
ssh -R 389:ldap:389 root@target
"DEBCONF_DB_FALLBACK='LDAP{host:localhost}' apt-get
upgrade"
```

In diesem Beispiel wird eine über ssh getunnelte LDAP-Datenbank genutzt. Wieder wird diese Datenbank nur als Fallback, also read-only, eingesetzt.

```
scp config.dat root@target: ssh root@target
"DEBCONF_DB_FALLBACK='File{/root/config.dat}' apt-get upgrade
```

In diesem Beispiel wird zunächst die Datenbank mittels ssh kopiert. Danach wird über ssh ein Update angestoßen, bei dem die zuvor kopierte Datenbank benutzt wird.

In allen Beispielen wird die zentrale Datenbank als Fallback eingesetzt. Dies bedeutet, dass auf diesem Wege Antworten gesetzt werden können, die auf dem Zielsystem noch nie aufgetaucht sind, aber diese Werte werden ausschließlich genutzt, wenn in anderen Datenbanken keine Informationen gefunden werden: deshalb der Name „Fallback“. Um bereits gesetzte Werte zu überschreiben, ist die Umgebungsvariable DEBCONF_DB_OVERRIDE einzusetzen. Diese bewirkt ebenfalls, dass eine andere debconf-Datenbank abgefragt wird, dies jedoch, bevor andere Datenbanken zu Rate gezogen werden.

Detaillierte Informationen zur Entwicklung von Debian Paketen, die die debconf-Datenbank nutzen, finden Sie in der Manpage zu debconf-devel. Im Wesentlichen kommuniziert debconf mit den Skripten über Standard-In und -Output. Es wird dabei ein einfaches, SMTP-ähnliches Protokoll verwendet. Die Fragen selbst werden in einem Template abgelegt, das vom Format her einer Debian control-Datei ähnelt.

Folgende Umgebungsvariablen können von debconf genutzt werden:

DEBIAN_FRONTEND

Wird benutzt, um das Frontend zu bestimmen.

DEBIAN_PRIORITY

Setzt die minimale Priorität der zu stellenden Fragen.

DEBCONF_DEBUG

Aktiviert Meldungen zur Fehlersuche auf der Standard-Ausgabe. Es kann eine Facility oder ein regulärer Ausdruck (beispielsweise .* für alle Meldungen) angegeben werden. Mögliche Facilities sind:

user

Für den Benutzer interessante Informationen

developer

Für Entwickler interessante Informationen

db

Informationen zur Datenbank

DEBCONF_NOWARNINGS

Deaktiviert Warnungen von debconf. Fehlermeldungen werden weiterhin ausgegeben.

DEBCONF_TERSE

Wird dieser Wert auf YES gesetzt, so gibt debconf so wenige Meldungen wie möglich aus.

DEBCONF_DB_FALLBACK

Setzt eine zusätzliche Datenbank mit geringerer Priorität. Alle anderen Datenbanken (aus /etc/debconf.conf) werden bevorzugt behandelt. Wird eine Fallback-Datenbank über diese Umgebungsvariable gesetzt, so ist diese immer nur les-, aber nicht beschreibbar.

DEBCONF_DB_OVERRIDE

Setzt eine zusätzliche Datenbank mit hoher Priorität. Alle anderen Datenbanken werden erst benutzt, wenn in dieser Datenbank keine brauchbaren Informationen gefunden werden.

DEBCONF_SYSTEMRC

Beim Start von debconf wird eine ggf. vorhandene Datei ~/.debconfrc des Benutzers ausgewertet. Dies kann durch Setzen dieser Variablen verhindert werden.

DEBCONF_FORCE_DIALOG

Mit dieser Variablen lässt sich die Verwendung von dialog statt whiptail zur Anzeige des Frontends erzwingen.

DEBCONF_FORCE_XDIALOG

Das Setzen dieser Variablen führt dazu, dass Xdialog statt dialog oder whiptail verwendet wird.

Neben dem zentralen System zur Verwaltung von Konfigurationsparametern debconf (siehe debconf) gibt es auch ein Kommandozeilenprogramm gleichen Namens.

Das Programm debconf führt ein Programm oder Skript aus, das normalerweise bei der Installation eines Pakets ausgeführt wird. So lassen sich Skripte bei der Entwicklung von Paketen testen. Die Syntax lautet:

```
debconf [options] command [args]
```

-opackage, --owner=package

Hiermit kann bestimmt werden, zu welchem Paket das Skript gehört. So werden beispielsweise Benutzer- und Gruppenrechte korrekt gesetzt.

-ftype, --frontend=type

Bestimmt das zu verwendende debconf-Frontend.

-pvalue, --priority=value

Gibt die minimale Priorität der debconf-Fragen an, die noch angezeigt werden sollen.

Um nun beispielsweise ein Shell-Skript zu testen, kann folgendes Kommando genutzt werden:

```
DEBCONF_DEBUG=developer debconf my-shell-prog
```

Alternativ funktioniert aber auch:

```
debconf --frontend=readline sh -x my-shell-prog
```

Mit debconf-show lassen sich gezielt Informationen zu einzelnen Paketen aus der Debconf-Datenbank extrahieren. Die Syntax lautet:

```
debconf-show packagename [...] [--db=dbname] debconf-show --
listowners [--db=dbname] debconf-show --listdbs
```

Am häufigsten wird debconf-show sicher zusammen mit einem Paketnamen verwendet, um Informationen zu diesem Paket aus der Datenbank auszulesen. Hier am Beispiel Apache gezeigt:

```
wasabi:/home/fr# debconf-show apache * apache/server-name:
hoshi.homeunix.net * apache/document-root: /var/www *
apache/server-port: 80 * apache/enable-suexec: false * apache/init: true
* apache/server-admin: webmaster@wasabi
```

Fragen, die dem Benutzer bereits gestellt wurden, sind am Anfang der Zeile mit einem * gekennzeichnet.

--db=dbname

Name der abzufragenden Datenbank.

--listowners

Zeigt die „Besitzer“ der Fragen in der Datenbank an. Ein Besitzer entspricht immer einem Paketnamen.

--listdbs

Zeigt alle verfügbaren Datenbanken an.

debconf-get-selections liest die gespeicherten Informationen aus der Debconf-Datenbank und schreibt diese auf die Standard-Ausgabe. Die Syntax für diesen Befehl lautet:

```
debconf-get-selections [--installer]
```

Das Ausgabeformat kann direkt wieder vom Kommando debconf-set-selections eingelesen werden. Soll dies auf einem anderen System geschehen, so ist die Ausgabe in eine Datei zu schreiben und diese auf das zweite System zu transferieren. Dieses Prinzip wird auch vom Debian Installer genutzt, um eine automatische Installation zu erlauben.

Informationen, die während der Installation des Systems gesetzt werden, sind in einer getrennten Datenbank (/var/log/installer/cdebconf) abgelegt. Diese können über die einzige Option von debconf-get-selections - --installer - ausgelesen werden. Um alle Debconf-Einstellungen eines Systems in eine Datei zu schreiben, sind zwei Kommandos notwendig:

```
debconf-get-selections --installer > config.cfg debconf-get-selections >> config.cfg
```

debconf-set-selections setzt oder verändert Werte in der Debconf-Datenbank. Zusätzlich werden die so mit Werten vorbelegten Fragen auch als "gesehen" markiert, so dass diese Werte nicht während der Installation des Systems oder eines Paketes abgefragt werden. Die Syntax für dieses Kommando lautet:

```
debconf-set-selections dateiname
```

Wie jedes gute Kommandozeilenprogramm liest debconf-set-selections zunächst aus einer Datei. Wird kein Dateiname auf der Kommandozeile übergeben, so wird versucht, aus der Standard-Eingabe zu lesen.

Sollen die Informationen von einem System auf ein anderes übertragen werden, so kann folgendes Kommando verwendet werden, ein funktionierendes Netzwerk zwischen beiden Systemen wird vorausgesetzt:

```
debconf-get-selections | ssh systemname debconf-set-selections
```

Das Format der Datei ist recht einfach aufgebaut. Zeilen ohne Inhalt werden ignoriert, Zeilen, die mit dem Zeichen # beginnen, sind Kommentare und werden ebenfalls nicht ausgewertet. Alle anderen Zeilen werden interpretiert und müssen vier Werte enthalten, die jeweils durch Leerzeichen getrennt sind. Der erste Wert ist der Paketname, zu dem die Frage gehört. Der zweite Wert ist der Name der Frage. Der dritte Wert ist der Typ der Frage, und der vierte Wert ist die Antwort. Hier ein einfaches Beispiel:

```
# Comment out extension_dir config from /etc/php4/apache/php.ini?
php4    php4/update_apache_php_ini      boolean true
```

--verbose, -v

Ausführliche Ausgabe von Informationen

--checkonly, -c

Prüft die Eingabedatei auf Formatfehler, speichert keine Informationen in der Datenbank.

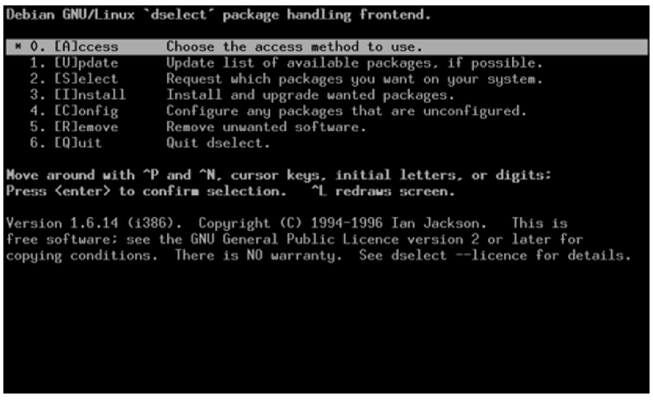
4.8 debconf

4.7 Das Debian Paketformat - .deb 4.9 dselect

4.9 dselect

4.8 debconf 4.10 APT und Verwandte

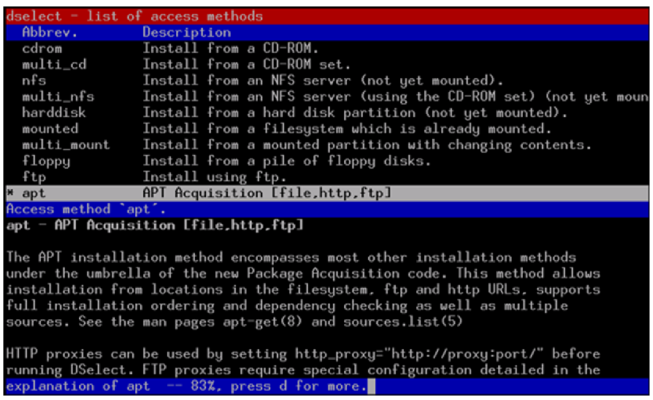
Das Frontend zur Paketinstallation dselect war bis zur Version 2.2 von Debian das wichtigste Werkzeug, um Pakete zu verwalten. Viele Benutzer waren aber mit der Benutzerführung nicht ganz zufrieden, so dass die Entwickler Alternativen wie apt-get und aptitude geschaffen haben. Doch noch immer ist dselect auf jedem Debian System installiert, und es gibt eine Reihe von alten Debian-Hasen, die auch heute noch gern dieses Programm einsetzen. dselect hat einige sehr mächtige Funktionen, und es lohnt sich durchaus, die folgenden Zeilen zu lesen...



Wenn Sie dselect zum ersten Mal aufrufen, ist es unbedingt notwendig, die Menüpunkte eins bis drei (Update, Select und Install) genau in dieser Reihenfolge aufzurufen.

Der nullte Punkt (Access) kann übersprungen werden; die Einstellungen an dieser Stelle sind bereits sinnvoll konfiguriert.

Mit diesem Menüpunkt legen Sie fest, von welchem Medium Sie die weiteren Pakete installieren wollen. Sie finden hier auch einen Eintrag für das Programm apt, das im nächsten Abschnitt beschrieben wird. Dieses sollte auch mit dselect die bevorzugte Installationsmethode sein.



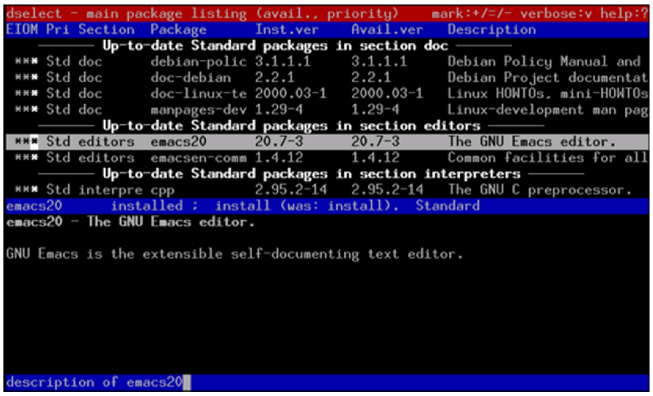
Wenn Sie andere Installationsmethoden verwenden wollen, lesen Sie in der Dokumentation zu dselect nach, welche Schritte dazu notwendig sind.

Update liest die Dateien mit den Paketinformationen ein und erstellt daraus eine Liste der verfügbaren Pakete.

Wenn Sie mehrere CDs zur Verfügung haben, werden Sie aufgefordert, diese nacheinander einzulegen.

Unter diesem Punkt können Sie weitere Pakete zur Installation auswählen. Sie bekommen zuerst eine kurze Hilfe angezeigt, die mit der **SPACE**-Taste verlassen werden kann. Sie können zu jeder Zeit mit der Taste ? ein Hilfe-Menü aufrufen. Debian GNU/Linux organisiert die Pakete nach verschiedenen Gruppen, wie zum Beispiel base oder x11. Sie können einzelne Pakete oder ganze Gruppen von Paketen mit der Taste + auswählen oder mit der Taste - von der Auswahl ausschließen.

Debian GNU/Linux umfasst einige Pakete, die unbedingt zum reibungslosen Betrieb benötigt werden, zum Beispiel die bash oder login. Diese Pakete können natürlich nicht entfernt werden. Wenn diese Pakete fehlen würden, würde Ihr System unbenutzbar werden.



Debian GNU/Linux überprüft schon bei der Auswahl von Paketen die Abhängigkeiten zwischen den mit + markierten Paketen. Sollte es zu Problemen kommen, sei es, weil Pakete sich im Betrieb stören würden oder weil ein anderes Paket unbedingt zur Installation des gewählten Pakets benötigt wird, so wird das Installationsprogramm Sie darauf hinweisen. Zwingend benötigte Pakete werden schon ausgewählt, überflüssige Pakete abgewählt. Weiterhin werden Ihnen auch Pakete zur Auswahl angeboten, deren Installation im Zusammenhang mit den ausgewählten Paketen sinnvoll ist.

Sollten Sie einmal nicht mit dem Vorschlag des Installationsprogramms einverstanden sein, können Sie den Vorschlag des Programms mit der Taste R wieder rückgängig machen.

Mit den Tasten o und I können Sie die Ansicht der Paketliste verändern. o beeinflusst die Sortierung und blendet die Informationen zu den Gruppen ein oder aus, I verändert die

Anzeige der Paketinformationen. Normalerweise wird der Bildschirm horizontal zur Hälfte geteilt, wobei in der oberen Hälfte die Paketliste und in der unteren Hälfte die Informationen zum gerade ausgewählten Paket zu sehen sind. I verändert die Länge der Paketinformationen oder blendet diese komplett aus.

Wenn Sie ein bestimmtes Paket suchen, können Sie mit der Taste / eine Suchroutine aktivieren und einen Suchbegriff eingeben. Mit der Taste \ können Sie den gleichen Begriff noch einmal suchen.

Wenn Sie Ihre Paketauswahl abgeschlossen haben, drücken Sie die Eingabetaste, und das Installationsprogramm überprüft noch einmal Ihre Auswahl auf eventuelle Konflikte oder Probleme. Sollten noch Konflikte bei der Auswahl auftreten, so wird das Installationsprogramm Sie darauf hinweisen. Sie können bei Konflikten auch wieder mit den Tasten + und - die Probleme beheben. Wenn Sie es trotz allem nicht schaffen, alle Konflikte zu beseitigen (was bei der Installation von Paketen aus dem Debian GNU/Linux-Entwicklungsbereich unstable passieren kann), so können Sie auch noch bestehende Konflikte mit der Taste Q akzeptieren und den Menüpunkt select verlassen. Dies sollte aber der Ausnahmefall sein.

Mit diesem Menüpunkt können Sie die ausgewählten Pakete auf Ihrem System installieren. Je nach der Installationsmethode, die Sie unter Access gewählt haben, werden Sie aufgefordert, die passenden CDs einzulegen; oder die Pakete werden über das Netzwerk installiert oder von einer anderen Partition installiert... Oder, oder, oder...

Die Installation der meisten Pakete erfolgt ohne weitere Rückfragen. Bei einigen Paketen werden aber weitere Angaben benötigt, beispielsweise bei der Installation von exim, das von Debian GNU/Linux zum Versenden von E-Mail verwendet wird. Auf die wichtigsten solcher Pakete werden wir in den weiteren Kapiteln eingehen. Bitte sehen Sie in den entsprechenden Kapiteln nach (exim ist im Kapitel „Internet“, Abschnitt „E-Mail“, beschrieben).

Sie können diesen Menüpunkt auswählen, wenn es während der Installation von Paketen zu Problemen gekommen ist. Für alle bereits installierten, aber noch nicht konfigurierten Pakete wird die dazugehörige Konfigurationsroutine aufgerufen. Sie können dies aber auch jederzeit aus einer Shell heraus mit dpkg --pending --configure durchführen.

Hiermit können Sie Pakete, die Sie unter Select mit der Taste „ - “ zum Löschen markiert haben, von Ihrer Festplatte entfernen. Ein großer Vorteil von Debian GNU/Linux ist es,

dass mit diesem Verfahren nur die Programme selbst entfernt werden; alle Konfigurationsdateien verbleiben auf der Festplatte. Wenn Sie zu einem späteren Zeitpunkt ein Paket, das Sie schon einmal installiert hatten, wieder installieren, so werden die noch vorhandenen Konfigurationsdateien verwendet.

Wenn Sie ein Programm inklusive der Konfigurationsdateien von Ihrem System entfernen wollen, benutzen Sie den Befehl: dpkg --purge irgendeinpaket.

Dieser Menüpunkt stellt die höchsten Ansprüche an den Benutzer: Wer möchte schon an dieser Stelle das Installationsprogramm beenden? Drücken Sie auf keinen Fall die Eingabetaste, bevor die Festplatte gut gefüllt ist...

4.9 dselect

4.8 debconf 4.10 APT und Verwandte

4.10 APT und Verwandte

4.9 dselect 4.11 Die Datei sources.list

APT („Advanced Package Tool“) stellt die nächste Generation der Debian GNU/Linux-Paketverwaltung dar.

In diesem Abschnitt wird zunächst auf die Konfiguration und Benutzung von Kommandozeilentools wie apt und dpkg sowie deren Verwandten eingegangen. Später werden dann auch grafische Frontends zur Paketverwaltung beschrieben.

Kommandovervollständigung

In der unter Linux verwendeten Shell bash lassen sich teilweise eingegebene Befehlsnamen durch Drücken der TAB-Taste vervollständigen. Zunächst funktioniert das ausschließlich für den Programmnamen und nicht für Kommandos und Optionen.

Mit dem Kommando source /etc/bash_completion können einige weitere Makros für die Shell geladen werden. Danach ist es möglich, auch Optionen von Befehlen automatisch zu vervollständigen (tippen Sie beispielsweise dpkg -TABTAB, so werden alle Optionen zum Befehl dpkg angezeigt). Dies funktioniert natürlich nicht nur mit Befehlen zur Verwaltung von Debian Paketen, sondern auch mit jedem anderen Befehl.

Darüber hinaus ist es möglich, auch Paketnamen, beispielsweise bei der Installation, zu vervollständigen. Tippen Sie hierzu: apt-get install im TABTAB, so werden alle verfügbaren Pakete angezeigt, die mit „im“ beginnen.

4.11 Die Datei sources.list

4.10 APT und Verwandte 4.12 apt.conf

Wenn Sie APT bereits über das Programm dselect als Installationsmethode benutzt haben, so haben Sie wahrscheinlich bereits APT an Ihre Bedürfnisse angepasst. Hier noch einige weitergehende Informationen zu APT:

APT steht für „ Advanced Package Tool“, ein Programm also, das den Systemadministrator bei der Installation und Verwaltung von Programmen unterstützen soll. Der erste Schritt zur Benutzung von APT ist die Anpassung der Konfigurationsdatei /etc/apt/sources.list . In dieser Datei befinden sich die Informationen, von welcher Quelle die Pakete geholt werden sollen. APT unterstützt eine große Zahl verschiedener Installationsquellen. Momentan sind dies: cdrom, file, http und ftp. Jede dieser Quellen wird in einer einzelnen Zeile in der Datei beschrieben. Dabei wird auch die Reihenfolge berücksichtigt; weiter oben stehende Einträge haben eine höhere Priorität.

Das Format der Einträge lässt sich wie folgt beschreiben:

deb uri distribution [component1] [component2] [...]

Ein Eintrag könnte also wie folgt aussehen:

deb ftp://ftp.debian.org/debian stable main

In der ersten Spalte findet sich der Hinweis auf die Art der Quelle. Mögliche Werte sind hier deb für Debian Pakete im Binärformat (dies ist der gebräuchliste Wert) oder aber deb-src für Pakete, die im Quellcode vorliegen. Einträge der letzteren Art benötigt man beispielsweise, um ein Paket aus den Quellpaketen neu zu übersetzen.

Das Feld uri beschreibt die Installationsquelle und den Pfad zum „root“-Verzeichnis der Debian Distribution. Auf einer CD-ROM und auf vielen Debian FTP-Servern ist der Pfad im Normalfall das Verzeichnis /debian .

Mit dem Feld distribution stellen Sie die gewünschte Version ein, die Sie installieren möchten. Normalerweise wird man sich zwischen stable für die aktuelle, stabile Version oder unstable (für die Entwicklerversion) sowie testing, (die nächste zu veröffentlichende Version) entscheiden. Wie schon am Anfang dieses Buches beschrieben, bekommt jede Debian Version einen Namen. Sie können diesen Namen auch hier einsetzen, also Woody für die Version 3.0 oder Sarge für die Version 3.1 bzw. sid für die jeweils in Entwicklung befindlichen aktuellsten Pakete.

Die Felder „component“ werden mit den einzelnen Bereichen der Distribution gefüllt. Hier können beispielsweise main, contrib, non-free, non-US (dieser Zweig ist ab Debian Sarge nicht mehr vorhanden) stehen. Zulässig sind einer oder mehrere Einträge, die durch Leerzeichen voneinander getrennt werden.

Nun noch einige genauere Informationen zu den Installationsquellen:

file

benutzt ein Verzeichnis als Quelle für die Pakete. Dies kann ein lokales oder ein per NFS gemountetes Verzeichnis sein.

cdrom

benutzt ein lokal installiertes CD-ROM-Laufwerk als Installationsquelle. Wenn die Distribution auf mehreren CDs vorliegt, wird auch dies unterstützt. Benutzen Sie das Programm apt-cdrom, um die nötigen Einträge in der Datei /etc/apt/sources.list vorzunehmen oder benutzen Sie das Programm apt-setup.

http

benutzt einen HTTP-Server als Installationsquelle. Wenn die Shell-Umgebungsvariable $http_proxy gesetzt ist (im Format: http://server:port/), so wird diese anstelle einer direkten Verbindung zum Server benutzt. Sie können auch Proxy-Server benutzen, die eine Authentifizierung verlangen. Hierzu ist der Proxy im Format: http://user:pass@server:port/ anzugeben. Bitte beachten Sie, dass die Benutzernamen und Passwörter auf diesem Wege unverschlüsselt übertragen werden.

ftp

stellt sicher die am häufigsten verwendete Methode für APT dar. Die Daten werden per FTP (File Transfer Protocol) auf den Rechner übertragen. Ein Beispiel finden Sie weiter unten.

copy

Diese Methode ist identisch mit der Methode „file“, mit dem Unterschied, dass die Pakete vor der Installation in das Verzeichnis /var/apt/cache/archives/ kopiert werden. Dies kann zum Beispiel auf Systemen Sinn machen, die keine Verbindung zum Netz haben und per ZIP-Medium aktualisiert werden sollen.

Hier nun einige Beispiele. Denken Sie daran, dass Sie durchaus mehrere dieser Einträge in der Konfigurationsdatei gleichzeitig verwenden können.

```
deb http://www.debian.org/archive stable main contrib
```

Ein solcher Eintrag benutzt das Archiv auf http://www.debian.org mit den Bereichen stable/main und stable/contrib.

Folgender Eintrag holt die Dateien via FTP aus dem Verzeichnis /debian. Es wird die noch nicht fertige („unstable“) Version von Debian GNU/Linux benutzt und auf die Bereiche main, contrib und non-free zugegriffen:

```
deb ftp://ftp.debian.org/debian unstable main contrib non-free
```

Nochmal ein ähnlicher Eintrag, diesmal für die stabile Version und lediglich den Bereich main.

```
deb ftp://ftp.debian.org/debian stable main
```

Wenn Sie die beiden vorhergehenden Zeilen in Ihrer Konfiguration einsetzen, werden beide Zeilen in einem FTP-Zugriff bearbeitet.

```
deb file:/home/fr/debian stable main contrib non-free
```

Ein solcher Eintrag benutzt eine lokale Kopie der Daten auf der Festplatte. Dies kann auch ein per NFS gemountetes Verzeichnis sein.

Zu jedem Paket existiert eine Beschreibung, jeweils als Kurzbeschreibung in einer Zeile und als ausführliche Beschreibung in mehreren Sätzen. Diese Beschreibungen sind in den Package.gz-Dateien auf den Debian Spiegeln abgelegt und zunächst ausschließlich in englischer Sprache verfügbar.

Das „Debian Description Translation Project“ (DDTP, http://ddtp.debian.org/) übersetzt diese Paketbeschreibungen in verschiedene Sprachen. Um die übersetzten Paketbeschreibungen zu nutzen, muss eine Anpassung an der Datei /etc/apt/sources.list vorgenommen werden. Dort ist der Eintrag für das verwendete Release von Debian durch einen entsprechenden Eintrag des DDTP-Servers zu ersetzen. So wird beispielsweise aus

```
deb ftp://ftp.de.debian.org/debian sarge main
```

der Eintrag:

```
deb http://ddtp.debian.org/aptable de/sarge main
```

Nach dem Ausführen des Kommandos apt-get update sollten dann die Beschreibungen zu den Paketen in deutscher Sprache erscheinen:

```
Package: bash Essential: yes Priority: required Section: base Installed-Size: 1228 Maintainer: Matthias Klose <doko@debian.org> Architecture: i386 Source: bash (2.05b-2-15) Version: 2.05b-15 Replaces: bash-doc (<= 2.05-1), bash-completion Depends: base-files (>= 2.1.12) Pre-Depends: libc6 (>= 2.3.2.ds1-4), libncurses5 (>= 5.4-1) Suggests: bash-doc Conflicts: bash-completion Filename: pool/main/b/bash/bash_2.05b-15_i386.deb Size: 622470 MD5sum: b804b1efff12046675a8826b41de7a37 Description: Die GNU Bourne Again SHell  Bash ist ein sh-kompatibler Befehlssprachen-Interpreter, der Befehle  ausführt, die von der Standardeingabe oder aus einer Datei kommen.  Bash beinhaltet auch nützliche Fähigkeiten der Korn- und C-Shells  (ksh und csh).  .  Letztendlich soll Bash eine kompatible Implementation der IEEE POSIX  Shell and Tools Spezifikation (IEEE Working Group 10003.2) sein.  .  Im bash-Paket ist der Programmable Completion Code von Ian  Macdonald enthalten.
```

Manchmal kann es notwendig oder sinnvoll sein, ältere Debian Releases einzusetzen, beispielsweise um Debian auf älterer Hardware zu installieren. Eine Übersicht der verfügbaren Debian Releases ist unter http://www.debian.org/distrib/archive zu finden.

Die jeweils aktuell in der Entwicklung befindliche Debian Version („testing“ oder „unstable“) unterliegt einem häufigen Wechsel der Paketversionen. Manchmal ist ein Paket nur für einen oder wenige Tage verfügbar. Da nun neuere Versionen nicht immer eine Verbesserung mit sich bringen, wollen Sie vielleicht auf ein älteres Paket ausweichen.

Leider sind auf den Debian Servern ältere Pakete nicht mehr verfügbar. Unter http://snapshot.debian.net sind alle Pakete, auch in älteren Versionen, weiterhin zugänglich.

Der Zugriff auf dieses Archiv erfolgt mit einigen besonderen Einträgen in der Datei /etc/apt/sources.list. Hierbei kann zwischen dem Zugriff über ein absolutes oder ein relatives Datum gewählt werden.

```
deb http://snapshot.debian.net/archive/YYYY/MM/DD/debian unstable main contrib non-free deb http://snapshot.debian.net/archive/YYYY/MM/DD/debian-non-US unstable/non-US main contrib non-free deb-src http://snapshot.debian.net/archive/YYYY/MM/DD/debian unstable main contrib non-free deb-src http://snapshot.debian.net/archive/YYYY/MM/DD/debian-non-US unstable/non-US main contrib non-free
```

Im vorhergehenden Beispiel kann gezielt auf den tagesaktuellen Stand zugegriffen werden. Die Buchstabenkombination YYYY/MM/DD ist dabei durch das gewünschte Datum zu ersetzen.

```
deb http://snapshot.debian.net/archive/date/datestr/debian unstable main contrib non-free deb http://snapshot.debian.net/archive/date/datestr/debian-non-US unstable/non-US main contrib non-free deb-src http://snapshot.debian.net/archive/date/datestr/debian unstable main contrib non-free deb-src http://snapshot.debian.net/archive/date/datestr/debian-non-US unstable/non-US main contrib non-free
```

In diesem Beispiel muss datestr durch einen String ersetzt werden, der dem Kommando date als Datum bekannt ist. Dies kann beispielsweise yesterday, 2-days-ago oder last-week sein.

4.11 Die Datei sources.list

4.10 APT und Verwandte 4.12 apt.conf

4.12 apt.conf

4.11 Die Datei sources.list 4.13 apt-setup

Die Konfigurationsdatei /etc/apt/apt.conf stellt die zentrale Stelle für alle Einstellungen von Programmen dar, die auf APT aufsetzen. Von allen Programmen wird zum einen die gemeinsame Konfigurationsdatei, zum anderen auch ein Kommandozeilen-Interpreter (apt-config) genutzt, um die Konfigurationsdatei auszuwerten. Dies garantiert für alle Programme eine einheitliche Arbeitsumgebung. Beim Start eines auf APT basierenden Programmes wird zunächst die Umgebungsvariable APT_CONFIG ausgewertet. Über diese Umgebungsvariable lässt sich eine alternative Konfigurationsdatei angeben.

Die Konfigurationsdatei ist in einer Baumstruktur organisiert und trennt verschiedene Optionen in funktionale Gruppen. Werte für Optionen werden durch zwei Doppelpunkte eingeleitet, beispielsweise so: APT::Get::Assume-Yes. Optionen übernehmen keine Werte von übergeordneten Gruppen.

Die Syntax der Konfigurationsdatei ist an die von Programmen wie bind und dhcp angelehnt. Zeilen, die mit den Zeichen // beginnen, werden als Kommentare behandelt und ignoriert. Alle weiteren Zeilen entsprechen dem Aufbau von APT::Get::Assume-Yes "true";. Wichtig ist hierbei das abschließende Semikolon am Ende jeder Zeile, die Werte selbst können, müssen aber nicht von Anführungszeichen umschlossen werden. Ein neuer Bereich wird durch geschweifte Klammern umschlossen.

```
APT {Get {  Assume-Yes "true";Fix-Broken "true"; };};
```

Leerzeilen können innerhalb eines Bereiches genutzt werden, um die Datei besser lesbar zu halten. Einfache Einträge können erzeugt werden, indem innerhalb eines Bereiches ein Kommando abgelegt wird und der Eintrag mit einem Semikolon beendet wird. Ein Beispiel:

```
DPkg::Pre-Install-Pkgs {"/usr/sbin/dpkg-preconfigure --apt";};
```

Hier ein einfaches Beispiel für eine APT-Konfigurationsdatei (/etc/apt/apt.conf):

```
// $Id: apt.conf,v 1.43 1999/12/06 02:19:38 jgg Exp $ /* This file is a
sample configuration file with a few harmless sample     options.    */
APT  {   // Options for apt-get   Get    {      Download-Only "false";   };
};  // Options for the downloading routines Acquire {   Retries "0"; };  //
Things that effect the APT dselect method DSelect  {   Clean "auto";   //
always|auto|prompt|never };  DPkg  {   // Probably don't want to use
force-downgrade..   Options {"--force-overwrite";} }
```

Innerhalb der Konfigurationsdatei sind zwei spezielle Einträge erlaubt, #include und #clear. #include liest die angegebene Datei an der entsprechenden Stelle ein, solange der Dateiname nicht mit einem Slash (/) endet. #clear löscht eine Liste von Namen.

Allen APT-Programmen kann die Option -o übergeben werden, um eine Konfigurationsoption auf der Kommandozeile zu übergeben. Die Syntax hierfür ist der volle Name der Option, beispielsweise APT::Get::Assume-Yes, gefolgt von einem Gleichheitszeichen und dem gewünschten Wert für diese Option. Listen können Werte hinzugefügt werden, indem abschließend zwei Doppelpunkte (::) angegeben werden.

Im Folgenden werden die verschiedenen Gruppen in der Konfigurationsdatei beschrieben.

APT

Diese Gruppe enthält generelle Parameter zu allen APT-Programmen.

Architecture

Die System Architektur. Über diesen Wert wird ermittelt, welche Paketlisten und Debian Pakete passend zur Hardware-Architektur geholt werden. Voreingestellt ist immer die Architektur für die APT auf dem System übersetzt wurden.

Ignore-Hold

Setzt den „Hold“-Status global für alle Pakete außer Kraft.

Clean-Installed

Voreingestellt ist hier der Wert „on“.

Immediate-Configure

Mit dieser Option lässt sich das unmittelbare Konfigurieren von Paketen abschalten. APT ruft nach einer Anzahl von Paketen, die zur Installation vorgemerkt sind, die Konfiguration dieser Pakete auf. Dies ist sinnvoll, da Pakete aufeinander basieren und es erforderlich ist, dass ein bestimmtes Paket vor einem anderen bereits installiert und konfiguriert ist. Das Abschalten dieser Option kann die Installation von vielen Paketen auf einem langsamen System beschleunigen, führt aber unter Umständen zu Problemen bei der Installation.

Force-LoopBreak

Diese Option erlaubt APT, ein essenzielles Paket kurzfristig zu entfernen, um eine Endlosschleife in einem Konflikt oder einer Abhängigkeit (Pre-Depend) auflösen zu können. Solch eine Endlosschleife darf eigentlich nicht existieren und ist ein schwerwiegender Fehler. Diese Option wirkt nicht bei den essenziellen Paketen tar, gzip, libc, dpkg, bash oder allen anderen Paketen, von denen diese Pakete abhängen. Diese Option darf nur eingesetzt werden, wenn Sie genau verstanden haben, was dies für Konsequenzen haben kann.

Cache-Limit

APT nutzt einen festgelegten Speicherbereich, um Informationen über die verfügbaren Pakete zu speichern. Mit dieser Option kann der Wert verändert werden.

Get

Dieser Abschnitt kontrolliert das Verhalten von apt-get.

Cache

Dieser Abschnitt kontrolliert das Verhalten von apt-cache.

CDROM

Dieser Abschnitt kontrolliert das Verhalten von apt-cdrom.

Acquire

In dieser Gruppe sind Optionen für das Herunterladen von Paketen und die Behandlung von URIs enthalten.

Queue-Mode

Dieser Eintrag betrifft die Behandlung von ausgehenden Verbindungen bei der Übertragung von Paketen. Der Wert „host“ bedeutet, dass zu jedem System, von dem Pakete geholt werden, eine Verbindung aufgebaut wird. Der Wert „access“ bewirkt, dass je URI-Typ eine Verbindung aufgebaut wird.

Retries

Anzahl der Versuche bei einem fehlgeschlagenen Download.

Source-Symlinks

Nutzt symbolische Links für Quellcode-Archive, anstatt eine Kopie der Datei anzulegen. Voreinstellung ist „True“.

http

HTTP-Adressen können mit diesem Eintrag über einen Proxy-Server erreicht werden. Die Angabe http::Proxy stellt der Standard-Proxy für alle Anfragen dar. Die Adresse des Proxy-Servers ist wie üblich als http://[[user][:pass]@]host[:port]/ anzugeben. Alternativ kann für einzelne Server ein gesonderter Proxy in der Form http::Proxy::<host> gesetzt werden. Soll für einen einzelnen Server kein Proxy verwendet werden, so ist das Schlüsselwort DIRECT zu benutzen. Die Umgebungsvariable http_proxy überschreibt alle diese Angaben.

Es sind drei Werte im Zusammenhang mit diesem Eintrag erlaubt. No-Cache teilt dem Proxy mit, dass in keinem Fall Daten aus dem Cache benutzt werden sollen. Max-Age wird für Index-Dateien verwendet und gibt die Zeit in Sekunden an, die das maximale Alter der Index-Dateien bezeichnet. Die Index-Dateien auf Debian Servern werden täglich aktualisiert, der vorgegebene Wert entspricht ebenso einem Tag. No-Store führt dazu, dass die Anfrage niemals im Cache gespeichert wird.

Mit der Option timeout lässt sich die Zeit für Verbindungs- und Daten-Timeouts einstellen.

ftp

FTP-Adressen können mit diesem Eintrag über einen Proxy-Server erreicht werden. Die Angabe ftp::Proxy stellt der Standard-Proxy für alle Anfragen dar. Die Adresse des Proxy-Servers ist wie üblich als ftp://[[user][:pass]@]host[:port]/ anzugeben. Um einen FTP-Proxy zu nutzen, muss in der Konfigurationsdatei der Wert ftp::ProxyLogin mit einem Skript belegt sein. Dieser Eintrag beschreibt die Kommandos, die an den FTP-Server beim Login-Vorgang gesendet werden.

```
ProxyLogin {     "USER $(PROXY_USER)";     "PASS
$(PROXY_PASS)";     "USER
$(SITE_USER)@$(SITE):$(SITE_PORT)";     "PASS
$(SITE_PASS)"; };
```

Die Umgebungsvariable ftp_proxy überschreibt alle diese Angaben.

Mit der Option timeout lässt sich die Zeit für Verbindungs- und Daten-Timeouts einstellen.

cdrom

Der Wert für einen Eintrag, der eine CD (ein CD-ROM- oder DVD-Laufwerk) betrifft, bezieht sich auf das Verzeichnis im Dateisystem, an dem der Inhalt eingebunden wird. Die Syntax für solch einen Eintrag lautet:

```
"/cdrom/"::Mount "foo";
```

Wichtig ist bei einem solchen Eintrag der vorangestellte Slash (/). Das Kommando umount kann ebenso durch den Eintrag UMount mit einer solchen Syntax genutzt werden.

Verzeichnisse

Dir::State

Dieser Abschnitt betrifft Verzeichnisse mit lokalen Status-Informationen. lists bezeichnet das Verzeichnis, in dem die heruntergeladenen Paketlisten abgelegt werden. status ist der Dateiname der dpkg-Status-Datei. preferences ist der Name der APT-Konfigurationsdatei. Dir::State bezeichnet das Verzeichnis, welches allen anderen Objekten in diesem Abschnitt vorangestellt wird, wenn diese nicht mit / oder ./ beginnen.

Dir::Cache

In diesem Abschnitt werden Verzeichisse festgelegt, die mit heruntergeladenen Dateien zu tun haben. Die Angaben pkgcache und srcpkgcache enthalten das Verzeichnis für Binär- und Quellcode-Pakete. Die Verwendung des Cache kann deaktiviert werden, in dem die Verzeichnisnamen nicht angegeben werden. Das voreingestellte Verzeichnis für alle Caches wird in Dir::Cache festgelegt.

Dir::Etc

Enthält den Pfad zu den Konfigurationsdateien, sourcelist bezeichnet dabei die Datei sources.list, main ist die voreingestellte Konfigurationsdatei.

Dir::Parts

Dieser Abschnitt liest alle Konfigurationsteile in alphabetischer Reihenfolge aus dem angegebenen Verzeichnis. Danach wird die zentrale Konfigurationsdatei eingelesen.

Dir::Bin

In diesem Verzeichnis werden binäre Programme gesucht. Dir::Bin::Methods enthält den Pfad zu den zusätzlichen Methoden. Die Angaben zu gzip, dpkg, apt-get, dpkg-source, dpkg-buildpackage und apt-cache zeigen auf die jeweiligen Programme im Verzeichnisbaum.

APT innerhalb von dselect

Wird APT als Zugriffsmethode von dselect verwendet, so können die folgenden Parameter zur Konfiguration verwendet werden.

Clean

Modus, um den Paketcache zu verwalten.

options

Der Inhalt dieser Variablen wird an apt-get als Kommandozeilenoption weitergereicht, wenn eine Installation durchgeführt wird.

Updateoptions

Der Inhalt dieser Variablen wird an apt-get als Kommandozeilenoption weitergereicht, wenn ein Update durchgeführt wird.

PromptAfterUpdate

Wird die Funktion Update im Programm dselect verwendet, so wird immer nachgefragt, um eine Aktion fortzusetzen, wenn diese Variable auf True gesetzt ist.

Aufruf von DPKG über APT

Im Abschnitt DPkg der Konfigurationsdatei können verschiedene Parameter eingestellt werden, die sich mit dem Aufruf von dpkg aus APT-basierenden Programmen beschäftigen.

options

Eine Liste von Optionen, die dem Aufruf von dpkg übergeben werden.

Pre-Invoke, Post-Invoke

Eine Liste von Shell-Kommandos, die vor bzw. nach dem Aufruf von dpkg ausgeführt werden. Die Kommandos werden mittels /bin/sh ausgeführt, sollte eines der Kommandos mit einem Fehler beendet werden, so bricht auf APT an dieser Stelle die weitere Ausführung ab.

Pre-Install-Pkgs

Dies ist eine Liste von Kommandos, welche vor dem Aufruf von dpkg ausgeführt werden. Die Kommandos werden mittels /bin/sh ausgeführt, sollte eines der Kommandos mit einem Fehler beendet werden, so bricht auf APT an dieser Stelle die weitere Ausführung ab. APT übergibt die Namen aller zu installierenden Debian Paketdateien (.deb) an die Kommandos.

Run-Directory

APT wechselt in das hier angegebene Verzeichnis vor dem Aufruf von dpkg. Voreingestellt ist hier das Verzeichnis /.

Build-options

Diese Optionen werden dem Programm dpkg-buildpackage übergeben. Voreingestellt ist, dass Pakete nicht digital signiert werden und alle Binär-Pakete erzeugt werden.

Abschließend ein Beispiel für die APT-Konfigurationsdatei mit allen Optionen. Dieses Beispiel ist so in dieser Form nicht sinnvoll einzusetzen, zeigt aber sehr gut auf, wie verschiedene Optionen mit Werten belegt werden.

```
// $Id: configure-index,v 1.10 2004/07/17 19:37:16 mdz Exp $ /* This file is an index of all APT configuration directives. It should NOT actually be used as a real config file, though it is (except for the last line) a completely valid file. Most of the options have sane default values, unless you have specific needs you should NOT include arbitary items in a custom configuration. In some instances involving filenames it is possible to set the default directory when the path is evaluated. This means you can use relative paths within the sub scope. The configuration directives are specified in a tree with {} designating a subscope relative to the tag before the {}. You can further specify a
```

```
subscope using scope notation eg, APT::Architecture "i386"; This is prefixed with the current scope. Scope notation must be used if an option is specified on the command line with -o. */ quiet "0"; // Options for APT in general APT { Architecture "i386"; Build-Essential "build-essential"; // Options for apt-get Get { Arch-Only "false"; Download-Only "false"; Simulate "false"; Assume-Yes "false"; Force-Yes "false"; // I would never set this. Fix-Broken "false"; Fix-Missing "false"; Show-Upgraded "false"; Show-Versions "false"; Upgrade "true"; Print-URIs "false"; Compile "false"; Download "true"; Purge "false"; List-Cleanup "true"; ReInstall "false"; Trivial-Only "false"; Remove "true"; Only-Source ""; Diff-Only "false"; Tar-Only "false"; }; Cache { Important "false"; AllVersions "false"; GivenOnly "false"; RecurseDepends "false"; ShowFull "false"; Generate "true"; NamesOnly "false"; AllNames "false"; Installed "false"; }; CDROM { Rename "false"; NoMount "false"; Fast "false"; NoAct "false"; }; // Some general options Ignore-Hold "false"; Clean-Installed "true"; Immediate-Configure "true"; // DO NOT turn this off, see the man page Force-LoopBreak "false"; // DO NOT turn this on, see the man page Cache-Limit "4194304"; Default-Release ""; }; // Options for the downloading routines Acquire { Queue-Mode "host"; // host|access Retries "0"; Source-Symlinks "true"; // HTTP method configuration http { Proxy "http://127.0.0.1:3128"; Proxy::http.us.debian.org "DIRECT"; // Specific per-host setting Timeout "120"; Pipeline-Depth "5"; // Cache Control. Note these do not work with Squid 2.0.2 No-Cache "false"; Max-Age "86400"; // 1 Day age on index files No-Store "false"; // Prevent the cache from storing archives }; ftp { Proxy "ftp://127.0.0.1/"; Proxy::http.us.debian.org "DIRECT"; // Specific per-host setting /* Required script to perform proxy login. This example should work for tisfwtk */ ProxyLogin { "USER $(PROXY_USER)"; "PASS $(PROXY_PASS)"; "USER $(SITE_USER)@$(SITE):$(SITE_PORT)"; "PASS $(SITE_PASS)"; }; Timeout "120"; /* Passive mode control, proxy, non-proxy and per-host. Pasv mode is prefered if possible */ Passive "true"; Proxy::Passive "true";
```

```
Passive::http.us.debian.org "true"; // Specific per-host setting   };
cdrom   {     mount "/cdrom";      // You need the trailing slash!
"/cdrom"     {        Mount "sleep 1000";        UMount "sleep 500";     }
}; };  // Directory layout Dir "/" {   // Location of the state dir   State
"var/lib/apt/"     {      Lists "lists/";      xstatus "xstatus";      userstatus
"status.user";      status "/var/lib/dpkg/status";      cdroms "cdroms.list";
};      // Location of the cache dir   Cache "var/cache/apt/" {      Archives
"archives/";      srcpkgcache "srcpkgcache.bin";      pkgcache
"pkgcache.bin";        };      // Config files   Etc "etc/apt/" {      SourceList
"sources.list";      Main "apt.conf";      Preferences "preferences";
Parts "apt.conf.d/";   };      // Locations of binaries   Bin {      methods
"/usr/lib/apt/methods/";      gzip "/bin/gzip";      dpkg "/usr/bin/dpkg";
dpkg-source "/usr/bin/dpkg-source";      dpkg-buildpackage
"/usr/bin/dpkg-buildpackage"      apt-get "/usr/bin/apt-get";      apt-cache
"/usr/bin/apt-cache";   }; };  // Things that effect the APT dselect
method DSelect   {    Clean "auto";   // always|auto|prompt|never
Options "-f";    UpdateOptions "";    PromptAfterUpdate "no";
CheckDir "no"; }  DPkg   {    // Probably don't want to use force-
downgrade..    Options {"--force-overwrite";"--force-downgrade";}
// Auto re-mounting of a readonly /usr    Pre-Invoke {"mount -o
remount,rw /usr";};    Post-Invoke {"mount -o remount,ro /usr";};        //
Prevents daemons from getting cwd as something mountable (default)
Run-Directory "/";        // Build options for apt-get source --compile
Build-Options "-b -uc";     // Pre-configure all packages before they are
installed using debconf.    Pre-Install-Pkgs {"dpkg-preconfigure --apt --
priority=low --frontend=dialog";};     // Flush the contents of stdin
before forking dpkg.    FlushSTDIN "true";     // Control the size of the
command line passed to dpkg.    MaxBytes 1024;    MaxArgs 350; }  /*
Options you can set to see some debugging text They correspond to
names    of classes in the source code */ Debug   {
pkgProblemResolver "false";   pkgAcquire "false";
pkgAcquire::Worker "false";   pkgDPkgPM "false";   pkgOrderList
"false";      pkgInitialize "false";   // This one will dump the
configuration space   NoLocking "false";   Acquire::Ftp "false";    //
Show ftp command traffic   Acquire::Http "false";   // Show http
command traffic   aptcdrom "false";        // Show found package files
IdentCdrom "false"; }  /* Whatever you do, do not use this
```

configuration file!! Take out ONLY the portions you need! */ This Is Not A Valid Config File

4.12 apt.conf

4.11 Die Datei sources.list 4.13 apt-setup

4.13 apt-setup

4.12 apt.conf 4.14 apt-cdrom

Mit apt-setup können über eine Benutzeroberfläche die Einträge in der Datei /etc/apt/sources.list ergänzt werden. apt-setup kann hierzu die Methoden „http“„ftp“ und „filesystem“ benutzen. Für die Methode „cdrom“ wird auf das Programm apt-cdrom zurückgegriffen. Ein weiterer Menüpunkt („edit sources list by hand“) ruft einen Editor (vi) auf; Sie können dann weitere Einträge von Hand aufnehmen. Nach Verlassen des Editors wird versucht, von den angegebenen Quellen die Packages-Dateien zu lesen.

Als einzige Option kann beim Starten der Wert „probe“ übergeben werden. Dies führt dazu, dass eine eingelegte CD-ROM sofort eingelesen wird.

4.13 apt-setup

4.12 apt.conf 4.14 apt-cdrom

4.14 apt-cdrom

4.13 apt-setup 4.15 apt-get

Dieses Kommando wird von apt-setup zur Integration von neuen CD-ROMs verwendet, kann aber auch eigenständig auf der Kommandozeile eingesetzt werden.

Im einfachsten Fall führt das Kommando apt-cdrom add dazu, dass die eingelegte CD durchsucht und mit den entsprechenden Werten in die Datei sources.list aufgenommen wird. Hierzu ist ein entsprechender Eintrag für das CD-ROM-Laufwerk in der Datei /etc/fstab notwendig. Ein solcher Eintrag wird bereits bei der Installation angelegt und muss normalerweise nur bei Veränderungen an der Hardware angepasst werden.

Wird apt-cdrom ohne weitere Angaben aufgerufen, so werden Informationen über weitere Optionen ausgegeben.

```
apt 0.5.9 for linux i386 compiled on Aug 10 2003 19:58:08 Usage: apt-cdrom [options] command  apt-cdrom is a tool to add CDROM's to APT's source list. The CDROM mount point and device information is taken from apt.conf and /etc/fstab.  Commands:    add - Add a CDROM ident - Report the identity of a CDROM  Options:   -h   This help text   -d   CD-ROM mount point   -r   Rename a recognized CD-ROM   -m   No mounting   -f   Fast mode, don't check package files   -a   Thorough scan mode   -c=? Read this configuration file   -o=? Set an arbitary configuration option, eg -o dir::cache=/tmp See fstab(5)
```

4.14 apt-cdrom

4.13 apt-setup 4.15 apt-get

4.15 apt-get

4.14 apt-cdrom 4.16 apt - Offline nutzen

apt-get ist die zeitgemäße Benutzerschnittstelle zur Verwaltung von Paketen auf einem Debian System. Während es mit dpkg möglich ist, jedes beliebige Debian Paket zu installieren (wenn es zuvor auf die lokale Festplatte kopiert wurde), so ist apt-get zwingend auf eine Paketliste und damit auch eine sinnvolle sources.list angewiesen. Somit ist immer gewährleistet, dass das System in einem stabilen und fest definierten Zustand verbleibt. Mit apt-get wird dem Administrator ein ebenso elegantes wie auch mächtiges Werkzeug gegeben. Dies haben auch viele andere Distributionshersteller erkannt und haben apt-get in die eigenen Distributionen aufgenommen. apt-get ist somit auch auf RPM-basierten Systemen, wie beispielsweise Novell/SuSE oder RedHat/Fedora zu Hause. Die Entwicklung wird dabei über die Webseite koordiniert. Eine Übersicht von Repositories mit RPM-Paketen, die via APT installiert werden können, finden Sie unter http://freshrpms.net/apt/repositories.html.

Dieses Programm ohne grafische Benutzeroberfläche ist recht einfach zu benutzen. Grafische Frontends zu apt sind mit aptitude und gpk-application verfügbar.

Bevor apt-get die gewünschten Aktionen wie zum Beispiel das Löschen oder Installieren einzelner Pakete ausführt, die das System verändern, werden Sie über den zukünftigen Zustand des Systems informiert. Es werden folgende Informationen angezeigt: die Anzahl der Pakete, die aktualisiert werden (neuere Version), die Anzahl der Pakete, die nicht verändert werden (kept back), die Anzahl der zu löschenden Pakete sowie die Anzahl der neu zu installierenden Pakete.

Zusätzlich werden, wenn Sie die Option install benutzen, die Pakete ausgewählt und angezeigt, die aufgrund der Abhängigkeiten der zu installierenden Pakete benötigt werden. Hier ein Beispiel:

The following extra packages will be installed: libdbd-mysql-perl xlib6 zlib1 xzx libreadline2 libdbd-msql-perl mailpgp xdpkg fileutils pinepgp zlib1g xlib6g perl-base bin86 libgdbm1 libgdbmg1 quake-lib gmp2 bcc xbuffy squake pgp-i python-base debmake ldso perl libreadlineg2 ssh

Sollte es notwendig sein, Pakete zu löschen, oder wurden Pakete zum Entfernen ausgewählt, so werden auch diese gesondert angezeigt.

The following packages will be REMOVED: xlib6-dev xpat2 tk40-dev xkeycaps xbattle xonix xdaliclock tk40 tk41 xforms0.86 ghostview xloadimage xcolorsel xadmin xboard perl-debug tkined xtetris libreadline2-dev perl-suid nas xpilot xfig

Diese Liste sollte aufmerksam geprüft werden, um sicherzugehen, dass nicht versehentlich Pakete entfernt werden.

Wenn Pakete neu installiert werden, so wird auch darüber informiert:

The following NEW packages will be installed: zlib1g xlib6g perl-base libgdbmg1 quake-lib gmp2 pgp-i python-base

Dies dient nur zur Information; diese Pakete sind bisher nicht auf Ihrem System installiert.

Bei der Auswahl der Pakete kann es passieren, dass aufgrund von nicht erfüllten Abhängigkeiten einzelne Pakete in der aktuellen Version behalten werden. Da nicht alle benötigten Komponenten installiert werden können, werden diese Pakete nicht aktualisiert (kept back).

The following packages have been kept back: compface man-db tetex-base msql libpaper svgalib1 gs snmp arena lynx xpat2 groff xscreensaver

Immer wenn ein System mit der Option upgrade aktualisiert wird, kann es passieren, dass einzelne Pakete nicht aktualisiert werden können, weil diese auf Paketen basieren, die noch nicht in der benötigten Version verfügbar sind, oder aber es bestehen Konflikte mit bereits installierten Paketen. In diesem Fall werden die Pakete im aktuellen Zustand gehalten. Sie können apt-get install benutzen, um diese Pakete trotzdem zu aktualisieren.

```
The following held packages will be changed:   cvs
```

Wenn Sie ein auf „hold“ gesetztes Paket aktualisieren, bekommen Sie die Meldung, dass dieses Paket verändert (changed) wird. Außer bei apt-get install kann diese Meldung auch bei apt-get dist-upgrade auftreten.

Abschließend findet sich noch eine Zusammenfassung über die Anzahl der betroffenen Pakete. Weiterhin wird angezeigt, welches Datenvolumen übertragen werden muss und wie viel Festplattenplatz zusätzlich benötigt wird (oder auch frei wird).

```
206 packages upgraded, 8 newly installed, 23 to remove and 51 not
upgraded. 12 packages not fully installed or removed. Need to get
65.7M/66.7M of archives. After unpacking 26.5M will be used.
```

Die erste Zeile ist eine knappe Zusammenfassung in Zahlen zu den bereits vorher ausgegebenen Informationen. In der zweiten Zeile finden Sie die nicht installierten Pakete, die bereits entpackt, aber nicht konfiguriert wurden („N packages not fully installed or removed“). In der dritten Zeile finden Sie die Informationen über den benötigten Platz und die zu übertragenden Daten. Hierbei steht die erste Zahl für die tatsächlich noch zu übertragenden Daten und die zweite für den Gesamtwert. Wenn bereits Pakete bei einer früheren Installation (möglicherweise auch nur teilweise) übertragen worden sind, aber nicht installiert werden konnten, so werden diese gespeichert und nicht noch einmal übertragen.

Wenn Sie nun mit der Installation beginnen (dies müssen Sie bei apt-get install mit einem y anstoßen), werden die Pakete von dem ausgewählten Medium gelesen und installiert.

Während des Herunterladens von Archiven und Paketdateien zeigt apt-get eine Reihe von Informationen an:

```
sushi:/root# apt-get update Get:1 ftp://linux frozen/main Packages [833kB] Get:2 ftp://linux frozen/main Release [93B] Get:3 ftp://linux frozen/contrib Packages [33.4kB] Get:4 ftp://linux frozen/contrib Release [96B] Get:5 ftp://linux frozen/non-free Packages [78.1kB] Get:6 ftp://linux frozen/non-free Release [97B] Get:7 ftp://linux stable/non-US Packages [8880B] Get:8 ftp://linux stable/non-US Release [95B] Fetched 953kB in 12s (74.1kB/s) Reading Package Lists... Done Building Dependency Tree... Done
```

Unmittelbar während des Downloads wird auch der Fortschritt angezeigt, inklusive der voraussichtlich noch benötigten Zeit für die Übertragung:

```
11% [5 frozen/non-free `Waiting for file' 0/32.1k 0%] 2203b/s 1m52s
```

In den mit Get: beginnenden Zeilen stehen die benutzte Methode (hier ftp) und der benutzte Server. Den Servernamen können Sie in einem lokalen Netz auch verkürzt angeben, wie hier mit linux gezeigt. Weiterhin werden der Bereich (frozen und stable) sowie die Verzeichnisse (main, contrib usw.) und die Dateinamen mit der Dateigröße angezeigt.

Bei der Installation von Paketen sieht der Vorgang ähnlich aus:

```
sushi:/root# apt-get install aptitude Reading Package Lists... Done Building Dependency Tree... Done The following NEW packages will be installed: aptitude 0 packages upgraded, 1 newly installed, 0 to remove and 3 not upgraded. Need to get 148kB of archives. After unpacking 471kB will be used. Get:1 ftp://ftp.debian.de potato/main aptitude 0.0.4a-3 [148kB] Fetched 148kB in 5s (25.3kB/s) Selecting previously deselected package aptitude. (Reading database ... 22625 files and directories currently installed.) Unpacking aptitude (from .../aptitude_0.0.4a-3_i386.deb) ... Setting up aptitude (0.0.4a-3) ...
```

Hier wird in der Zeile „Get:“ der Paketname der zu installierenden Software angezeigt. Danach wird das Paket entpackt und installiert.

Installation von Paketen überwachen

Im laufenden Betrieb werden von den verschiedenen Diensten auf einem System Logdateien (im Verzeichnis /var/log/) erzeugt. Wünschenswert wäre eine solche Funktionalität auch für die Installation von Software-Paketen, dpkg stellt ab der Version 1.13.5 die Option --log zur Verfügung.

Die Option --log protokolliert sowohl den Aufruf von dpkg (z.B. 2005-12-30 18:10:33 install hello 1.3.18 2.1.1-4) als auch die Ergebnisse (z.B. 2005-12-30 18:10:35 status installed hello 2.1.1-4)

Wenn jeder Aufruf von dpkg protokolliert werden soll (auch wenn dieser über Oberflächen wie aptitude geschehen ist), könnte der Eintrag log /var/log/dpkg.log der Datei /etc/dpkg/dpkg.cfg hinzugefügt werden. Es ist dafür zu sorgen, dass die erstellte Protokolldatei regelmäßig rotiert wird, da die Größe der Datei bei jeder Installation (oder auch beim Entfernen) eines Pakets zunimmt. Wenn das Programm logrotate benutzt wird, kann dies erreicht werden, indem eine Datei /etc/logrotate.d/dpkg mit folgendem Inhalt erstellt wird:

```
/var/log/dpkg {        missingok        notifempty        }
```

apt-get benutzt folgende Syntax:

apt-get [options] [command] [package ...]

Die Optionen werden Sie sicher seltener benötigen, daher zuerst einige Worte zu den einzelnen Kommandos:

check

Bei jedem Start von apt (es sei denn, Sie benutzen die Option update) wird eine Reihe von Prüfungen durchgeführt, um sicherzustellen, dass apt funktionsfähig und Ihr System

in einem guten Zustand ist. Sie können diese Prüfungen auch zu jeder Zeit selbst anstoßen:

```
sushi:/root# apt-get check Reading Package Lists... Done Building Dependency Tree... Done
```

Als Erstes werden die Paketdateien anhand der Informationen in /etc/apt/sources.list eingelesen. Wenn Sie diesen Vorgang noch einmal wiederholen, werden Sie feststellen, dass der zweite Test deutlich schneller beendet ist, die Informationen werden von apt zwischengespeichert. Für Paketdateien, die nicht gefunden werden, wird ggf. eine Warnung ausgegeben; diese Pakete werden dann ignoriert.

Im zweiten Schritt wird eine detaillierte Analyse des Systems durchgeführt, inklusive aller Abhängigkeiten. Es werden von jedem installierten oder bereits entpackten, aber noch nicht konfigurierten Paket die Abhängigkeiten geprüft. Wenn dabei ein Problem auftaucht, wird dies angezeigt, und apt-get bricht die weitere Bearbeitung ab.

```
sushi:/root# apt-get check Reading Package Lists... Done Building Dependancy Tree... Done You might want to run apt-get -f install' to correct these. Sorry, but the following packages have unmet dependencies:   9fonts: Depends: xlib6g but it is not installed   uucp: Depends: mailx but it is not installed   blast: Depends: xlib6g (>= 3.3-5) but it is not installed   adduser: Depends: perl-base but it is not installed aumix: Depends: libgpmg1 but it is not installed   debiandoc-sgml: Depends: sgml-base but it is not installed   bash-builtins: Depends: bash (>= 2.01) but 2.0-3 is installed   cthugha: Depends: svgalibg1 but it is not installed           Depends: xlib6g (>= 3.3-5) but it is not installed libreadlineg2: Conflicts:libreadline2 (< 2.1-2.1)
```

In diesem Beispiel gibt es eine ganze Reihe Probleme: Einige Pakete sind gar nicht installiert, bei anderen Paketen sind falsche Versionen installiert. Für jedes einzelne Paket, bei dem ein Problem festgestellt wurde, wird eine Zeile ausgegeben mit den Informationen, um welches Paket es sich handelt und welches Problem festgestellt wurde.

Es gibt zwei Möglichkeiten, wie es zu solchen Problemen kommen kann: Bei einem „Upgrade“ kann es passieren, dass ein benötigtes Paket fehlt, oder es ist während der Installation eines Pakets ein Problem aufgetreten. Wenn Letzteres der Fall ist, dann wurde das Paket zwar entpackt, aber nicht korrekt konfiguriert.

Beide Situationen können von apt in den meisten Fällen mit der Option -f selbst behoben werden. Wenn Sie dselect mit der Methode apt benutzen, wird immer die Option -f verwendet, um eine einfache Verwendung zu gewährleisten.

Trotzdem kann es passieren, dass bei einem schweren Problem apt nicht in der Lage ist, das Problem zu beheben. In diesem Fall müssen Sie von Hand mit dem Programm dpkg den Konflikt beseitigen. Sie können danach mit apt-get fortfahren.

update

aktualisiert die Übersicht der verfügbaren Pakete, liest also die Informationen aus den Dateien Packages.gz der jeweiligen Distribution ein. Sie sollten update immer ausführen, wenn Sie wissen, dass sich die Inhalte der Paketdateien beziehungsweise die verfügbaren Pakete auf dem Server geändert haben. Auf jeden Fall sollten Sie update vor dem Aufruf von upgrade oder dist-upgrade aufrufen.

Eine neue Fähigkeit von apt seit Debian 4.0 ist die Möglichkeit, lediglich die Änderungen der Packages-Dateien seit ihrer letzten Aktualisierung herunterzuladen.

upgrade

Um alle auf dem System installierten Pakete auf die aktuelle Version zu bringen, können Sie upgrade benutzen. Alle bereits installierten Pakete, von denen eine neuere Version verfügbar ist, werden aktualisiert. Es werden keine Pakete gelöscht, die bereits installiert sind (und in den neuen Paketdateien nicht mehr enthalten sind), oder Pakete installiert, die noch nicht auf dem System vorhanden sind. Pakete, die bereits installiert sind und es bei einem Upgrade erforderlich machen, den Status anderer Pakete zu verändern, werden nicht aktualisiert. apt-get update muss vorab ausgeführt werden, so dass sichergestellt ist, dass die Paketinformationen auf dem neuesten Stand sind.

dist-upgrade

ist eigentlich eine Ergänzung oder Erweiterung zu upgrade. Es wird hierbei dafür Sorge getragen, dass für das System wichtigere Pakete zuerst installiert werden. Hierbei werden in engen Grenzen auch Abhängigkeiten verändert, um ein Paket installieren zu können. Dies ist dann notwendig, wenn ein System komplett auf eine neue Version umgestellt werden soll und kein „sanfter“ Übergang gewährleistet werden kann.

dselect-upgrade

Mit dieser Option werden auch die Paketinformationen „recommends“ und „suggests“ ausgewertet. Dies ist sonst nur mit dselect möglich. Diese Vorschläge für weitere Pakete sind zur Funktion des gewünschten Pakets nicht zwingend erforderlich, können aber durchaus sinnvoll sein.

install

Diese Option benötigt noch einen oder mehrere Paketnamen. Jedes dieser Pakete (der Paketname, also beispielsweise sendmail, reicht hierbei aus) sowie die noch benötigten Pakete (diese werden automatisch ermittelt) werden auf Basis der Informationen in /etc/apt/sources.list geholt und installiert. Wenn Sie hinter den Namen des Pakets ein Minuszeichen (-) setzen, wird das Paket aus dem System entfernt, wenn es bereits installiert ist. Achten Sie darauf, dass das Minuszeichen (ohne ein Leerzeichen dazwischen) unmittelbar nach dem Paketnamen steht. Diese Funktion kann auch bei Konflikten bei der Installation einzelner Pakete sehr nützlich sein: Sie können so Pakete in einem Vorgang löschen und installieren. Die Option „install" wählt immer die aktuellste verfügbare Version eines Pakets aus. Es kann jedoch sinnvoll sein, auch Zugriff auf ältere Programmversionen zu haben. Hierzu kann ab der Version 0.5.3 von apt folgende Syntax verwendet werden: apt-get install sane/unstable. Durch einen Slash (/) getrennt, kann hinter dem Paketnamen das gewünschte Release eines Pakets angegeben werden.

Eine weitere Möglichkeit ist es, falls verschiedene Versionen eines Pakets verfügbar sind, direkt die gewünschte Versionsnummer anzugeben. Zunächst sollte man mit apt-cache prüfen, welche Versionen eines Pakets verfügbar sind. Mit apt-get install vim=6.0.093-2 kann dann ganz gezielt die gewünschte Version installiert werden.

--reinstall

Diese Option installiert das gewünschte Paket komplett neu, auch wenn es bereits installiert ist. Beachten Sie, dass hierbei auch die Option install anzugeben ist!

remove

Diese Funktion ist analog zu „install", nur mit dem Unterschied, dass die Pakete standardmäßig entfernt werden, anstatt sie zu installieren. Analog zu dem eben Beschriebenen können Sie hier ein Pluszeichen (+) verwenden, um Pakete zu installieren.

check

dient lediglich zur Diagnose. Es wird überprüft, ob sich irgendwelche Unstimmigkeiten in den Paketen finden.

--purge remove

Löscht das gewünschte Paket und alle weiteren Pakete, die direkt von diesem Paket abhängig sind. Weiterhin werden alle zu den Paketen gehörenden Konfigurationsdateien gelöscht.

clean

Dies löscht das lokale Verzeichnis, in dem sich die zu installierenden Pakete befinden. Alles, mit Ausnahme der lock-Datei, wird aus /var/cache/apt/archives/ und /var/cache/apt/archives/partial/ gelöscht.

source

Dieses Kommando holt die notwendigen Dateien zur Erzeugung eines Binärpakets. Es können so angepasste Pakete erzeugt werden oder aber Pakete auf einer anderen Architektur übersetzt werden. Wichtig ist hierbei ein entsprechender „deb-src“-Eintrag in der Datei sources.list. Um alle zu einem Paket gehörenden Dateien von einem Server zu holen, reicht das Kommando apt-get source paketname. Es werden die aktuellen Versionen der Dateien paketname.orig.tar.gz, paketname.dsc und paketname.diff.gz im aktuellen Verzeichnis gespeichert.

Um aus den Quellen direkt ein Binärpaket zu erzeugen, kann die Option -b angegeben werden. Das komplette Kommando lautet dann apt-get source -b paketname. Wenn die benötigten Source-Pakete bereits vorliegen, kann das Binärpaket auch direkt mit dem Kommando dpkg-buildpackage -rfakeroot -us -uc erzeugt werden.

build-dep

Die meisten Software-Pakete benötigen bei der Übersetzung aus den Quellpaketen weitere Entwicklungspakete wie Libraries und Header-Dateien. Diese werden häufig nicht mitgeliefert und liegen in gesonderten Paketen vor. Das Debian Paketsystem sieht so genannte „build dependencies“ vor, in denen alle Abhängigkeiten zur Erzeugung eines Pakets beschrieben sind.

Das Kommando apt-get build-dep paketname sorgt dafür, dass alle zur Übersetzung eines Pakets benötigten Dateien auf dem System vorhanden sind.

moo

Als kleine versteckte Option haben die Entwickler auch noch ein so genanntes „Easter-Egg“ eingebaut:

```
fr@sushi:~$ apt-get moo         (__)         (oo)    /------\/    / |    ||    *
/\---/\     ~~   ~~    ...."Have you mooed today?"...
```

Soweit zu den Parametern. Sie können noch folgende Optionen benutzen, um apt-get zu steuern:

-h

Zeigt die Hilfe zu apt-get an.

-m

Ignoriert eventuell fehlende Pakete.

-d

Holt die gewünschten Pakete vom Server, ohne diese zu installieren.

-f

Behebt die defekten Abhängigkeiten zwischen den Paketen. apt-get -f install versucht, diese automatisch zu reparieren.

-s

Simulation, führt keine Aktionen aus.

-u

Zeigt auch die zu aktualisierenden Pakete an.

-y

Aktualisiert alle Pakete automatisch, indem alle Fragen mit „Yes“ beantwortet werden.

Hier nun ein Beispiel aus der Praxis zur Benutzung von APT:

Beachten Sie bitte, dass Pakete nur mit Superuser-Rechten (root) installiert werden können.

Zuallererst muss apt die Informationen über die verfügbaren Pakete erhalten; hierzu dient das Kommando apt-get update.

```
sushi:/root # apt-get update Hit ftp://192.168.0.5 potato/main Packages Hit ftp://192.168.0.5 potato/main Release Get:1 ftp://192.168.0.5 potato/non-free Packages [78.6kB] Get:2 ftp://192.168.0.5 potato/non-free Release [99B] Fetched 68.5kB in 0s (104kB/s) Reading Package Lists... Done Building Dependency Tree... Done
```

Nachdem das System nun über den aktuellen Stand der Pakete informiert ist, können Sie weitere Pakete mittels apt installieren, was hier am Beispiel von sane gezeigt wird:

```
sushi:/root# apt-get install sane Reading Package Lists... Done Building Dependency Tree... Done The following extra packages will be installed:   libgimp1 libsane The following NEW packages will be installed:   libgimp1 libsane sane 0 packages upgraded, 3 newly installed, 0 to remove and 3 not upgraded. Need to get 703kB of archives. After unpacking 1729kB will be used. Do you want to continue? [Y/n]
```

Beachten Sie hierbei, dass automatisch die benötigten Pakete libgimp1 und libsane ausgewählt wurden. Insgesamt werden also drei Pakete installiert. Weiter unten erhalten Sie noch Informationen über den später benötigten Festplattenplatz der Pakete sowie über die Größe der Pakete, die ja eventuell via FTP erst übertragen werden müssen. Wenn Sie mit den Angaben so einverstanden sind, bestätigen Sie dies mit der Eingabetaste, oder drücken Sie die Taste n und danach die Eingabetaste, um den Vorgang abzubrechen.

Wenn mehrere Pakete gleichzeitig installiert werden sollen, so können diese einfach hintereinander auf der Kommandozeile angegeben werden.

Download aller installierten Pakete

Manchmal kann es wünschenswert sein, alle bereits installierten Pakete eines Systems erneut auf den Rechner zu kopieren, beispielsweise um eine Sicherungskopie zu haben oder um ein System zu duplizieren. Natürlich können die Pakete auch auf eine CD-ROM/DVD gebrannt oder in einen Spiegel eingefügt werden. Vorteilhaft bei dieser Methode ist es, dass ein in sich konsistentes System entsteht, in dem alle Abhängigkeiten erfüllt sind.

Ein Download der gewünschten Pakete kann mit folgenden Befehlen erzeugt werden:

```
# COLUMNS=200 dpkg -l | grep '^ii' | awk '{ print $2 }' > /tmp/pkgliste # cat /tmp/pkgliste | xargs apt-get --download-only --reinstall -y install
```

Die erste Zeile listet via dpkg alle Pakete auf und filtert mittels grep nur die installierten heraus. Danach wird die zweite Spalte in die Datei /tmp/pkgliste geschrieben.

Die zweite Zeile gibt via cat die Liste der Dateien aus. Mit xargs wird jede Zeile an das Kommando apt-get weitergegeben. Die Kombination der apt-get Optionen --reinstall und --download-only sowie -y führt dazu, dass auch bereits installierte Pakete aus dem Netz geholt werden, aber keine Installation durchgeführt wird.

Die Debian Pakete sind nach dem Download im Verzeichnis /var/cache/apt/archives/ zu finden und können mit apt-move (siehe apt-move) in eine Verzeichnisstruktur wie auf einem Debian Spiegel verschoben werden. apt-move ist auch in der Lage, die entsprechenden Packages.gz-Dateien zu erzeugen.

Über die APT Konfigurationsdatei /etc/apt/preferences kann detailliert festgelegt werden, welche Versionen von Paketen installiert werden sollen.

Enthält die Datei /etc/apt/sources.list unterschiedliche Quellen für Pakete, dies kann auch unterschiedliche Distributionen (beispielsweise „stable“ und „testing“) betreffen, so kann es vorkommen, dass ein bestimmtes Paket in verschiedenen Versionen verfügbar ist. APT vergibt für jede Version eines bekannten Paketes eine Priorität. Installiert wird jeweils die Version eines Paketes mit der höchsten Priorität. Die APT Konfigurationsdatei überschreibt die ermittelte Priorität und erlaubt so eine genaue Kontrolle darüber, welche Version installiert wird.

Sind über die Informationen in der sources.list gleiche Versionen eines Paketes aus unterschiedlichen Quellen verfügbar, so wird die in der sources.list zuerst genannte Quelle genutzt. Bei gleichen Versionsnummern eines Paketes ist keine Auswahl der Quelle über die APT Konfigurationsdatei möglich.

Ist keine APT-Konfigurationsdatei oder kein Eintrag in der Konfigurationsdatei vorhanden, der auf ein Paket passt, so wird für dieses Paket die voreingestellte Priorität verwendet, welche der Distribution entspricht. Es besteht die Möglichkeit, ein bestimmtes Ziel-Release festzulegen, aus dem die Pakete installiert werden sollen. Diese Distribution wird als „Target Release“ bezeichnet. Das Ziel-Release kann dem Programm apt-get auf der Kommandozeile übergeben werden oder in der Konfigurationsdatei /etc/apt/apt.conf(siehe apt.conf) gesetzt werden. Auf der Kommandozeile wird das Ziel-Release wie folgt angegeben:

```
apt-get install -t testing paketname
```

Wurde ein Ziel-Release angegeben, so nutzt APT den folgenden Algorythmus, um die Prioritäten eines Paketes zu ermitteln.

Priorität 100

Wenn das Paket bereits installiert ist.

Priorität 500

Wenn das Paket nicht installiert ist und auch nicht zum Ziel-Release gehört.

Priorität 990

Wenn das Paket nicht installiert ist und zum Ziel-Release gehört.

Wird das Ziel-Release nicht angegeben, so vergibt APT die Priorität 100 an alle installierten Pakete und die Priorität 500 an alle nicht installierten Pakete. Danach folgt APT den nachfolgend angeführten Regeln, wobei die Regeln in der beschriebenen Reihenfolge abgearbeitet werden.

Ein Paket wird niemals mit einer kleineren Versionsnummer installiert (Downgrade), wenn die Priorität unter 1001 liegt. „Downgrade“ bedeutet, dass eine nicht aktuelle Version eines Paketes installiert wird, obwohl bereits eine aktuelle Version des Paketes installiert ist. Keine voreingestellte Priorität innerhalb von APT übersteigt den Wert 1000, ein solcher Wert kann ausschließlich über die Voreinstellungsdatei (preferences) gesetzt werden. Der Downgrade eines Paketes ist in jedem Fall als problematisch anzusehen und sollte nach Möglichkeit vermieden werden. In vielen Fällen führt ein Downgrade zu einem nicht funktionierenden Paket oder gar zu schweren Problemen mit dem Gesamtsystem.

Das Paket mit der höchsten Priorität wird installiert.

Wenn zwei oder mehrere Versionen eines Paketes die gleiche Priorität haben, so wird die Version mit der höchsten Versionsnummer installiert.

Wenn zwei oder mehr Versionen eines Paketes die gleiche Priorität und Versionnummer haben, aber die Pakete sich in den Metadaten unterscheiden, oder aber die Option --reinstall auf der Kommandozeile angegeben ist, so wird das nicht installierte Paket ausgewählt.

Am häufigsten ist die Situation anzutreffen, dass ein bereits installiertes Paket (Priorität 100) in einer Version installiert ist, die nicht so aktuell (hoch) ist wie ein Paket, welches über eine Quelle in der Datei /etc/apt/sources.list (Priorität 500 oder 990) verfügbar ist. Dieses Paket wird aktualisiert, wenn das Kommando apt-get install paketname oder apt-get upgrade ausgeführt wird.

Seltener tritt der Fall ein, dass ein bereits installiertes Paket in einer aktuelleren Version vorhanden ist und kein Paket aus einer der in der Datei /etc/apt/sources.list genannten Quellen eine höhere Versionsnummer aufweist. In diesem Fall wird kein Downgrade durchgeführt, wenn das Kommando apt-get install paketname oder apt-get upgrade ausgeführt wird.

Manchmal ist die bereits installierte Version eines Paketes aktueller als die zum Ziel-Release gehörende Version, jedoch nicht so aktuell wie ein Paket, welches aus einer anderen Distribution verfügbar ist. In diesem Fall wird das Paket durch das Kommando apt-get install paketname oder apt-get upgrade aktualisiert, da eine der verfügbaren Versionen eine höhere Priorität besitzt als die aktuell installierte Version.

Die APT Konfigurationsdatei /etc/apt/preferences erlaubt dem Administrator, die Zuweisung von Prioritäten detailliert zu steuern. Die Konfigurationsdatei besteht aus Einträgen, die sich über mehrere Zeilen erstrecken und untereinander durch Leerzeilen getrennt sind. Die Einträge beziehen sich auf ein einzelnes Paket (spezifische Form) oder auf eine Anzahl von Paketen (generelle Form).

Die spezifische Form weist einem Paket eine (Pin-)Priorität zu und übergibt dabei eine gewünschte Version oder einen Versionsbereich. Das folgende Beispiel weist allen Versionen des Paketes perl eine hohe Priorität zu, solange die Versionsnummer mit „5.8“ beginnt.

```
Package: perl Pin: version 5.8* Pin-Priority: 1001
```

Die generelle Form setzt eine Priorität für die angegebene Distribution oder für alle Pakete von einem bestimmten Server, welcher über den vollen Domainnamen (FQDN) identifiziert wird.

Diese generelle Form wirkt sich ausschließlich auf Gruppen von Paketen aus. Folgendes Beispiel weist allen Paketversionen auf einem lokalen Repository mit Paketen eine hohe Priorität zu.

```
Package: * Pin: origin "" Pin-Priority: 999
```

Zu beachten ist hier das Schlüsselwort „origin“. Dieses entspricht nicht dem gleich lautenden Begriff in einer „Release“-Datei! In einer „Release“-Datei wird mit dem Schlüsselwort „origin“ ein Autor oder Hersteller (beispielsweise „Debian“ oder „Ximian“) beschrieben, nicht eine Internetadresse.

Der folgende Eintrag setzt eine geringe Priorität für alle Paketversionen, die zu einer als „unstable“ bezeichneten Distribution gehören.

```
Package: * Pin: release a=unstable Pin-Priority: 50
```

Das letzte Beispiel zeigt einen Eintrag, der eine hohe Priorität allen Paketen zuweist, die zum Release „stable“ gehören und eine Versionsnummer von „3.0“ aufweisen.

```
Package: * Pin: release a=unstable, v=3.0 Pin-Priority: 50
```

Prioritäten (P), die in der APT-Konfigurationsdatei vergeben werden, müssen positive oder negative Integerzahlen sein. Diese werden wie folgt interpretiert.

P > 1000

Installiert ein Paket, auch wenn dies ein „Downgrade“ des Paketes bewirkt.

990 < P <= 1000

Installiert ein Paket, auch wenn es nicht aus dem „Target“-Release stammt, außer die bereits installierte Version ist aktueller.

500 < P <= 990

Installiert ein Paket, wenn kein Paket passend zum „Target“-Release vorhanden ist oder die bereits installierte Version aktueller ist.

100 < P <= 500

Bewirkt, dass eine Version installiert wird, solange keine andere Version aus irgeneiner anderen Distribution verfügbar ist oder aber die installierte Version aktueller ist.

0 < ? <= 100

Installiert eine Version eines Paketes nur, wenn dieses Paket noch nicht installiert ist.

P < 0

Verhindert, dass diese Version eines Paketes installiert wird.

Die Einträge werden in der aufgeführten Reihenfolge in der Konfigurationsdatei abgearbeitet. Wird ein auf das Paket oder die Paketgruppe passender Eintrag gefunden, so werden die folgenden Einträge ignoriert.

Sind die vorab beschriebenen Einträge in einer Konfigurationsdatei in der beschriebenen Reihenfolge abgelegt, so ergibt sich folgende Konfigurationsdatei:

```
Package: perl Pin: version 5.8* Pin-Priority: 1001  Package: * Pin: origin "" Pin-Priority: 999  Package: * Pin: release unstable Pin-Priority: 50
```

In dieser Zusammenstellung gelten folgende Regeln:

Es wird die aktuellste Version des Paketes „perl“ installiert, solange diese Version mit „5.8“ beginnt. Ist eine Version 5.8 von Perl verfügbar und ist die Version 5.9 bereits installiert, so wird ein „Downgrade“ durchgeführt.

Jedes andere Paket, welches nicht das Paket „perl“ ist und auf dem lokalen System verfügbar ist, bekommt eine höhere Priorität. Dies bewirkt, dass nur wenige, nicht lokal

verfügbare Pakete aus dem Netz installiert werden. Dies betrifft auch Versionen des „Target“-Release.

Ein Paket, welches nicht auf dem lokalen System verfügbar ist, aber in einer Quelle in der Datei /etc/apt/sources.list aufgeführt wird und zu einem „unstable“ Release gehört, wird nur installiert, wenn es zur Installation ausgewählt wird und noch keine Version dieses Paketes installiert ist.

Die Paketquellen, welche in der Datei /etc/apt/sources.list aufgeführt werden, enthalten im Idealfall die Dateien Packages (bzw. Packages.gz) und Release, mit denen die an der jeweiligen Quelle verfügbaren Pakete beschrieben werden.

Die Datei Packages befindet sich normalerweise im Pfad …/dists/dist-name/component/arch, also beispielsweise …/dists/stable/main/binary-i386/Packages. Diese Datei besteht aus einem mehrzeiligen Eintrag für jedes Paket, von dem für die APT Prioritäten lediglich die beiden Zeilen Package und Version benötigt werden.

Die Datei Release befindet sich im Pfad …/dists/dist-name, also beispielsweise unter …/dists/stable/Release oder auch …/dists/sarge/Release. Abweichend von der Datei Packages werden fast alle Informationen aus der Release-Datei benötigt. Diese sind:

Archive:

Bezeichnet das Archiv, zu dem alle Pakete dieses Verzeichnisses gehören. Beispielsweise gibt die Zeile Archive: stable an, dass alle Pakete in dem Verzeichnisbaum unterhalb der Datei Release zum „stable“-Release von Debian gehören.

In der APT Konfigurationsdatei wird dieser Wert wie folgt gesetzt:

```
Pin: release a=stable
```

Version:

Diese Zeile bezeichnet die Release-Version der Distribution. Der Wert „3.1“ besagt auch, dass alle Pakete im Verzeichnisbaum zur Release 3.1 „Sarge“ von Debian gehören. Für Versionen wie „testing“ und „unstable“ werden keine Versionsnummern vergeben, da

diese Versionen noch nicht veröffentlicht sind. Um diesen Wert in der APT-Konfigurationsdatei anzugeben, ist eine der folgenden Zeilen notwendig:

```
Pin: release v=3.0 Pin: release a=stable, v=3.0 Pin: release 3.0
```

Component:

Bezeichnet den Abschnitt der Distribution, aus der die Pakete stammen. Beispielsweise bezeichnet die Zeile Component: main in der Release-Datei, dass alle Pakete unterhalb diese Verzeichnisses zum Abschnitt „main“ gehören. Dies bedeutet auch, dass alle diese Pakete einer Lizenz entsprechen, die in den Debian Free Software Guidelines (siehe auch DFSG) festgelegt ist. Diese Angabe wird wie folgt in der APT-Konfigurationsdatei preferences festgelegt

```
Pin: release c=main
```

Origin:

Bezeichnet den Ursprung der Pakete, meist ist hier „Debian“ angegeben. In der APT preferences-Datei sieht diese Zeile wie folgt aus:

```
Pin: release o=Debian
```

Label:

Eine Bezeichnung für die Pakete in diesem Verzeichnisbaum, dieser Eintrag hat meistens den Wert „Debian“. In der APT preferences-Datei sieht diese Zeile wie folgt aus:

```
Pin: release l=Debian
```

Alle verfügbaren Dateien (Packages und Release), die über Quellen in der Datei /etc/apt/sources.list verfügbar sind, werden im Verzeichnis /var/lib/apt/lists/ abgelegt. Dieses Verzeichnis kann durch die Variable Dir::State::Lists in der Datei apt.conf verändert werden. Die Dateinamen der gespeicherten Packages und Release-Dateien werden mit dem Namen des Servers sowie dem Pfad und der Architektur ergänzt, beispielsweise debiananwenderhandbuch.de_debian_dists_stable_main_binary-i386_Packages.

Weiterhin ist es möglich, eine oder mehrere Zeilen, beginnend mit dem Schlüsselwort Explanation:, mit einer Erklärung dieses Eintrages einzufügen.

Die Zeile Pin-Priority: in jedem Eintrag ist nicht zwingend erforderlich. Wird ein solcher Eintrag nicht gefunden, so wird eine um den Wert 1 reduzierte Priorität gegenüber dem vorhergehenden Eintrag vergeben.

Das erste Beispiel zeigt, wie man allen Paketen der „stable“ Distribution eine Priorität über der vorgegebenen (von 500) zuweist. Bei allen anderen Paketen wird die Priorität um den Wert 10 verringert.

```
Explanation: Uninstall or do not install any Debian-originated
Explanation: package versions other than those in the stable distro
Package: * Pin: release a=stable Pin-Priority: 900  Package: * Pin:
release o=Debian Pin-Priority: -10
```

Die beschriebene Konfiguration führt dazu, dass alle Pakete aus der aktuellen „stable“ Distribution installiert werden bzw. bereits installierte Pakete immer aktuell gehalten werden. Darüber hinaus ist es möglich, einzelne Pakete aus anderen Distributionen zu installieren, hier am Beispiel eines Paketes aus „testing“ gezeigt:

```
apt-get install paketname/testing
```

Das so aus „testing“ installierte Paket wird zu einem späteren Zeitpunkt nicht weiter aktualisiert. Dies muss von Hand durch den erneuten Aufruf des oben beschriebenen Kommandos erfolgen.

Das zweite Beispiel setzt eine hohe Priorität für Pakete aus dem „testing“-Zweig, eine etwas niedrigere Priorität für den „unstable“-Zweig und eine noch geringere Priorität für Pakete aus anderen Bereichen.

```
Package: * Pin: release a=testing Pin-Priority: 900  Package: * Pin: release a=unstable Pin-Priority: 800  Package: * Pin: release o=Debian Pin-Priority: -10
```

Diese Konfiguration bewirkt, dass bevorzugt Pakete aus „testing“ in den jeweils aktuellen Versionen installiert werden. Auch hier ist es möglich, mittels

```
apt-get install paketname/unstable
```

gezielt einzelne Pakete aus dem „unstable“-Bereich zu installieren.

Eine besondere Stärke des Debian Paketmanagements ist, dass unterschiedliche Release-Stände verwendet werden können. Hierzu sind zumindest zwei Releases (beispielsweise „woody“ und „testing“) mit entsprechenden Einträgen in der Datei `/etc/apt/sources.list` anzugeben. Nun kann mittels `apt-get --target-release` der gewünschte Release-Stand für ein Paket gewählt werden.

Sinnvoll einsetzbar ist dieses Feature leider nur mit zwei Releases; werden drei Debian Releases (beispielsweise zusätzlich „unstable“) oder aber eine Nicht-Debian Paketquelle

gemischt, so kommt es zu seltsamen Ergebnissen. Die Lösung für dieses Problem ist eine Erweiterung der Datei /etc/apt/preferences.

Hier zunächst ein Beispiel für eine sinnvolle /etc/apt/preferences-Datei. Auf den meisten Systemen wird diese Datei nicht existieren (sie wird auch nicht zwingend benötigt) und muss daher neu angelegt werden.

```
* Track stable: Explanation: see http://www.argon.org/~roderick/apt-pinning.html Package: * Pin: release o=Debian,a=stable Pin-Priority: 900  Package: * Pin: release o=Debian,a=testing Pin-Priority: 400  Package: * Pin: release o=Debian,a=unstable Pin-Priority: 300  Package: * Pin: release o=Debian Pin-Priority: -1      * Track testing: Explanation: see http://www.argon.org/~roderick/apt-pinning.html Package: * Pin: release o=Debian,a=testing Pin-Priority: 900  Package: * Pin: release o=Debian,a=unstable Pin-Priority: 300  Package: * Pin: release o=Debian Pin-Priority: -1
```

Bereits beim einmaligen Aufruf von apt-get kann mit der Option -t der Wert für „APT::Default-Release“ gesetzt werden. Diese Einstellung gilt jedoch lediglich für diesen einen Aufruf von apt-get. Beispiel: apt-get -t unstable install paketname. Dies kann durchaus ein sehr nützliches Feature sein, kann sich aber in anderen Fällen auch problematisch auswirken.

Die gute Seite: Sind in der sources.list die Releases testing und unstable definiert und ist in der Datei apt.conf der Wert für APT::Default-Release auf „testing“ gesetzt, so ergeben sich folgende Prioritäten für das Paket „foo“:

```
release    version        priority                    no -t switch       -t unstable
testing      1.1          990              500 unstable      1.2          500
990
```

Ist das Paket „foo“ nicht installiert und wird apt-get install foo aufgerufen, so wird die Version 1.1 aus „testing“ installiert, da diese die höchste Priorität hat.

Mittels apt-get -t unstable install foo kann gezielt die Version 1.2 aus „unstable“ installiert werden, da für diesen einen Aufruf von apt-get die Priorität erhöht wird.

Übrigens kann auch die Ausgabe des Kommandos apt-cache policy (ohne weitere Parameter) als Gerüst für eine eigene Datei /etc/apt/preferences dienen:

```
fr@inari:~$ apt-cache policy Paketdateien:  100 /var/lib/dpkg/status
release  500 http://security.debian.org testing/updates/main Packages
release o=Debian,a=testing,l=Debian-Security,c=main  500
http://ftp2.de.debian.org testing/main Packages      release
o=Debian,a=testing,l=Debian,c=main      origin ftp2.de.debian.org
Festgehaltene Pakete (»Pin«):
```

Die Manpage zu dieser Konfigurationsdatei wird mittels man apt_preferences 5 angezeigt und enthält weitere Informationen.

4.15 apt-get

◁ 4.14 apt-cdrom △ 4.16 apt - Offline nutzen ▷

4.16 apt - Offline nutzen

4.15 apt-get 4.17 apt-key

Normalerweise wird apt mit direktem Zugriff zu einem Archiv benutzt, sei es, dass die Daten auf einer lokalen Festplatte liegen, von einer CD-ROM kommen oder über ein Netzwerk installiert werden sollen. Trotzdem kann es sinnvoll sein, wenn Sie keine aktuellen CDs zur Hand haben oder Ihnen der Download der Dateien zu langwierig ist, die Daten von einem Rechner mit den aktuellen Paketen auf ein anderes Medium, zum Beispiel ein ZIP, zu kopieren und zu installieren.

Problematisch ist dabei, dass ein solches Medium nicht ausreicht, um eine komplette Debian Distribution aufzunehmen. Aber es reicht ja auch aus, nur die benötigten Pakete zu kopieren. Sie müssen also mit apt eine Liste der benötigten Pakete erstellen und diese auf dem Rechner mit einer schnellen Anbindung ans Netz herunterladen. Hierzu ist das Programm wget gut geeignet.

Weiterhin müssen Sie apt dazu bringen, die Paketdateien von dem neuen Medium zu lesen. Dieses sollte mit einem Dateisystem formatiert sein, das mit langen Dateinamen umgehen kann; dies wären zum Beispiel: ext2, vfat oder fat32.

Wenn apt auf beiden Rechnern installiert ist, können Sie über die Statusdatei (status) feststellen lassen, welche Pakete auf dem Zielrechner fehlen. Die Verzeichnisstruktur auf dem Medium sollte wie folgt aussehen:

```
/medium/ archives/ partial/ lists/ partial/ status sources.list apt.conf
```

In der Konfigurationsdatei teilen Sie apt mit, dass die Debian Pakete auf dem Medium gespeichert werden sollen und dass dort auch die Konfigurationsdateien zu finden sind. Die Datei sources.list sollte den Server enthalten, von dem die Pakete geholt werden

sollen. Die Datei status ist eine Kopie der Datei /var/lib/dpkg/status von Ihrem Zielrechner.

Die Datei apt.conf enthält die nötigen Informationen, um das Medium zu benutzen:

```
APT {   /* nur notwendig, wenn die beiden Rechner       nicht die gleiche Architektur haben */   Architecture "i386";    Get::Download-Only "true"; };  Dir {   /* die status-Datei auf dem Medium nutzen. */ State "/medium/";   State::status "status";    /* auf dem Medium cachen */   Cache::archives "/medium/archives/";    /* da ist auch die Datei sources.list */   Etc "/medium/"; };
```

Auf Ihrem Zielrechner mounten Sie zuerst das Medium und kopieren die Datei /var/lib/dpkg/status dorthin. Erzeugen Sie dann die Verzeichnisse, die oben in der Struktur gezeigt sind (archives/partial/ und lists/partial/).

Die weiteren Schritte können Sie auf dem zweiten Rechner durchführen. Passen Sie die Datei sources.list an. Dann führen Sie Folgendes auf diesem Rechner aus:

```
export APT_CONFIG="/medium/apt.conf" apt-get update apt-get dist-upgrade
```

Sie können auch alle anderen apt-Optionen oder gpk-application benutzen.

Wenn alle Dateien auf das Medium übertragen worden sind, können Sie diese zu Ihrem Zielrechner transportieren und die Pakete wie folgt installieren:

```
export APT_CONFIG="/medium/apt.conf" apt-get check apt-get --no-d -o dir::etc::status=/var/lib/dpkg/status dist-upgrade
```

Bitte beachten Sie, dass hierbei wieder die lokale status-Datei benutzt wird!

Wenn Sie nicht die Möglichkeit haben, die Dateien auf einem Rechner mit apt aus dem Netz zu kopieren, können Sie alternativ auch das Programm wget benutzen. Dieses ist nicht Teil der Debian GNU/Linux-Basisinstallation, Sie müssen also zunächst das Paket installieren.

wget dient zum Kopieren von Webseiten inklusive der dazugehörigen Grafiken und eventueller weiterer mit dieser Seite verlinkter Seiten. Wir werden wget aber zum gezielten Kopieren der benötigten Debian Pakete von einem Server benutzen.

Sie können die Pakete mittels wget kopieren, indem Sie apt-get mit der Option --print-uris dazu benutzen, eine Liste der benötigten Pakete mit der kompletten URL für wget zu erzeugen.

Entgegen dem eben vorgestellten Beispiel brauchen Sie hier keine extra Konfigurationsdateien (außer natürlich der Datei /etc/apt/sources.list, die von apt benötigt wird). Führen Sie die folgenden Kommandos auf dem Rechner aus, auf dem die Pakete später installiert werden sollen.

```
apt-get update apt-get -qq --print-uris dist-upgrade > uris awk '{print "wget -O " $2 " " $1}' < uris > /medium/wget-script
```

Drücken Sie „n“ nach dem ersten Kommando und prüfen Sie, ob die Auswahl korrekt ist. Die benötigten Pakete wurden nun zur späteren Installation markiert. In der zweiten Zeile wird mit apt-get und den entsprechenden Optionen eine Liste der zu installierenden Pakete erzeugt und in die Datei uris geschrieben. Das letzte Kommando ergänzt die Liste der Pakete um die nötigen Aufrufe des Kommandos wget.

Die Datei /medium/wget-script enthält nun eine Liste mit Kommandos, die die benötigten Pakete aus dem Netz holen. Sie können dieses jetzt auf dem Medium ausführen:

```
cd /medium sh -x ./wget-script
```

Hierbei ist es unerheblich, ob das Transportmedium sich noch in dem Rechner befindet, auf dem die Pakete installiert werden sollen (was keinen Sinn machen würde, Sie könnten sich den Umweg über wget dann sparen), oder ob Sie das Medium in einem Rechner mit einer besseren Netzanbindung anmelden und dort das Skript ausführen. Die Pakete werden lediglich auf das Medium übertragen und nicht installiert.

Wenn alle Dateien übertragen wurden, können Sie das Medium zu Ihrem Zielrechner transportieren und die Pakete mit folgendem Kommando installieren:

```
apt-get -o dir::cache::archives="/medium/" dist-upgrade
```

4.16 apt - Offline nutzen

4.15 apt-get 4.17 apt-key

4.17 apt-key

4.16 apt - Offline nutzen 4.18 apt-rdepends

Mit apt-key kann der von apt zur Authentifizierung von Paketen verwendete Schlüsselring verwaltet werden. Pakete, die mit einem Schlüssel aus dem Schlüsselring authentifiziert wurden, werden als vertrauenswürdig angesehen.

add

fügt einen Schlüssel aus einer Datei oder der Standardeingabe (-) dem Schlüsselring hinzu.

del

Löscht den Schlüssel mit der angegebenen KeyID aus dem Schlüsselring

list

Zeigt die Liste der verfügbaren Schlüssel an.

update

Aktualisiert den lokalen Schlüsselring mit den offiziellen Informationen aus dem Debian Schlüsselring.

4.17 apt-key

4.16 apt - Offline nutzen 4.18 apt-rdepends

4.18 apt-rdepends

4.17 apt-key 4.19 apt-cache

Hiermit können die „reverse-dependencies“ eines Pakets ermittelt werden. Ähnliche Informationen kann auch apt-cache liefern. Ohne weitere Angaben wird ein komplettes Listing aller Abhängigkeiten erzeugt; hier nur ein kleines Beispiel:

```
fr@inari:~$ apt-rdepends bash bash   Depends: base-files (>= 2.1.12)
base-files   Depends: awk   Depends: base-passwd (>= 2.0.3.4) awk
base-passwd   Depends: libc6 (>= 2.3.1-1) libc6
```

4.18 apt-rdepends

4.17 apt-key 4.19 apt-cache

4.19 apt-cache

4.18 apt-rdepends 4.20 apt-cacher

Mit apt-cache können die verschiedensten Abfragen zu den bekannten Paketen (also auch zu den noch nicht installierten) erstellt werden. Dies beschränkt sich nicht nur auf Paketnamen oder die zu einem Paket gehörenden Dateinamen, sondern es können auch Informationen zu den Metadaten erfragt werden.

Wenn Sie nicht die Option -h oder --help benutzen, muss mindestens eines der folgenden Kommandos angegeben werden.

add

fügt die angegebene Package-Datei dem Package-Cache hinzu.

gencaches

Diese Option bewirkt das Gleiche wie apt-get check. Es werden die Source- und Package-Caches aus den Informationen in /etc/apt/sources.list und /var/lib/dpkg/status erstellt.

showpkg

Diese Option zeigt Informationen über die weiterhin auf der Kommandozeile angegebenen Pakete. Es werden die verfügbaren Versionen der Pakete sowie die gesamten Abhängigkeiten dieser Pakete angezeigt. Hierbei wird zwischen so genannten „Forward"-Dependencies und „Reverse"-Dependencies unterschieden. Forward-Dependencies (dies sind die im normalen Sprachgebrauch verwendeten Abhängigkeiten) beziehen sich auf die Pakete, die das angefragte Paket benötigt, um einwandfrei zu funktionieren. Reverse-Dependecies beschreiben die Pakete, die das angefragte Paket benötigen, um zu funktionieren. Ein Beispiel: apt-cache showpkg bash gibt folgendes Ergebnis aus:

```
Package: bash Versions:  2.05a-
12(/var/lib/apt/lists/192.168.1.1_home_ftp_debian_dists_sid_
main_binary-i386_Packages)(/var/lib/dpkg/status) 2.05a-
11(/var/lib/apt/lists/192.168.1.1_home_ftp_debian_dists_woody_
main_binary-i386_Packages)  Reverse Depends:    kernel-patch-ltt,bash
2.0   kernel-patch-lkcd,bash 2.0   kernel-patch-kdb,bash 2.0   kernel-
patch-acl,bash 2.0   horde2,bash   cron-apt,bash 2.03-6   common-lisp-
```

controller,bash 2.04-9 cdcontrol,bash 2.0 bash-doc,bash 2.03-1 bash-builtins,bash 2.05a-12 base-files,bash 2.03-3 txt2regex,bash 2.04 openmosix,bash 2.0 mosix,bash 2.0 mkrboot,bash mason,bash kernel-patch-ulog,bash 2.0 kernel-patch-ttl,bash 2.0 kernel-patch-ltt,bash 2.0 kernel-patch-kiobuf-bigmem,bash 2.0 kernel-patch-kiobuf,bash 2.0 kernel-patch-kdb,bash 2.0 kernel-patch-irc,bash 2.0 kernel-patch-badram,bash 2.0 htmlheadline,bash 2.04-1 horde,bash 2.03-6 gopherweblink,bash gibraltar-bootsupport,bash foomatic-bin,bash 2.05 diffmon,bash 2.0 cron-apt,bash 2.03-6 common-lisp-controller,bash 2.04-9 colorgcc,bash cdcontrol,bash 2.0 bug,bash 2.04-1 bash-doc,bash 2.03-1 bash-builtins,bash 2.05a-11 base-files,bash 2.03-3 Dependencies: 2.05a-12 - base-files (2 2.1.12) libc6 (2 2.2.4-4) libncurses5 (2 5.2.20020112a-1) grep-dctrl (0 (null)) bash-completion (0 (null)) bash-doc (1 2.05-1) bash-completion (0 (null)) 2.05a-11 - base-files (2 2.1.12) libc6 (2 2.2.4-4) libncurses5 (2 5.2.20020112a-1) bash-completion (0 (null)) bash-doc (1 2.05-1) bash-completion (0 (null)) Provides: 2.05a-12 - 2.05a-11 - Reverse Provides:

Es müssen also, damit dieses Paket funktioniert, mindestens die unter „Dependencies“ aufgeführten Pakete installiert sein. Eine weitere wichtige Information sind die „Reverse Depends“. Dies sind Pakete, die von diesem Paket (bash) abhängen.

stats

Über den aktuellen Cache wird eine Statistik ausgegeben. Es sind keine weiteren Argumente notwendig. Folgende Informationen werden ausgegeben:

fr@sushi:~$ apt-cache stats Total Package Names : 8953 (358k) Normal Packages: 6872 Pure Virtual Packages: 259 Single Virtual Packages: 176 Mixed Virtual Packages: 115 Missing: 1531 Total Distinct Versions: 9133 (438k) Total Dependencies: 47613 (1143k) Total Ver/File relations: 19322 (309k) Total Provides Mappings: 1889 (37.8k) Total Globbed Strings: 110 (1273) Total Dependency Version space: 179k Total Slack space: 86.3k Total Space Accounted for: 2374k

Total Package Names - Anzahl der Paketnamen, die im Cache gefunden wurden.

Normal Packages - Dies sind Pakete, deren Namen in einer Abhängigkeit zu einem anderen Paket stehen; hierunter fällt eine große Zahl der Pakete.

Pure Virtual Packages - Anzahl der „virtuellen" Paketnamen im Cache. Dies sind Pakete, die ein Paket zur Verfügung stellen, deren Name aber nichts mit diesem zu tun hat. Als Beispiel sei hier mail-transport-agent genannt: Einige Pakete (beispielsweise exim, qmail und sendmail) stellen das Paket mail-transport-agent zur Verfügung, es gibt aber kein Programm mit dem Namen mail-transport-agent.

Single Virtual Packages - Dies ist die Anzahl der Pakete, die ein virtuelles Paket zur Verfügung stellen, das aber nur einmal im Cache auftaucht. Das virtuelle Paket X11-text-viewer wird beispielsweise nur von xless zur Verfügung gestellt.

Mixed Virtual Packages - Anzahl der Pakete, die sowohl als virtuelle als auch als reale Pakete vorhanden sind. debconf ist als reales Paket vorhanden, wird aber auch von debconf-tiny zur Verfügung gestellt.

Missing - Pakete, die in einer Abhängigkeit benannt werden, aber nicht im Cache zu finden sind. Dies kann vorkommen, wenn kein Zugriff auf eine komplette Debian Distribution gegeben ist oder wenn Pakete aus der Distribution entfernt wurden.

Total Distinct Versions - Die Anzahl der Paketversionen im Cache. Diese ist im Normalfall gleich der Anzahl der gesamten Pakete. Wenn jedoch zwei Distributionen (beispielsweise „stable" und „testing") benutzt werden, kann es vorkommen, dass mehrere Versionen eines Pakets verfügbar sind. In diesem Fall kann die Zahl deutlich über der der gesamten Pakete liegen.

Total Dependencies - Die Anzahl der gesamten Abhängigkeiten zwischen allen Paketen im Cache.

Total Ver/File relations - Die Gesamtzahl an Version/Datei-Beziehungen.

Total Provides Mappings - Die Gesamtzahl an Bereitstellungen von Mappings.

Total Globbed Strings - Gesamtanzahl von Mustern

Total Dependency Version space - Gesamtmenge an Abhängigkeits/Versionsspeicher.

Total Slack space - Gesamtmenge an Slack Speicher.

Total Space Accounted for - Gesamtmenge des Speichers.

dump

Zeigt eine kurze Information zu jedem Paket an. Dies ist für den normalen Benutzer wenig sinnvoll und ist zur Fehlersuche für Entwickler gedacht.

dumpavail

Zeigt eine Liste der verfügbaren Pakete.

unmet

Zeigt eine Zusammenfassung aller nicht erfüllten Abhängigkeiten.

show

Hat einen ähnlichen Effekt wie das Kommando dpkg --print-avail und zeigt die Paketinformationen für die angegebenen Pakete.

search

Führt eine Volltextsuche über alle verfügbaren Paketdateien durch. Es können reguläre Ausdrücke benutzt werden. Die Paketnamen und Beschreibungen werden nach dem Suchbegriff durchsucht, und es werden der Paketname und die Kurzbeschreibung der entsprechenden Pakete ausgegeben. Wenn die Option --full angegeben wird, entspricht die Ausgabe der von show. Mit der Option --names-only wird die Paketbeschreibung nicht durchsucht; die Suche beschränkt sich auf den Paketnamen.

Mehrere Suchargumente können angegeben werden und werden dann über eine UND-Verknüpfung ausgewertet.

depends

Zeigt alle Abhängigkeiten eines Pakets an. Weiterhin werden alle anderen Pakete angezeigt, die die geforderten Abhängigkeiten erfüllen können.

```
fr@sushi:~$ apt-cache depends bash bash  Depends: base-files
PreDepends: libc6  PreDepends: libncurses5  Suggests: grep-dctrl
Conflicts: <bash-completion>  Replaces: bash-doc  Replaces: <bash-
completion>
```

policy

Die Option policy zeigt zu einem Paket die installierte sowie die verfügbaren Versionen aus den erreichbaren Quellen an.

```
fr@sushi:~$ apt-cache policy bash bash:   Installed: 2.05b-5
Candidate: 2.05b-5   Version Table:  *** 2.05b-5 0         500
ftp://ftp.freenet.de sid/main Packages         100 /var/lib/dpkg/status
2.05b-3 0         500 ftp://ftp.freenet.de testing/main Packages         500
ftp://ftp.freenet.de sarge/main Packages      2.05a-11 0         500
ftp://ftp.freenet.de woody/main Packages
```

pkgnames

Zeigt eine Liste aller Paketnamen. Optional kann eine Zeichenkette angegeben werden, die als Suchpräfix verwendet wird.

dotty

Dieser Option kann eine Liste von Paketnamen mitgegeben werden. Die Ausgabe erfolgt in einem Format, das vom Programm dot aus dem Paket GraphVis (http://www.research.att.com/sw/tools/graphviz/) gelesen werden kann. GraphVis ist auch als Debian Paket verfügbar, kann also mittels apt-get installiert werden. Mit dieser Option und dem Programm dot kann eine grafische Darstellung der Paketabhängigkeiten erstellt werden. Normalerweise werden alle Abhängigkeiten verfolgt, was zu sehr großen Dateien führen kann.

Um die Grafik etwas zu verkleinern, kann die Zeile

```
APT::Cache::GivenOnly "true";
```

in der APT-Konfigurationsdatei /etc/apt/apt.conf hinzugefügt werden. Dies bewirkt, dass nur die auf der Kommandozeile angegebenen Pakete ausgewertet werden. Es kann passieren, dass eine sehr lange Liste von Paketen zum Abbruch des Programmes führt.

Das Ergebnis ist eine Grafik, die unterschiedliche Symbole für Pakete verwendet. Normale Pakete werden als schwarzes Rechteck dargestellt. Direkt abhängige Pakete (provides) werden als Trapez dargestellt. Diamant-förmige Pakete sind mixed-provides, und fehlende Pakete werden als Hexagon dargestellt. Orange-farbene Rechtecke bedeuten,

dass die rekursive Suche abgebrochen wurde. Blaue Linien stellen Pre-Dependencies dar, und grüne Linien stehen für Konflikte.

Hier zwei kleine Beispiele dafür, wie diese Option sinnvoll eingesetzt werden kann:

```
apt-cache dotty vim | dot -Tps > packgraph.ps  apt-cache dotty $(dpkg --get-selections | grep -v deinstall \          | awk 'print $1') | dot -Tps > packgraph.ps
```

Beide Beispiele erzeugen eine Postscript-Datei packgraph.ps. Diese kann beispielsweise mit dem Programm gv angezeigt werden.

Suchen ohne apt-cache

Die aus der Bash-Shell bekannte Vervollständigung von Dateinamen (mittels **TAB**-Taste) kann auch auf Debian Pakete ausgeweitet werden. Hierzu ist die Datei /etc/bash_completion zu „sourcen“, beispielsweise durch das Kommando source /etc/bash_completion. Nun kann auch apt-get install durch das Drücken der Taste **TAB** vervollständigt werden.

4.19 apt-cache

4.18 apt-rdepends 4.20 apt-cacher

4.20 apt-cacher

4.19 apt-cache 4.21 apt-proxy

apt-cacher unterscheidet sich deutlich vom zuvor beschriebenen apt-cache. Die Problematik, in einem Netzwerk verschiedene Debian Systeme mit Paketen zu versorgen und dabei nicht für jedes System alle Pakete erneut aus dem Netz zu holen, ist sicher jedem Administrator vertraut. apt-cacher bietet für dieses Problem eine elegante Lösung und lässt sich besonders effektiv in einem Netzwerk mit langsamer Anbindung an das Internet einsetzen.

apt-cacher wurde ursprünglich von Nick Andrew entwickelt, um zwei Debian Systeme hinter einer Modem-Anbindung mit Paketen zu versorgen. Die Entwicklung ist mittlerweile jedoch weiter fortgeschritten und hat apt-cacher zu einem leistungsfähigen Caching-System für Debian Pakete gemacht. apt-cacher lässt sich auch auf anderen Linux-Distributionen, wie beispielsweise RedHat/Fedora oder Novell/SuSE, im Zusammenspiel mit APT einsetzen und ist auch via „Fink“ unter MacOS lauffähig.

Fink: Mac OS X durch freie Software ergänzen

Fink ist ein Projekt, das es sich zur Aufgabe gemacht hat, die ganze Welt der Unix-Open-Source-Software auf Darwin und Mac OS X zu bringen. Daraus ergeben sich zwei Hauptziele. Erstens: bereits existierende Open-Source-Software so zu modifizieren, dass sie unter Mac OS X kompiliert bzw. ausgeführt werden kann (diesen Prozess nennt man „Portieren“); zweitens: die Ergebnisse den Nutzern zugänglich zu machen, und zwar als eine zusammenhängende, komfortable Distribution, die den Gewohnheiten der Linux-Nutzer entspricht. (Dieser Prozess wird „Packaging“ genannt.) Das Projekt bietet sowohl vorkompilierte Binär-Pakete als auch ein vollautomatisiertes System, das es ermöglicht, direkt von den Quellcodes zu kompilieren.

Um diese Ziele zu erreichen, vertraut Fink auf die exzellenten Paket-Management-Tools, die vom Debian-Projekt entwickelt wurden: dpkg, dselect und apt-get. Darüber hinaus bietet Fink seinen eigenen Paket-Manager, genannt fink, an. Man kann fink als eine „build engine“ ansehen - es benutzt die Paketbeschreibungen und erstellt daraus binäre Debian Pakete. Während dieses Prozesses lädt es den originalen Quellcode aus

dem Internet, führt, wenn nötig, Patches aus und geht dann durch den gesamten Konfigurations- und Build-Prozess des Pakets. Zuletzt packt es die Ergebnisse in ein Archiv, das dann von dpkg installiert werden kann.

Da Fink auf Mac OS X aufbaut, gilt die strikte Regel, es zu verhindern, dass es dem Basissystem irgendwie in die Quere kommt. Daraus folgt, dass Fink in einer eigenständigen Ordnerstruktur residiert und die Infrastruktur anbietet, um es einfach zu benutzen.

Im Gegensatz zu anderen Caching-Programmen läuft apt-cacher nicht als eigenständiger Server, sondern als CGI-Skript innerhalb des Apache-Webservers auf einem System im Netzwerk. Dies hat den Vorteil, dass apt-cacher kleingehalten werden und beispielsweise auf die vom Apache bereitgestellten Protokolle (http) zurückgreifen kann. Ebenfalls können die Zugriffsbeschränkungen von Apache benutzt werden, um nur bestimmten Systemen im Netz Zugriff zu gewähren.

apt-cacher wird auf einem System im Netz installiert; dieses dient als lokaler Cache. Alle anderen Systeme im Netz müssen in der Konfiguration so angepasst werden, dass sie im Folgenden ausschließlich den apt-cacher Server verwenden.

apt-cacher bearbeitet Anfragen der anderen Systeme und holt bei Bedarf das gewünschte Debian Paket aus dem Netz. Dieses wird lokal auf dem Server gespeichert und steht so weiteren Systemen sofort zur Verfügung. Das Ganze verhält sich für die anderen Systeme im Netz völlig transparent, alle gewohnten Werkzeuge zur Paketverwaltung können weiterhin verwendet werden.

Die eigentliche Installation von apt-cacher erfolgt ausschließlich auf dem Server. An den anderen Systemen, die apt-cacher nutzen sollen, sind nur Anpassungen notwendig.

Um einen apt-cacher-Server zu betreiben, sind keine besonderen Anforderungen an die Hardware zu erfüllen. Lediglich der Festplattenplatz sollte, je nach Nutzung, ausreichend bemessen sein. Werden die Adressen im Netzwerk via DHCP vergeben, so ist darauf zu achten, dass der Name/die IP-Adresse des Servers immer beibehalten wird.

Das Kommando

```
apt-get install apt-cacher
```

installiert die notwendige Software auf dem Server. Ebenso werden, falls noch nicht geschehen, der Apache-Webserver und einige weitere Pakete installiert. Abschließend ist noch der Webserver mittels

```
/etc/init.d/apache restart
```

von der Änderung zu unterrichten.

Um die Installation zu testen, kann mit einem Webbrowser die Adresse http://localhost/apt-cacher aufgerufen werden. Läuft der Browser nicht direkt auf dem Server, so ist natürlich „localhost“ durch den Rechnernamen/die IP-Nummer zu ersetzen. Ist die Installation geglückt, so wird eine Informationsseite von apt-cacher ausgegeben.

Auf der Client-Seite muss keinerlei Software installiert werden, lediglich eine Anpassung der Konfigurationsdatei für APT ist notwendig. Dabei können die bisherigen Einträge übernommen werden, müssen jedoch jeweils um die Angabe des Cache-Servers ergänzt werden.

```
deb http://cacheserver/apt-cacher/ftp.de.debian.org/debian stable main contrib non-free
```

Wichtig ist nur, dass diese Einträge auf allen Clients im Netz einheitlich sind.

Nun steht bereits ein funktionsfähiger Cache im Netz zur Verfügung. Es können jedoch noch einige Anpassungen an der Konfiguration vorgenommen werden.

Die Konfigurationsdatei findet sich unter: /etc/apt-cacher/apt-cacher.conf

admin_email

Die E-Mail-Adresse, die in Reports angezeigt wird und an die bei Problemen Mails versendet werden.

generate_reports

Diese Option kann auf den Wert „1“ gesetzt werden, um tägliche Nutzungsreports zu erstellen.

cache_dir

Ein Verzeichnis, in dem die Pakete und Paketlisten auf dem Server gespeichert werden.

logfile

In diesem Logfile wird die Nutzung des Servers dokumentiert. Diese Datei wird ebenfalls zur Erstellung von Reports genutzt.

errorfile

Logdatei für Fehlermeldungen.

expire_hours

Anzahl von Stunden, nach denen Paketlisten gelöscht werden. Dies beeinflusst nicht das Löschen von Paketen; diese werden entfernt, wenn sie nicht mehr in den Paketlisten aufgeführt werden.

http_proxy

Adresse und Port eines weiteren, externen Proxy-Servers.

use_proxy

Aktiviert die Nutzung des Proxys.

debug

Schaltet den Debug-Modus ein oder aus.

Ist die entsprechende Option aktiviert, so generiert apt-cacher einen täglichen Report. Dieser kann über die URL http://localhost/apt-cacher/report abgerufen werden.

4.20 apt-cacher

4.19 apt-cache 4.21 apt-proxy

4.21 apt-proxy

4.20 apt-cacher 4.22 apt-move

apt-proxy ist ein Cache, der zusammen mit apt von allen Debian Systemen aus im Netz genutzt werden kann. Lokale Netzwerke verfügen normalerweise über eine höhere Bandbreite, als für das gesamte Netz nach außen ins Internet zur Verfügung steht. Durch einen gemeinsam genutzten Cache können bereits auf einem System installierte Pakete auch allen anderen Systemen zur Verfügung gestellt werden, ohne dass diese Pakete bei der Installation auf den einzelnen Clients nochmals aus dem Internet geholt werden.

apt-proxy kann als Cache effizienter mit Debian Paketen umgehen als beispielsweise Squid. Dies liegt zum einen daran, dass apt-proxy die Paketdateien auswertet und alte Versionen löscht, die nicht mehr benötigt werden. Zum anderen verwendet apt-proxy zur Übertragung von Dateien das Programm rsync, so dass abgebrochene Downloads fortgesetzt werden können. Alternativ können auch die Protokolle HTTP und FTP zum Download genutzt werden.

Nach der Installation muss zunächst in der Datei apt-proxy.conf ein gut erreichbarer Debian Server eingetragen werden. Von diesem werden dann alle Pakete geholt. Auf allen Clients, die den APT-Proxy benutzen sollen, ist die Konfiguration so anzupassen, dass der Rechner, auf dem apt-proxy installiert ist, als Quelle verwendet wird. apt-proxy läuft auf dem Port 9999; die Einträge in der Datei sources.list auf den Clients müssen folgendes Format haben:

```
deb http://SERVER:9999/main woody main contrib non-free deb
http://SERVER:9999/non-US woody/non-US main contrib non-free deb
http://SERVER:9999/security woody/updates main contrib non-free
deb-src http://SERVER:9999/main woody main contrib non-free deb-
src http://SERVER:9999/non-US woody/non-US main contrib non-free
deb-src http://SERVER:9999/security woody/updates main contrib non-
free
```

Hierbei ist zu beachten, dass „SERVER“ durch den entsprechenden Rechnernamen (bzw. die IP-Nummer) ersetzt wird. Nachdem diese Anpassungen vorgenommen worden sind, wird auf einem der Clients das Kommando apt-get update aufgerufen, um auf dem Server die Verzeichnisse zu initialisieren und die Paketlisten verfügbar zu machen. Nun können alle Clients diesen zentralen Dienst transparent nutzen.

Wenn bereits auf dem Server ein Verzeichnis mit Debian Paketen vorliegt, so kann dies mittels apt-proxy-import in den Cache von apt-proxy integriert werden.

4.21 apt-proxy

4.20 apt-cacher 4.22 apt-move

4.22 apt-move

4.21 apt-proxy 4.23 apt-ftparchive

apt-move ist ein Shell-Skript, mit dem eine Anzahl von lokal vorliegenden Debian Paketen in eine Verzeichnisstruktur verschoben werden kann, die wiederum einem Debian Archiv entspricht. Die Dateien werden dabei in einer Verzeichnisstruktur unterhalb von $LOCALDIR/pool/... abgelegt; die Variable $LOCALDIR kann dabei in der Konfigurationsdatei festgelegt werden. Hauptsächlich wurde apt-move entwickelt, um via apt-get aus dem Netz übertragene Dateien aus dem Datei-Cache in ein Debian Archiv zu übernehmen. Die Konfiguration kann jedoch so angepasst werden, dass jede beliebige Sammlung von Paketen (beispielsweise von CD-ROMs) dem Archiv hinzugefügt werden kann.

Weiterhin besteht die Möglichkeit, mittels der Optionen sync und mirror eine Kopie eines Debian Servers zu erstellen (oder von Teilen des Servers). Auch können überflüssige, veraltete Pakete gelöscht werden, und es können lokale Versionen der Dateien Packages.gz und Sources.gz erzeugt werden.

apt-move kennt eine ganze Reihe von Kommandos und Optionen:

$Id: apt-move,v 1.96 2003/07/27 04:18:04 herbert Exp $ Usage: apt-move [-c conffile] [-d dist] [-fqt] COMMAND Commands: get - update your master files from local apt. getlocal - alias of get. fsck - fix broken repositories, use with caution. move - move cache files into mirror tree. movefile - move files specified on the command line. delete - delete obsolete packages. packages - create new local Packages files. update - alias for 'get move delete packages'. local - alias for 'move delete packages'. localupdate - alias for 'getlocal move delete packages'. mirror - update your local mirror from remote rsync site. sync - same as mirror, but only gets packages that you currently have installed on your system. exclude - prints a list of all packages EXCLUDED from the mirror by the .exclude file (ignores -t). listbin - prints lists of packages which can serve as the input to mirrorbin. Takes the arguments mirror, sync, or repo. listsrc - same as listbin, but lists source packages. mirrorbin - same as mirror, but gets the packages specified on stdin. mirrorsrc - same as mirrorbin, but gets source packages.

Options: -c Specify an alternative configuration file. -d Override the DIST setting. -f Override the MAXDELETE setting (use with caution). -q Be quiet; suppress normal output. -t Show what apt-move would do, but do not actually do anything. See the apt-move(8) manpage for further details.

Bevor apt-move eingesetzt werden kann, ist die Konfigurationsdatei /etc/apt-move.conf anzupassen. Diese Datei ist, wie alle Konfigurationsdateien auf einem Debian System, recht gut kommentiert. Wichtig ist es, die Optionen APTSITES (entsprechend den Einträgen in der Datei /etc/apt/sources.list), LOCALDIR und DIST anzupassen. Diese beschreiben die Server, von denen die Dateien geholt werden sollen, den lokalen Bereich, in dem die Pakete gespeichert werden sollen, und die gewünschte Distribution.

Nach der Anpassung dieser drei Optionen wird apt-move zumindest keinen größeren Schaden anrichten. Weitere Anpassungen können später noch vorgenommen werden. Beispielsweise ist es sinnvoll, nach einiger Zeit nicht mehr benötigte Dateien zu löschen; dies wird durch Setzen der Option DELETE auf yes erreicht.

In jedem Fall sollten die ersten Versuche mit apt-move immer den Parameter -t beinhalten; dieser testet nur die Einstellungen, und es werden keine Aktionen durchgeführt.

Sollen bestimmte Dateien (beispielsweise Kernel-Sourcen) ausgeschlossen werden, so können diese in einer Exclude-Datei festgelegt werden. Ein Beispiel für eine solche Datei finden Sie unter /usr/share/doc/apt-move/examples/SAMPLE.exclude.

4.22 apt-move

4.21 apt-proxy 4.23 apt-ftparchive

4.23 apt-ftparchive

4.22 apt-move 4.24 apt-show-source

apt-ftparchive erzeugt Index-Dateien, die von apt benutzt werden können, um auf ein Paketverzeichnis zuzugreifen. Die Funktionalität ist vergleichbar mit der von dpkg-scanpackages und dpkg-scansources; apt-ftparchive kann diese Programme ersetzen.

Die Syntax von apt-ftparchive lautet:

```
apt-ftparchive [-hvdsq] [--md5] [--delink] [--readonly] [--contents] [-
o= config string] [-c=file] {packages path... [override [pathprefix]] |
sources path... [override [pathprefix]] | contents path | release path |
generate config-file section... | clean config-file}
```

Intern verwendet apt-ftparchive eine (binäre) Datenbank, die die Informationen zu den Debian Paketen zwischenspeichert. Es werden, mit Ausnahme von gzip , keine weiteren externen Programme benötigt.

packages

Das packages-Kommando erzeugt einen Paket-Index (Datei Packages) aus dem angegeben Verzeichnis. Dieses wird dazu rekursiv nach Debian Paketen durchsucht. Jedes gefundene Debian Paket wird analysiert, und es wird ein entsprechender Eintrag für die Packages-Datei erzeugt. Die Ausgabe erfolgt auf der Standardausgabe und muss in die gewünschte Datei umgeleitet werden (meist Packages). Dieses Kommando entspricht im Wesentlichen dem Befehl dpkg-scanpackages.

Mit der Option --db kann die Verwendung einer binären Datenbank zur Zwischenspeicherung der Ergebnisse aktiviert werden.

sources

Erzeugt eine Index-Datei für Quellcode-Pakete aus dem angegebenen Verzeichnisbaum. Dabei wird nach Dateien mit der Endung .dsc gesucht, und es wird für jede gefundene Datei ein Eintrag auf der Standardausgabe erzeugt. Dieses Kommando ähnelt dpkg-scansources .

Mit der Option --source-override kann eine andere Override-Datei verwendet werden.

contents

Mit diesem Kommando wird die Datei Contents erzeugt. Das angegebene Verzeichnis wird rekursiv nach Debian Paketen durchsucht, jedes Paket wird analysiert, und die in dem Paket enthaltenen Dateien werden, in sortierter Reihenfolge, in der Contents-Datei vermerkt. Verzeichnisse, die keine Dateien beinhalten, werden nicht in die Auflistung aufgenommen. Ist eine Datei in mehreren Paketen enthalten, so werden diese Pakete - durch Komma getrennt - in der Auflistung aufgeführt.

Mit der Option --db kann die Verwendung einer binären Datenbank zur Zwischenspeicherung der Ergebnisse aktiviert werden.

release

Dieses Kommando dient zum Erzeugen einer Release-Datei. Das angegebene Verzeichnis wird rekursiv nach den Dateien Packages, Packages.gz, Packages.bz2, Sources, Sources.gz, Sources.bz2, Release und md5sum.txt durchsucht. Auf der Standardausgabe wird darauf basierend eine Ausgabe erzeugt, die einen MD5- und SHA1-Extract für jede Datei enthält.

Weitere Felder mit Metadaten können über die Konfigurationsdatei eingefügt werden. Hierzu dient die Variable APT::FTPArchive::Release. Eine erlaubte Angabe kann beispielsweise APT::FTPArchive::Release::Origin sein. Erlaubte Werte sind hier Origin, Label, Suite, Version, Codename, Date, Architectures, Components und Description.

generate

Dieses Kommando wird im Allgemeinen aus einem Cron-Job heraus aufgerufen und erzeugt Index-Dateien aus einer Konfigurationsdatei.

clean

Dieses Kommando löscht alle nicht mehr benötigten Einträge aus den Datenbanken, die in der Konfiguration definiert sind.

apt-ftparchive nutzt eine Konfigurationsdatei, welche mit dem Kommando generate erzeugt werden kann. In dieser Konfigurationsdatei werden die Archive beschrieben, die erzeugt werden sollen. Die Syntax dieser Datei folgt dem üblichen ISC Format, welches

auch von Programmen wie bind oder auch dhcpd bekannt ist. Die Manpage zu apt.conf(5) enthält eine Beschreibung dieser Syntax.

Die erzeugte Konfiguration teilt sich in die vier im Folgenden beschriebenen Abschnitte.

Dir

Dieser Abschnitt beschreibt die Verzeichnisse, in denen nach Dateien gesucht werden soll. Diese Angaben werden mit Verzeichnisangaben aus späteren Abschnitten verknüpft, um einen absoluten Pfad zu jeder Datei zu erzeugen.

ArchiveDir

Dieser Wert gibt das Wurzelverzeichnis des FTP-Archives an. Dies ist normalerweise das Verzeichnis, in dem sich das Unterverzeichnis dists und die Datei ls-LR befinden.

OverrideDir

Gibt an, wo sich die Override-Dateien befinden.

CacheDir

Pfadangabe zu den Cache-Dateien.

FileListDir

Wird die weiter unten beschriebene Option FileList genutzt, so kann hier ein Verzeichnis angegeben werden, in dem Dateilisten abgelegt werden.

Default

Dieser Abschnitt setzt einige Vorgabewerte, diese können in anderen Abschnitten überschrieben werden.

Packages::Compress

Bestimmt den verwendeten Kompressionsalgorythmus, um die Packages-Dateien zu komprimieren. Dieser Wert ist ein String, in dem die einzelnen Werte durch Leerzeichen (Space) getrennt werden. Es sind die Werte gzip, bzip2 und . (für keine Kompression) erlaubt. Wird dieser Wert nicht angegeben, so wird gzip verwendet.

Packages::Extensions

Bestimmt die Dateiendung für Debian Pakete, dies ist normalerweise .deb.

Sources::Compress

Wie Packages::Compress, jedoch für Quellcode Pakete.

Sources::Extensions

Dateiendung für Quell-Codepakete, normalerweise .dsc.

Contents::Compress

Wie bei Packages::Compress, jedoch für das Komprimieren der Packages-Dateien.

DeLinkLimit

Wert in Kilobyte der Dateien, die gelöscht und durch Hard-Links ersetzt werden sollen. Dies ist sinnvoll im Zusammenhang mit Einstellungen für externe Links.

FileMode

Setzt die Zugriffsrechte für alle erzeugten Index-Dateien. Voreingestellt ist der Wert 0644. Diese Rechte werden unabhängig von einem eventuell via umask gesetzten Wert verwendet.

TreeDefault

Diese Variablen wirken sich nur auf die folgenden Abschnitte zum Verzeichnisbaum aus. Alle Variablen werden substituiert, und die Strings $(DIST), $(SECTION) und $(ARCH) werden in die entsprechenden Inhalte umgesetzt.

MaxContentsChange

Bestimmt die maximale Anzahl von Kilobyte der Contents- Dateien, die pro Tag erzeugt werden. Die Contents-Dateien werden rotiert, so dass nach ein paar Tagen alle Dateien neu erzeugt wurden.

ContentsAge

Setzt die maximale Anzahl von Tagen, die eine Contents-Datei überprüft wird, ohne eine Veränderung vorzunehmen. Ist dieses Limit überschritten, so wird die mtime dieser Datei überschrieben.

Es kann vorkommen, dass eine Contents-Datei nicht verändert werden muss, obwohl die Packages-Datei eine Änderung erfahren hat. Dies ist beispielsweise der Fall, wenn das Paket nicht verändert wurde, aber Informationen in der Override-Datei verändert wurden.

Directory

Setzt das Wurzelverzeichnis für den Verzeichnisbaum mit Debian Binär-Paketen. Wird nichts angegeben, so wird $(DIST)/$(SECTION)/binary-$(ARCH)/ verwendet.

SrcDirectory

Setzt das Wurzelverzeichnis für den Verzeichnisbaum mit Debian Quellcode-Paketen. Wird nichts angegeben, so wird $(DIST)/$(SECTION)/source/ verwendet.

Packages

Setzt den Dateinamen der Datei für die Packages-Datei. Voreingestellt ist $(DIST)/$(SECTION)/binary-$(ARCH)/Packages.

Sources

Setzt den Dateinamen der Datei für die Sources-Datei. Voreingestellt ist $(DIST)/$(SECTION)/source/Sources.

InternalPrefix

Setzt den Prefix, der bei einem symbolischen Link dazu führt, dass ein interner statt eines externen Links verwendet wird. Voreingestellt ist $(DIST)/$(SECTION)/.

Contents

Setzt den Dateinamen und Pfad für die erzeugte Contents-Datei. Voreingestellt ist $(DIST)/Contents-$(ARCH). Führt diese Einstellung dazu, dass die Informationen aus verschiedenen Packages-Dateien in einer einzigen Contents-Datei zusammengefasst werden, so wird dies von apt-ftparchive automatisch gehandelt.

Contents::Header

Setzt einen Header, welcher an den Anfang der Contents-Datei geschrieben wird.

BinCacheDB

Setzt eine binäre Datenbank für diesen Abschnitt, diese kann auch von anderen Abschnitten gemeinsam genutzt werden.

FileList

Diese Option gibt an, dass die Liste der Dateien nicht mittels Durchsuchens des Verzeichisbaumes erzeugt werden soll, sondern die Informationen aus einer Datei gelesen werden. Relative Dateinamen werden mit dem Pfad des Archive-Verzeichnisses versehen.

SourceFileList

Diese Option gibt an, dass die Liste der Dateien nicht mittels Durchsuchens des Verzeichisbaumes erzeugt werden soll, sondern die Informationen aus einer Datei gelesen werden. Relative Dateinamen werden mit dem Pfad des Archive-Verzeichnisses versehen. Diese Option wird für Quellcode-Indices verwendet.

Tree

Dieser Abschnitt beschreibt den Verzeichnisbaum, welcher aus einem Basisverzeichnis, darunter liegenden Verzeichnissen für die verschiedenen Abschnitte und darin Verzeichnissen für die Architekturen besteht.

In diesem Abschnitt wird die Variable $(DIST) benutzt, welche das Hauptverzeichnis für den Verzeichnisbaum darstellt. Zusammen mit dem Wert für ArchiveDir ergibt dies üblicherweise so etwas wie dists/sarge.

Alle Werte, die im Abschnitt TreeDefault benutzt werden, können auch in diesem Abschnitt gesetzt werden, hinzu kommen drei neue Variablen.

Sections

Eine mittels Leerzeichen (**SPACE**) separierte Liste von Abschnitten, die unterhalb der Angabe der Distribution aufgeführt werden, meistens ist dies „main“, „contrib“ und „non-free“.

Architectures

Eine mittels Leerzeichen (**SPACE**) separierte Liste von Architekturen, die unterhalb der Angabe des Abschnittes „Sections“ aufgeführt werden, meistens ist dies „i386“, möglich ist aber auch „powerpc“, „sparc“ usw. Der Wert „source“ ist für Quellcode-Distributionen vorgesehen.

BinOverride

Setzt den Pfad und Dateinamen für die Override-Datei für Binär-Pakete. In der Override-Datei sind Informationen zum Abschnitt und der Priorität eines Paketes sowie die Mail-Adresse des Maintainers enthalten.

SrcOverride

Wie „BinOverride“, jedoch für Quellcode-Pakete. Diese Override-Datei enthält Informationen zum Abschnitt.

ExtraOverride

Die Override-Datei Extra-Pakete.

SrcExtraOverride

Die Override-Datei Quellcode Extra-Pakete.

BinDirectory

Dieser Abschnitt beschreibt ein Verzeichnis für Binär-Pakete ohne eine besondere Struktur.

Packages

Pfad und Dateiname der Packages-Datei.

SrcPackages

Pfad und Dateiname der Packages-Datei für Quellcode-Pakete. Es muss einer der beiden Werte für Packages oder SrcPackages gesetzt sein.

Contents

Die Contents-Datei, ein optionaler Wert.

BinOverride

Die binäre Override-Datei.

SrcOverride

Die Quellcode Override-Datei.

ExtraOverride

Override-Datei für Extras.

SrcExtraOverride

Override-Datei für Extra Quellcode-Pakete.

BinCacheDB

Pfad und Dateiname zur Datenbank.

PathPrefix

Pfad zu allen Ausgabedateien.

FileList und SourceFileList

Datei mit Liste der Dateien.

Diese Datei ist voll kompatibel mit dem Programm dpkg-scanpackages. Sie enthält vier Spalten, welche durch Leerzeichen getrennt werden. In der ersten Spalte steht der Paketname, in der zweiten die Priorität, welche für dieses Paket gesetzt werden soll. Die dritte Spalte beschreibt den Abschnitt, in dem das Paket gelistet werden soll, die vierte Spalte nennt den Betreuer des Paketes.

In der vierten Spalte kann, neben einem Namen, welcher dann ersetzt wird, auch eine Liste von Namen aufgeführt werden. Diese Namen müssen durch doppelte Slashes (//) getrennt und von eckigen Klammern umschlossen werden. Wird einer der aufgeführten Einträge gefunden, so wird dieser durch den neuen Wert ersetzt. Die Syntax hierfür lautet:

```
old [// oldn]* => new
```

Diese Datei ist voll kompatibel mit dem Programm dpkg-scanpackages. Der Inhalt dieser Datei besteht aus zwei Spalten, durch Leerzeichen getrennt. Die erste Spalte enthält den Namen des Quellcode-Paketes, die zweite Spalte den Abschnitt, in dem das Paket aufgeführt werden soll.

Diese Datei erlaubt es, neue Tags hinzuzufügen oder bestehende Tags zu verändern. In drei Spalten wird zunächst der Paketname, dann der Tag-Name und zuletzt der Inhalt des Tag aufgeführt.

Alle Optionen, die auf der Kommandozeile übergeben werden, können auch in der Konfigurationsdatei gesetzt werden. Logische Operatoren können dabei in der Form -f-, --no-f, -f=no angeben werden.

--md5

Erzeugt MD5-Checksummen der Pakete. Diese werden in die generierten Index-Dateien aufgenommen. Wird diese Option deaktiviert, so werden keine MD5-Checksummen erzeugt, soweit dies möglich ist. Der entsprechende Eintrag in der Konfigurationsdatei lautet APT::FTPArchive::MD5.

-d, --db

Benutzt eine Cache-Datenbank in einem binären Format zur Verbesserung der Zugriffszeiten. Der entsprechende Eintrag in der Konfigurationsdatei lautet APT::FTPArchive::DB.

-q, --quiet

Erzeugt weniger Ausgaben; es werden zunächst keine Fortschrittsanzeigen ausgegeben. Diese Option kann auch doppelt eingesetzt (oder mit der Syntax -q=2) werden, um den Effekt zu verstärken. Der entsprechende Eintrag in der Konfigurationsdatei lautet quiet.

--delink

Aktiviert das Löschen von Dateien. Deaktiviert wird dieses Verhalten durch die Option --no-delink. Der entsprechende Eintrag in der Konfigurationsdatei lautet APT::FTPArchive::DeLinkAct.

--contents

Mit dieser Option lassen sich Contents-Dateien erzeugen, als Basis werden die Informationen aus der Cache-Datenbank genommen. Der entsprechende Eintrag in der Konfigurationsdatei lautet APT::FTPArchive::Contents.

-s, --source-override

Gibt in Zusammenhang mit dem source-Kommando die Source-Override-Datei an. Der entsprechende Eintrag in der Konfigurationsdatei lautet APT::FTPArchive::SourceOverride.

--readonly

Öffnet die Cache-Datenbank im Nur-lesen-Modus. Der entsprechende Eintrag in der Konfigurationsdatei lautet APT::FTPArchive::ReadOnlyDB.

-h, --help

Gibt eine Hilfe zu apt-ftparchive aus.

-v, --version

Zeigt die Version des Programms an.

-c, --config-file

Gibt den Namen einer Konfigurationsdatei an. Zunächst wird die Standard-Konfigurationsdatei gelesen, danach die mit dieser Option übergebene Konfigurationsdatei. Die Syntax entspricht der Datei apt.conf, siehe apt.conf .

-o, --option

Hiermit kann eine beliebige Option gesetzt werden, die Syntax lautet -o Foo::Bar=bar.

Um nun beispielsweise eine komprimierte Datei Packages.gz für ein Verzeichnis mit einigen Binär-Paketen zu erzeugen, kann apt-ftparchive wie folgt benutzt werden:

```
apt-ftparchive packages directory | gzip > Packages.gz
```

4.23 apt-ftparchive

4.22 apt-move 4.24 apt-show-source

4.24 apt-show-source

4.23 apt-ftparchive 4.25 apt-extracttemplates

Dieses Programm vergleicht die apt-Listen mit Source-Dateien und dem dpkg-Status-File und zeigt auf, von welchen Paketen neuere Versionen als die bereits installierten verfügbar sind. Um festzustellen, welche Version die Sourcen eines Programmes haben, müssen natürlich entsprechende Einträge in der sources.list vorhanden sein. Hierbei ist auch darauf zu achten, welches Release (Woody, Sarge usw.) in den jeweiligen Zeilen angegeben ist.

```
fr@surimi:~$ apt-show-source -p bash Sorry, no newer source package available for bash
```

In diesem Beispiel ist keine aktuellere Version dieses Pakets als Source-Paket verfügbar. Zeigt jedoch die „deb-src“-Zeile auf ein älteres Release, so kann durchaus der Fall eintreten, dass das Source-Paket auf einem älteren Stand ist.

```
fr@surimi:~$ apt-show-source -p bash Inst. Package (Version) | Newest Source Package (Version) ---------------------------------------- ---------------------------- bash (2.05a-12) | bash (2.05a-11)
```

4.24 apt-show-source

4.23 apt-ftparchive 4.25 apt-extracttemplates

4.25 apt-extracttemplates

4.24 apt-show-source 4.26 apt-file

apt-extracttemplates ist ein Werkzeug zum Extrahieren von debconf-Konfigurations- und Template-Informationen aus Debian Paketen. Als Argument können eines oder mehrere Debian Pakete angegeben werden (diese müssen lokal vorliegen), die entsprechenden Konfigurations- und Template-Informationen werden in einem temporären Verzeichnis (/tmp/) abgelegt.

Die Syntax von apt-extracttemplates lautet

```
apt-extracttemplates [-hv] [-t=temporary directory] file...
```

Für jedes Paket wird eine Zeile im Format

```
package version template-file config-script
```

ausgegeben. Hier ein einfaches Beispiel:

```
wasabi:/home/fr# apt-extracttemplates /var/cache/apt/archives/console-data_2002.12.04dbs-42_all.deb  console-data 2002.12.04dbs-42 /tmp/console-data.template.5090 /tmp/console-data.config.5091
```

-t, --tempdir

Hiermit kann das Verzeichnis bestimmt werden, in dem die Dateien abgelegt werden sollen.

-h, --help

Zeigt eine kurze Hilfe zu apt-extracttemplates an.

-v, --version

Zeigt die Programmversion an.

-c, --config-file

Hiermit kann eine zusätzliche Konfigurationsdatei angegeben werden, diese wird vor der eigentlichen Konfigurationsdatei /etc/apt/apt.conf eingelesen.

-o, --option

Dies setzt eine bestimmte Konfigurationsoption. Die Syntax hierzu: -o Foo::Bar=bar .

4.25 apt-extracttemplates

◁ 4.24 apt-show-source △ 4.26 apt-file ▷

4.26 apt-file

4.25 apt-extracttemplates 4.27 apt-show-versions

apt-file sucht nach Dateien in Debian Paketen. Dabei ist es unerheblich, ob die entsprechenden Pakete bereits auf dem System installiert sind oder nicht. apt-file greift dazu auf die Datei Contents-<ARCHITEKTUR>.gz zurück, die von einem der Debian Spiegel geholt wird.

Die Syntax von apt-file lautet:

```
apt-file [ options ] [ action ] [ pattern ]
```

Als Aktionen („actions“) sind dabei vorgesehen:

update

Aktualisiert die Informationen zu den Paketen und Dateien, indem die Datei(en) Contents-<ARCHITEKTUR>.gz von den Servern anhand der Einträge in der Datei /etc/apt/sources.list geholt werden.

search

Sucht nach dem Paket, in dem die Datei vorhanden ist. Es werden alle Pakete aufgelistet, auf die der Suchbegriff zutrifft.

```
fr@wasabi:~$ apt-file search sources.list apt:
usr/share/doc/apt/examples/sources.list apt:
usr/share/doc/apt/examples/sources.list apt:
usr/share/man/es/man5/sources.list.5.gz apt:
usr/share/man/fr/man5/sources.list.5.gz apt:
usr/share/man/fr/man5/sources.list.5.gz apt:
usr/share/man/man5/sources.list.5.gz apt:
usr/share/man/man5/sources.list.5.gz cdd-dev: etc/cdd/sources.list cdd-
```

```
dev: etc/cdd/sources.list.local cdd-dev: etc/cdd/sources.list.stable cdd-
dev: etc/cdd/sources.list.testing cdd-dev: etc/cdd/sources.list.unstable
crosshurd: etc/crosshurd/sources.list/gnu crosshurd:
etc/crosshurd/sources.list/gnu crosshurd:
etc/crosshurd/sources.list/kfreebsd-gnu ...
```

list

Zeigt den Inhalt eines Pakets an, auf den der Suchbegriff zutrifft. Diese Suche ähnelt sehr der Suche mit dpkg -S, jedoch muss hier das Paket nicht auf dem System installiert sein.

purge

Löscht alle vorhandenen Contents-<ARCHITEKTUR>.gz-Dateien aus dem Cache-Verzeichnis (normalerweise /var/cache/apt/).

--cache, -c cache-directory

Setzt das Cache-Verzeichnis auf einen anderen Pfad (normalerweise /var/cache/apt/). So kann apt-file auch ohne Adminstratorrechte eingesetzt werden.

--verbose, -v

Gibt mehr Statusinformationen aus.

--cdrom-mount, -d cdrom-mount-point

Mit dieser Option kann ein alternativer Pfad in dem Dateisystem angegeben werden, an dem die CD eingehängt wurde.

--ignore-case, -i

Ignoriert Groß- und Kleinschreibung.

--regexp, -x

Interpretiert den Suchbegriff als regulären Ausdruck.

--version, -V

Zeigt die Versionsnummer von apt-file an.

--architecture, -a architecture

Setzt die Hardware-Architektur, für die die Suche durchgeführt werden soll. Zu beachten ist dabei, dass auch (mittels update) die entsprechende Contents-<ARCHITEKTUR>.gz-Datei vorhanden sein muss.

--sources-list, -s sources.list

Verwendet eine andere Datei sources.list statt der Datei /etc/apt/sources.list

--package-only, -l

Zeigt nur den Paketnamen, jedoch keine Dateinamen an.

--fixed-string, -F

Sucht nur genau nach der angegebenen Zeichenkette.

--dummy, -y

Führt keine Aktionen aus; eine reine Simulation.

--help, -h

Eine kurze Hilfe zu apt-file.

Weitere Anpassungen können in der Konfigurationsdatei von apt-file (/etc/apt/apt-file.conf) gemacht werden.

```
# Apt-file configuration file  # Substitutions are made as follow: #  host => remote hostname #       port => port #     uri => complete URI from sources.list #     path => path from / #  dist => the distrib name #       comp => the component name #   cache => path to the cache dir #       dest => the destination file name #       cdrom => cdrom mount point  # Where are located Packages (relative to <comp> directory) destination = <host>_<path>_dists_<dist>_Contents-<arch>gz  # Fetch methods http = wget -N -P "<cache> -O "<cache>lt;dest> "<uri>dists/<dist>Contents-<arch>gz" || rm -f "<cache>lt;dest> ftp = wget -N --passive-ftp -P "<cache> -O "<cache>lt;dest> "<uri>dists/<dist>Contents-<arch>gz" || \       rm -f "<cache>lt;dest> ssh = scp -l <user> -P <port|22> "<host>lt;path>dists/<dist>Contents-
```

<arch>gz" "<cache>lt;dest> rsh = rcp -l <user> "<host>lt;path>dists/<dist>Contents-<arch>gz" "<cache>lt;dest> file = cp "/<path>dists/<dist>Contents-<arch>gz" "<cache>lt;dest> copy = cp "/<path>dists/<dist>Contents-<arch>gz" "<cache>lt;dest> cdrom = echo "Put CDROM labeled <path> in the cdrom device" > /dev/stderr ; read ; mount "<cdrom> \ cp "<cdrom>dists/<dist>Contents-<arch>gz" "<cache>lt;dest> umount "<cdrom>

4.26 apt-file

4.25 apt-extracttemplates 4.27 apt-show-versions

4.27 apt-show-versions

4.26 apt-file 4.28 auto-apt

Dieses Kommando zeigt die aktuell installierten Pakete, die Version und das Release an. Mit der Option -p kann nur ein bestimmtes Paket angegeben werden, -r erlaubt die Filterung mittels Wildcard. Die Option -u zeigt alle Pakete, die aktualisiert werden können.

```
fr@surimi:~$ apt-show-versions -h Apt-Show-Versions v.0.02 (c)
Christoph Martin  Usage:  apt-show-versions          shows available
versions of installed packages.   Options:  -stf|--status-file=<file>  Use
<file> as the dpkg status file                       instead of
/var/lib/dpkg/status  -ld|list-dir=<directory>   Use <directory> as path to
apt's                                list files instead of /var/state/apt/lists/
                          or /var/lib/apt/lists/  -p|--package=<package>     Print
versions for <package>.  -r|--regex               Read package with -p as
regex  -u|--upgradeable          Print only upgradeable packages  -a|--
allversions          Print all available versions.  -b|--brief               Short
output.  -v|--verbose             Verbose messages.  -h|--help
Print this help.  fr@surimi:~$ apt-show-versions -r -p -u image*|grep -v
not linux-image-2.6.15-1-486 2.6.15-8 installed: No available version in
archive linux-image-486/testing upgradeable from 2.6.15-8 to 2.6.18+6
linux-image-2.6-486/testing upgradeable from 2.6.15-8 to 2.6.18+6
```

4.27 apt-show-versions

4.26 apt-file 4.28 auto-apt

4.28 auto-apt

4.27 apt-show-versions 4.29 apt-listchanges

auto-apt dient zur Installation von Programmen „bei Bedarf“. Dies kann beispielsweise bei der Software-Entwicklung oder auch schon beim einfachen Übersetzen eines neuen Kernels sinnvoll sein. Mitunter fehlt auf frisch installierten Debian Systemen noch das Paket bin86, das für das Erzeugen eines Kernels benötigt wird. Um automatisch die fehlenden Pakete zu installieren, stellen Sie einfach dem Aufruf von make das Kommando auto-apt voran: auto-apt make bzImage. Werden nun während des Durchlaufs fehlende Programme festgestellt, so ermittelt auto-apt, zu welchem Paket diese gehören, und installiert die fehlenden Pakete inklusive aller Abhängigkeiten.

4.28 auto-apt

4.27 apt-show-versions 4.29 apt-listchanges

4.29 apt-listchanges

4.28 auto-apt 4.30 apt-listbugs

In den meisten Debian Paketen sind so genannte „Changelog“-Dateien enthalten. In diesen dokumentiert der Betreuer eines Pakets die Änderungen zu vorhergehenden Versionen. Ein Blick in diese Dateien nach einem Update ist bei der Fehlersuche hilfreich, wenn Probleme aufgetreten sind. Beispielsweise können in der neuen Version eines Pakets Dateien an einem anderen Ort liegen oder ein Dienst „horcht“ plötzlich auf einem anderen Port.

Da diese Änderungen erst nach der Installation bemerkt werden, kann es zu unerwünschten Unterbrechungen im Betrieb kommen. Abhilfe schafft hier das Paket apt-listchanges. Dieses zeigt vor der Installation eines Pakets die Veränderungen an, die im Changelog dokumentiert sind. Der Administrator hat die Möglichkeit, den Installationsvorgang an dieser Stelle abzubrechen.

```
fr@surimi:~# apt-get install  libmng1  ...  Reading changelogs... Done
libmng (1.0.3-4) unstable; urgency=low    * Build with renamed version
of lcms.   -- Luis Arocha <data@debian.org>  Mon,  3 Jun 2002
18:27:19 +0100  apt-listchanges: Mailing changelogs to fr Selecting
previously deselected package liblcms1-dev. (Reading database ...
115754 files and directories currently installed.)  ...
```

apt-listchanges wird bei der Installation über debconf konfiguriert (siehe auch debconf). Dort kann eine E-Mail-Adresse angegeben werden, an die die Changelog-Informationen gesendet werden. Weiterhin kann bei der Konfiguration gewählt werden, ob nach dem Anzeigen des Changelogs die Installation fortgesetzt oder ob sie (wie in diesem Beispiel) abgebrochen werden soll.

4.29 apt-listchanges

4.28 auto-apt 4.30 apt-listbugs

4.30 apt-listbugs

4.29 apt-listchanges 4.31 apt-config

Dieses Programm zeigt kritische Fehler eines Pakets vor der Installation an. Somit hat der Administrator die Möglichkeit zu entscheiden, ob ein Problem so schwerwiegend ist, dass von der Installation eines Pakets besser Abstand genommen wird.

Während der Installation oder dem Upgrade eines Pakets mit APT wird apt-listbugs automatisch aufgerufen und sucht im Debian Bug Tracking System (http://www.debian.org/Bugs/) nach Fehlermeldungen zu dem zu installierenden Paket. Sind Fehlermeldungen vorhanden, so werden diese angezeigt, und die Installation des Pakets kann abgebrochen werden, falls schwere Probleme zu befürchten sind.

Wird eine stabile Version von Debian verwendet, so ist es sehr unwahrscheinlich, dass bei einem Upgrade eines Pakets schwerwiegende Fehler auftreten. Auf den Einsatz von apt-listbugs kann in solchen Fällen meist verzichtet werden.

Wenn mit einer Entwicklungsversion von Debian experimentiert wird, beispielsweise um die neuesten Programmversionen einzusetzen, kann es durchaus sinnvoll sein, apt-listbugs einzusetzen.

4.30 apt-listbugs

4.29 apt-listchanges 4.31 apt-config

4.31 apt-config

4.30 apt-listbugs 4.32 apt-spy

Mittels apt-config lassen sich der Inhalt der Datei /etc/apt/config sowie die voreingestellten Werte anzeigen. Weiterhin lassen sich auch einzelne Werte mit diesem Tool setzen.

Die Optionen von apt-config lesen sich recht übersichtlich:

```
apt 0.5.4 for linux i386 compiled on Aug 19 2001 01:02:39 Usage: apt-config [options] command  apt-config is a simple tool to read the APT config file  Commands:    shell - Shell mode    dump - Show the configuration  Options:   -h   This help text.   -c=? Read this configuration file   -o=? Set an arbitary configuration option, eg -o dir::cache=/tmp
```

apt-config sollte nicht als Kommando eingesetzt werden; es ist eigentlich als Systemkommando für die Verwendung innerhalb von Installationsskripten vorgesehen.

4.31 apt-config

4.30 apt-listbugs 4.32 apt-spy

4.32 apt-spy

4.31 apt-config 4.33 cron-apt

apt-spy dient zur automatischen Erzeugung von Einträgen für die Datei /etc/apt/sources.list für apt. Hierzu wird aus dem Netz die Datei http://http.us.debian.org/debian/README.mirrors.txt geholt und auf Einträge für Debian Spiegel-Server hin untersucht.

Die Syntax von apt-spy lautet:

```
apt-spy  -d distribution  [ -a area ] [ -c config ] [ -e number ] [ -f file ] [ -i file ]  [ -m mirror-list ] [ -o output-file ] [ -p proxy ] [ -s country-list ] [ -t time ]  [ -u update-URL ] [ -w file ] [ -n number ] [ -h ] [ -v ] [ update ]
```

Zumindest ist also die zu verwendende Distribution, beispielsweise „stable“, „unstable“ oder „testing“, anzugeben. Es werden dann alle (angegebenen) Server hinsichtlich der Bandbreite für Downloads analysiert, und die schnellsten werden in die Datei /etc/apt/sources.list eingetragen. Die ursprünglich an diesem Ort existierende Datei wird als sources.list.bak gesichert.

-d distribution

Gibt die zu verwendende Distribution, „stable“, „unstable“ oder „testing“ an. Diese Option ist immer anzugeben, mit Ausnahme bei der Option update.

-a area

Hiermit kann die Analyse auf bestimmte geografische Regionen eingegrenzt werden, beispielsweise „Africa“, „Asia“, „Europe“, „North-America“ oder auch „Oceania“. Diese Angaben können auch in der Datei /etc/apt-spy.conf abgelegt werden.

-c config

Zu verwendende Konfigurationsdatei statt /etc/apt-spy.conf.

-e number

„early finish“, beendet die Analyse nach der angegebenen Zahl von Servern.

-f file

Die Datei, die für den Test der Bandbreite vom Server geholt werden soll. Normalerweise ist dies die Datei ls-lR im Wurzelverzeichnis des Debian Servers. Wird eine andere Datei angegeben, so muss der Pfad relativ zum Wurzelverzeichnis auf dem Server angegeben werden.

-i file

Die Eingabedatei, aus der apt-spy die Liste der Server liest. Diese Liste sollte mit der Option -w erzeugt werden.

-m mirror-list

Lokale Datei mit den Informationen zu den Debian Spiegeln. Normalerweise wird die Datei /usr/share/apt-spy/README.mirrors.txt genutzt. Wird diese Option im Zusammenspiel mit update genutzt, so wird die Datei an dem gewünschten Ort abgelegt.

-o output-file

Datei, in die die ermittelten Server geschrieben werden, normalerweise ist dies /etc/apt/sources.list.

-p proxy

Mit dieser Option kann ein Proxy-Server benutzt werden. Dieser ist als hostname:port anzugeben.

-s country-list

Eine komma-separierte Liste von Ländern, aus denen die Debian Spiegel geprüft werden sollen. Momantan kann diese Option nicht zusammen mit der Option area verwendet werden.

-t time

Durchschnittliche Zeit, die für den Download-Test verwendet werden soll. Voreingestellt sind 15 Sekunden.

-u update-URL

Adresse, von der die Liste der Debian Spiegel bezogen werden soll.

-w file

Schreibt eine Anzahl von Servern in eine Datei, die mit der Option -i weiterverarbeitet werden kann. Die Anzahl ist auf 5 voreingestellt und kann mit der Option -n angepasst werden.

-n number

Anzahl der Server, die bei Verwendung der Option -w ausgegeben werden soll.

update

Aktualisiert die Liste der Debian Spiegel vom Server http.us.debian.org aus der Datei /usr/share/apt-spy/README.mirrors.txt.

-v

Gibt die Versionsnummer von apt-spy aus.

-h

Zeigt eine kurze Hilfe zu apt-spy an.

```
fr@wasabi:~$ apt-spy -h Usage: apt-spy [options] options: -d distribution Debian distribution (ie, stable). Required unless updating. -a area Area to benchmark. (eg, Europe). -c config Configuration file to use. -e number Number of servers to benchmark before exiting. -f file File to grab when benchmarking. (relative to Debian base). -i file Specify input file. For use with the -w option. -m mirror-list Mirror list to use, or mirror-list to update when updating. -o output-file Where to put output. -p proxy Proxy server to use. In format <server>:[port] -s country_list List of countries to benchmark. Cannot be used with -a. -t time Time to benchmark each server for. An approximate value only. -u update-URL URL to grab mirror-list from when updating. -w file Output top servers (5 by default) to file for use with -i. -n number Specifies number of top servers to output with -w. -v Output a version number. -h Display this message. update Update the mirror list.
```

4.32 apt-spy

4.31 apt-config 4.33 cron-apt

4.33 cron-apt

4.32 apt-spy 4.34 aptitude

cron-apt (http://inguza.com/software/cron-apt/) ist ein Programm, welches mittels eines Cron-Jobs das automatische Aktualisieren von Software-Paketen durch apt-get, aptitude oder apt-file übernehmen kann.

Die zeitliche Steuerung von cron-apt wird über die Datei /etc/cron.d/cron-apt geregelt. Voreingestellt ist dort, dass cron-apt täglich um 04:00 Uhr ausgeführt wird. Ist cron-apt auf einem System installiert, welches nicht 24 Stunden am Tag eingeschaltet ist oder bei dem nicht sichergestellt ist, dass das System zum angegebenen Zeitpunkt eingeschaltet ist, so kann ein symbolischer Link eingerichtet werden, damit das Programm stündlich, täglich, wöchentlich oder monatlich ausgeführt wird. Folgendes Kommando erzeugt einen Link, um cron-apt täglich zu starten.

```
ln -s /usr/sbin/cron-apt /etc/cron.daily/cron-apt
```

Links in die entsprechenden Verzeichnisse führen dazu, dass cron-apt stündlich, wöchentlich oder monatlich ausgeführt wird. Hierbei ist darauf zu achten, dass es nicht gewünscht sein kann, nach jedem Systemstart ein Update der Paketlisten durchzuführen und die zu aktualisierenden Pakete aus dem Netz herunterzuladen. Weiterhin sollte der Eintrag in /etc/cron.d/cron-apt deaktiviert werden.

Die Konfigurationsdateien von cron-apt liegen im Verzeichnis /etc/cron-apt/, die wichtigste dieser Dateien ist die Datei config.

cron-apt nutzt die Informationen in der Datei /etc/apt/sources.list, um auf die Paketquellen zuzugreifen. Es besteht die Möglichkeit, nur Pakete aus bestimmten Quellen aus dem Netz zu holen bzw. zu aktualisieren. Dies kann sinnvoll sein, wenn beispielsweise nur Sicherheitsupdates eingespielt werden sollen. Hierzu ist cron-apt mit einer speziellen sources.list zu starten. Dies kann über den Wert „Options“ in der Konfigurationsdatei geschehen.

```
OPTIONS="-q -o Dir::Etc::SourceList=/etc/apt/security.sources.list"
```

Weiterhin ist es sinnvoll zumindest den Wert für „MAILON“ zu setzen. Dies veranlasst cron-apt, eine Mail an root@localhost zu senden. Diese Variable kann die folgenden Werte annehmen:

```
# Value: error    (send mail on error runs) #       upgrade (when packages is upgraded) #       changes (mail when change in output from an action) #       output  (send mail when output is generated) #       always (always send mail) #                 (else never send mail)
```

4.33 cron-apt

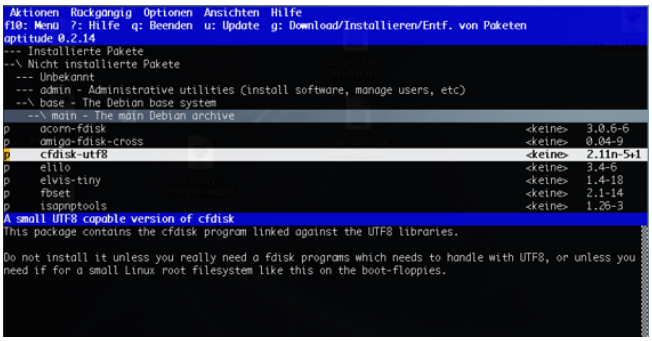
4.32 apt-spy 4.34 aptitude

4.34 aptitude

4.33 cron-apt 4.35 Synaptic

aptitude verfolgt eine etwas andere Philosophie als dselect. Es wird streng nach installierten, nicht installierten, virtuellen Paketen und Paketen, die in einer neueren Version vorliegen, unterschieden. Innerhalb dieser vier Gruppen werden alle Pakete in einer Baumstruktur dargestellt, die auch die Verzeichnisstruktur innerhalb des Debian Archives darstellt, also beispielsweise: main/admin oder non-US/non-free.

Sie können einzelne Teile der Struktur aufklappen und in den einzelnen Bereichen Pakete auswählen.



Bei aptitude decken sich die meisten Tastaturbelegungen mit dselect.

PFEIL-UNTEN

Bewegt den Auswahlbalken zum nächsten Eintrag.

PFEIL-OBEN

Bewegt den Auswahlbalken zum vorherigen Eintrag.

RETURN

Klappt ein Verzeichnis auf/zu.

^

Springt zum Verzeichnis, zu dem das Paket gehört.

+

Markiert ein Paket zur Installation.

-

Markiert ein Paket zum Löschen.

i

Zeigt die Beschreibung des Pakets an.

d

Zeigt die Abhängigkeiten des Pakets an.

v

Zeigt die verfügbaren Versionen des Pakets an.

u

Aktualisiert die Liste der verfügbaren Pakete.

g

Startet die Installation der ausgewählten Pakete.

Die verschiedenen Hintergrundfarben innerhalb des Programms haben folgende Bedeutung:

schwarz

„Normalzustand“ eines Pakets. Beim nächsten Installationsdurchlauf wird dieses Paket nicht verändert. Fett geschriebene Pakete sind bereits installiert.

rot

Das Paket ist in einem unbrauchbaren Zustand oder kann nicht installiert werden.

blau

Das Paket wird mit einer neueren Programmversion aktualisiert.

weiß

Dieses Paket könnte aktualisiert werden, es wurde aber auf dem aktuellen Stand fixiert (hold).

grün

Das Paket wird installiert.

magenta

Das Paket wird gelöscht.

Neben einer Benutzeroberfläche bietet aptitude auch die Möglichkeit, verschiedenste Aktionen auf der Kommandozeile auszuführen. Die Syntax ist dabei wie folgt definiert:

```
aptitude [Optionen] <Aktion> ...
```

Wird beim Aufruf von aptitude auf der Komandozeile keine Aktion angegeben, so startet aptitude im interaktiven Modus.

Aktionen beim Start von aptitude können sein:

install

Installiere/aktualisiere Pakete.

remove

Entferne Pakete.

purge

Entferne Pakete und ihre Konfiguration.

hold

Pakete auf den Status „halten“ setzen.

unhold

Entfernt den Status „halten“ des angegebenen Paketes.

markauto

Pakete als automatisch installiert markieren.

unmarkauto

Pakete als manuell installiert markieren.

update

Neue Listen neuer/aktualisierbarer Pakete laden.

upgrade

Sicheres Upgrade durchführen.

dist-upgrade

Upgrade durchführen, dabei - wenn notwendig - Pakete installieren/entfernen.

search

Pakete nach Namen oder Ausdruck suchen.

show

Gibt detaillierte Informationen zu dem Paket aus.

clean

Heruntergeladene Pakete löschen.

autoclean

Alte heruntergeladene Pakete löschen.

download

Die .deb-Datei eines Pakets herunterladen.

Darüber hinaus versteht aptitude die folgenden Optionen beim Aufruf auf der Kommandozeile. Es ist nicht zwingend notwendig, aptitude mit dem Frontend zu verwenden.

-h

Eine kurze Hilfe zu aptitude.

-s

Aktionen simulieren, aber nicht wirklich durchführen.

-d

Pakete nur herunterladen, nichts installieren oder entfernen.

-P

Zur Bestätigung nachfragen.

-y

Annehmen, dass die Antwort zu einfachen Ja/nein-Fragen „ja“ ist.

-F format

Format für die Suchergebnisse angeben.

-O order

Wie die Suchergebnisse sortiert werden sollen.

-w breite

Die Breite für die Formatierung der Suchergebnisse.

-f

Aggressiv versuchen, kaputte Pakete trotzdem zu installieren.

-V

Anzeigen, welche Version eines Pakets installiert wird.

-D

Abhängigkeiten automatisch veränderter Pakete anzeigen.

-Z

Größenveränderung der einzelnen Pakete anzeigen.

-t release

Release, aus dem Pakete installiert werden sollen.

--with(out)-recommends,

Beeinflusst, ob oder ob nicht Empfehlungen (Vorschläge) wie starke Abhängigkeiten gehandhabt werden.

-S fname

die erweiterten Aptitude-Statusinformationen aus der Datei fname laden.

-u

Neue Paketlisten beim Start laden.

-i

Installationslauf am Anfang durchführen.

Eine sehr nützliche Option von aptitude ist die Möglichkeit, zu einem Paket auch die empfohlenen (recommended) Pakete zu installieren. Hierbei werden die beiden Optionen install und --with[out]-recommends kombiniert.

Beachten Sie, dass dies nur bei noch nicht installierten Paketen funktioniert und nicht bei Paketen, von denen bereits Teile installiert sind.

Hier zunächst ein Beispiel für eine einfache Installation eines Pakets ohne die empfohlenen Pakete:

```
wasabi:~# aptitude install --without-recommends minicom Reading Package Lists... Done Building Dependency Tree       Reading extended state information       Initializing package states... Done Reading task descriptions... Done   The following NEW packages will be installed:   minicom  0 packages upgraded, 1 newly installed, 0 to remove and 0 not upgraded. Need to get 280kB of archives. After unpacking 922kB will be used. Writing extended state information... Done Get:1 ftp://ftp.de.debian.org sarge/main minicom 2.1-7 [280kB] 86% [1 minicom 241920/280kB 86%] ...
```

Es wird ausschließlich das Paket minicom installiert. Dies Verhalten entspricht dem Kommando apt-get install minicom.

```
wasabi:~# apt-get install minicom Reading Package Lists... Done
Building Dependency Tree... Done Recommended packages: lrzsz The
following NEW packages will be installed: minicom 0 upgraded, 1
newly installed, 0 to remove and 0 not upgraded. Need to get 280kB of
archives. After unpacking 922kB of additional disk space will be used.
Get:1 ftp://ftp.de.debian.org sarge/main minicom 2.1-7 [280kB] 88% [1
minicom 247680/280kB 88%] ...
```

Das Paket empfiehlt aber auch das Paket lrzsz, das normalerweise durch aptitude automatisch mitinstalliert wird.

```
wasabi:~# aptitude install minicom Reading Package Lists... Done
Building Dependency Tree       Reading extended state information
Initializing package states... Done Reading task descriptions... Done
The following NEW packages will be automatically installed:   lrzsz
The following NEW packages will be installed:   lrzsz minicom  0
packages upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 373kB of archives. After unpacking 1111kB will be used.
Do you want to continue? [Y/n/?]  ...
```

4.34 aptitude

4.33 cron-apt 4.35 Synaptic

4.35 Synaptic

4.34 aptitude 4.36 PackageKit

Synaptic ist ein weiteres grafisches Werkzeug zur Installation von Debian Paketen. Entwickelt wurde Synaptic zunächst von der Firma Conectiva, mittlerweile wird die Entwicklung vom Debian Team vorgenommen, und Synaptic ist in die Distribution integriert.

Synaptic sollte zunächst auf dem System mittels apt-get install synaptic installiert werden. Dies sollte für die nächste Zeit der letzte Aufruf von apt-get gewesen sein... Synaptic benötigt, wie auch schon apt-get, Superuser-Rechte, um Pakete installieren zu können.

Alternativ zum Aufruf aus der Kommandozeile kann Synaptic auch aus dem GNOME-Menü heraus gestartet werden. Da der Login unter GNOME auf einem Debian System nicht so ohne Weiteres als Administrator („root“) erfolgen kann, muss der Menüeintrag so konfiguriert sein, dass Synaptic via gksu gestartet wird (gksu benötigt im Gegensatz zu gnomesu keine GNOME-Bibliotheken). Das Debian Paket von Synaptik ist bereits entsprechend vorbereitet.

Synaptic benötigt nach dem Aufruf einige Zeit zum Start (die Dauer hängt unter anderem von der verwendeten Hardware und der Anzahl der verfügbaren Pakete ab). Mit APT vertraute Benutzer werden sich in Synpatic schnell zurechtfinden; für fast jede APT-Aktion gibt es eine entsprechende Funktion in Synaptic. Im Unterschied zu APT ist Synaptic allerdings in der Lage, die gewünschten Aktionen zu queuen und erst auszuführen, wenn der Administrator mit dem gewünschten Gesamtergebnis zufrieden ist. APT dagegen würde jede Aktion unmittelbar sofort ausführen.

Zunächst zeigt Synaptic in der Übersicht alle Paketgruppen an. Mit einem Mausklick können die einzelnen Gruppen aufgeklappt werden, und es können einzelne Pakete ausgewählt werden, beispielsweise um das gewünschte Paket neu zu installieren. Im hier gezeigten Beispiel wird das Paket bash aktualisiert. Das Fenster von Synaptic teilt sich in verschiedene Bereiche: Unter der Menüleiste, über die sich alle verfügbaren Aktionen aufrufen lassen, befindet sich eine Icon-Leiste mit den am häufigsten verwendeten Aktionen.

Direkt darunter können Sie Filter definieren und Suchbegriffe eingeben. Im zweiten Drittel des Fensters sehen Sie die Übersicht der Paketgruppen, die, wie beschrieben, die einzelnen Pakete beherbergen.

Im unteren, linken Bereich befinden sich verschiedene Reiter, über die Sie weitere Informationen zu einzelnen Paketen abrufen können. Ist noch kein Paket ausgewählt, so lassen sich diese Reiter nicht aktivieren und erscheinen grau. Neben den allgemeinen Informationen wie Paketversion und möglichen aktuelleren Versionen sowie der detaillierten Paketbeschreibung im zweiten Reiter sind dies noch die Abhängigkeiten dieses Pakets sowie, im letzten Reiter, alle Dateien, die in diesem Paket enthalten sind.

Rechts davon können Aktionen wie „Install“, „Keep“, „Remove“ oder auch „Hold“ auf jedes Paket angewendet werden. Ebenso können Pakete hier neu konfiguriert werden („Reconfigure“).

Um einen schnellen Zugriff auf die Paketinformationen zu haben, verwaltet das Debian Paketmanagement eine interne Liste auf dem System. Diese Liste muss von Zeit zu Zeit mit einer Quelle für Debian Pakete abgeglichen werden. Dies wird in den meisten Fällen ein Debian FTP-Server sein.

Um die Paketliste zu aktualisieren, verwenden Sie die Schaltfläche „Update List“.

Die Paketübersicht kann in verschiedenen Formen angezeigt werden. Diese können über das Menü „View“ ausgewählt werden. Die beiden ersten Menüpunkte, „Expand All“ und „Collapse All“, dienen dazu, die „Äste“ der baumförmigen Paketübersicht aus- und einzuklappen. Die Menüeinträge „Section Tree“ (Baumansicht mit Sektionen der Pakete), „Alphabetic Tree“ (nach Alphabet sortierte Übersicht) und „Status Tree“ (nach Paketstatus sortierte Liste) dienen dazu, die Paketliste nach verschiedenen Kriterien anzuzeigen, und erlauben so, je nach Anwendungsfall, eine schnelle Auswahl der Pakete.

Synaptic kann Pakete auf unterschiedliche Weisen suchen, alle Verfahren basieren aber auf der Paketübersicht in der Mitte des Fensters. Jede Suche muss man sich als Filter über diese Paketübersicht vorstellen. Ein erster grober Filter kann über die Auswahl im Pull-Down-Menü „Show“ angewendet werden. Hier können beispielsweise alle Pakete („All Packages“), installierte Pakete („Installed“) oder zu aktualisierende („Upgradable“) Pakete ausgewählt werden.

Unabhängig von der getroffenen Auswahl erscheint die Paketübersicht immer im gleichen Format. Der Filter beeinflusst dabei lediglich, welche Pakete angezeigt werden.

Wirklich interessant und somit auch erwähnenswert ist das Eingabefeld für die Suche nach Paketnamen. Dieses Feld verhält sich etwas ungewöhnlich, weil es sofort, nach

jedem eingegebenen Zeichen, die Suche beginnt. Es ist nicht notwendig, die **RETURN**-Taste zu betätigen. Eine solche Suchfunktion benötigt leider etwas Prozessorleistung, so dass von der Benutzung auf sehr schwachbrüstigen Systemen abgeraten wird.

Um Pakete zu installieren oder vom System zu löschen, reicht es aus, das gewünschte Paket auszuwählen und die Schaltfläche „Install“ bzw. „Remove“ anzuwählen. Synaptic merkt sich diese Auswahl und wird die gewünschten Pakete für die vorgesehene Aktion vormerken.

Die Aktion (installieren oder löschen) wird erst nach dem Anklicken der Schaltfläche „Proceed“ ausgeführt. Synaptic wird dann die notwendigen Abhängigkeiten zwischen den Paketen auflösen und ggf. weitere Pakete installieren oder auch löschen. Sollten dabei Probleme auftauchen, so wird der Administrator um eine Bestätigung gebeten.

Einige der zu installierenden Pakete müssen unter Umständen zunächst konfiguriert werden; auch diese Fragen leitet Synaptic an den Administrator weiter.

Um alle bereits installierten Pakete zu aktualisieren, von denen in einer bekannten Quelle neuere Versionen vorliegen, können die beiden Schaltflächen „Upgrade All“ und „Dist Upgrade“ benutzt werden. „Upgrade All“ aktualisiert alle Pakete, von denen eine neuere Version vorliegt, die aber keine zusätzlichen Pakete benötigen. Mitunter wurden aber Pakete vom Maintainer in kleinere Pakete aufgeteilt. Dies ist insbesondere beim Wechsel von einer Debian Version (Woody) zur nächsten (Sarge) möglich. Um auch diese Pakete zu aktualisieren, wählen Sie den Button „Dist Upgrade“.

In jedem Fall müssen Sie aber vorher mit „Update List“ die Liste der verfügbaren Debian Pakete aktualisieren.

Debian benutzt die Informationen in der Datei /etc/apt/sources.list, um verschiedene Quellen mit Debian Paketen zu verwalten. Über Änderungen in dieser Datei können sowohl die Server, von denen die Pakete bezogen werden sollen, als auch die gewünschten Releases („stable“, „unstable“, „testing“) und die Bereiche („main“, „contrib“, „non-free“) verwaltet werden.

Normalerweise werden Anpassungen an dieser Datei vom Administrator mit einem Texteditor vorgenommen. Synaptic benutzt hierzu ein handliches Werkzeug, das über das Menü „Setting“/„Repository“ zu erreichen ist.

Jeder Eintrag in der Datei kann über die entsprechenden Felder geändert werden; so werden Syntaxfehler vermieden. Weiterhin können einzelne Einträge in der Datei deaktiviert werden. Über die Schaltfläche „Vendors“ können Hersteller von Paketen

definiert werden. Diese Angaben landen in der Datei /etc/apt/vendors.list und werden über einen (bisher optionalen Eintrag) in der Datei /etc/apt/sources.list referenziert.

Wichtig ist es insbesondere, neben einem Kürzel und dem Namen des Herstellers den kryptographischen Fingerprint anzugeben. Diese Funktion wird noch nicht ausgiebig genutzt, wird es aber zukünftig erlauben, die Authentizität von Paketen zu überprüfen.

Bisher wurden alle Aktionen lediglich vorgemerkt, aber nicht ausgeführt. Um die gewünschten Pakete zu (de-)installieren, müssen Sie auf die Schaltfläche „Proceed!“ klicken. Synaptic wird dann alle Aktionen ausführen.

4.35 Synaptic

4.34 aptitude 4.36 PackageKit

4.36 PackageKit

4.35 Synaptic 4.37 dpkg

PackageKit (packagekit.org/) ist ein System verschiedener Programme, welches entwickelt wurde um die Installation und Aktualisierung von Software auf Ihrem Computer zu erleichtern - unabhängig von der verwendeten Linux-Distribution. Primäres Ziel ist es, die zur Installation von Software verwendeten grafischen Werkzeuge der verschiedenen Distributionen zu vereinheitlichen. PackageKit setzt dabei auf die neuesten Technologien wie beispielsweise PolicyKit.

Die tatsächliche Distributions Werkzeuge zur Installation von Paketen (also Beispielsweise yum, apt, Conary, etc) werden von PackageKit eingesetzt und können parallel weiterhin benutzt werden. PackageKit ist nicht dazu gedacht, diese Tools zu ersetzen, vielmehr soll die Installation und das Paketmanement auf allen Distributionen vereinheitlicht werden.

PackageKit selbst besteht aus einem Daemon - genannt packagekitd. packagekitd wird erst gestartet wenn ein Programm diesen Daemon benötigt. Das bedeutet auch, dass packagekitd nach dem Einsatz wieder beendet wird.

gnome-packagekit ist eine Sammlung von grafischen Werkzeugen die PackageKit in den GNOME-Desktop integrieren. Apper ist der Name der KDE Tools für PackageKit.

Standardmäßig verwendet PackageKit PolicyKit für die Benutzerauthentifizierung. Dies bedeutet, dass Sie als Administrator mit einer feinkörnigen Kontrolle angeben, was Ihre Benutzer tun können und was nicht.

4.36 PackageKit

4.35 Synaptic 4.37 dpkg

4.37 dpkg

4.36 PackageKit 4.38 dpkg-reconfigure

dpkg (http://wiki.debian.org/dpkg) ist die Basis der Debian Paketverwaltung. dpkg steht für „Debian GNU/Linux package manager“. Da dpkg mittlerweile auch von anderen Distributionen wie Novell/SuSE und RedHat/Fedora eingesetzt wird, ist die Bezeichnung nicht mehr ganz korrekt.

Die wichtigste Funktion von dpkg besteht darin, das System in einem stabilen Zustand zu halten. Da die Werkzeuge zur Paketverwaltung in anderen Disrtibutionen nicht den Anforderungen der Debian Entwickler genügten, wurde dpkg entworfen. Das Paketformat RPM, das u.a. von RedHat/Fedora- und Novell/SuSE-Distributionen verwendet wird, prüft lediglich, ob eine benötigte Datei vorhanden ist. Ob diese Datei in der gewünschten oder gar benötigten Version vorliegt, wird nicht geprüft.

Mit diesem Programm kommen Sie nur in seltenen Fällen in Berührung; meist werden Sie Frontends wie apt-get oder aptitude benutzen. Hier folgen trotzdem einige nützliche Beispiele, wie dpkg eingesetzt werden kann.

Natürlich bringt auch dpkg, wie alle guten Unix-Werkzeuge, eine eingebaute Hilfefunktion mit. Mit der Option --help kann man sich einen ersten Überblick über die beachtliche Vielzahl von Optionen verschaffen.

```
Usage:  dpkg -i|--install      <deb file name> ... | -R|--recursive <dir> ...
dpkg --unpack          <deb file name> ... | -R|--recursive <dir> ...  dpkg -
A|--record-avail <deb file name> ... | -R|--recursive <dir> ...  dpkg --
configure              <package name> ... | -a|--pending  dpkg -r|--remove |
-P|--purge <package name> ... | -a|--pending  dpkg --get-selections
[<pattern> ...]  get list of selections to stdout  dpkg --set-selections
set package selections from stdin  dpkg --update-avail <Packages-file>
replace available packages info  dpkg --merge-avail <Packages-file>
merge with info from file  ...    --no-force-...|--refuse-...    Stop when
problems encountered  --abort-after <n>              Abort after
encountering <n> errors  Comparison operators for --compare-versions
are:  lt le eq ne ge gt       (treat no version as earlier than any version);
lt-nl le-nl ge-nl gt-nl (treat no version as later than any version);  < <<
<= = => >> >       (only for compatibility with control file syntax).  Use
`dselect' for user-friendly package management.
```

Die meisten Optionen haben eine kurze und eine lange Schreibweise, zum Beispiel -i oder alternativ --install; der Einfachheit halber wird im Folgenden immer die Kurzform erwähnt. Einige wenige Optionen sind ausschließlich in der ausführlichen Version verfügbar.

Diese Option zeigt den Inhalt eines Debian Paketes an, hierzu muss das Paket nicht zwingend installiert sein. Als weiterer Parameter ist der Dateiname des Debian Paketes anzugeben.

Mit dieser Option können Pakete installiert werden, die lokal im Dateisystem, also auch auf einer gemounteten CD-ROM, vorliegen. Hierzu ist neben der Option -i lediglich der Dateiname (ggf. inklusive Pfad) anzugeben. Wenn eine Reihe Pakete installiert werden soll, so können Sie auch ein Verzeichnis statt eines Dateinamens angeben; zusätzlich ist die Option -R anzugeben. Mit dieser wird das Verzeichnis rekursiv nach Debian Paketen durchsucht.

Mit --pending können bereits entpackte, aber noch nicht konfigurierte Pakete endgültig konfiguriert werden. Es kann zu einem solchen Zustand kommen, wenn das System bei der Installation von Paketen auf einen Fehler trifft und abbricht. Zu diesem Zeitpunkt können aber bereits andere (eventuell wichtige) Pakete im aktuellen Installationsdurchlauf entpackt worden sein. Diese können durch Angabe des Paketnamens gezielt konfiguriert und damit abschließend installiert werden.

Sollen nicht nur bestimmte, sondern alle bereits entpackten Pakete dieser Prozedur unterzogen werden, so müssen Sie die Option -a oder --pending verwenden.

Mit der Option -r, für „remove“, können Pakete entfernt werden. Dazu geben Sie lediglich den Namen des Pakets oder die Namen der Pakete (durch Leerzeichen getrennt) an, das/die gelöscht werden soll(en). Diese Option löscht lediglich die Programmdateien eines Pakets. Die eigentlichen Konfigurationsdateien (die nur sehr wenig Platz auf dem Datenträger einnehmen) werden nicht gelöscht. Wird ein Paket einige Zeit später erneut installiert, so liegt die alte Konfiguration noch vor und kann weiterhin genutzt werden.

Die Option „purge“ entfernt ein Paket komplett, inklusive aller Konfigurationsdateien, aus dem System.

Mit der Option -l können die tatsächlichen Paketnamen ermittelt werden. Diese weichen häufig von den eigentlichen Befehlen ab.

```
surimi:~$ dpkg -l mozilla*
Gewünscht=Unbekannt/Installieren/R=Entfernen/P=Säubern/Halten |
Status=Nicht/Installiert/Config/U=Entpackt/Fehlgeschl. Konf./Halb
install. |/Fehler?=(keiner)/Halten/R=Neuinst.notw/X=beides
(Status,Fehler: GROSS=schlecht) ||/ Name           Version
Beschreibung +++-==============-==============-
============================================== ii
mozilla        1.0.0-1        Mozilla Web Browser - dummy package ii
mozilla-browse 1.0.0-1        Mozilla Web Browser - core and browser rc
mozilla-browse 0.0.20020226.1 An Open Source WWW browser for X
and GTK+ (C ii  mozilla-browse 0.0.20020610.0 An Open Source
WWW browser for X and GTK+ (C ii  mozilla-chatzi 1.0.0-1
Mozilla Web Browser - irc client pn  mozilla-chatzi <keine>       (keine
Beschreibung vorhanden) pn  mozilla-chatzi <keine>       (keine
Beschreibung vorhanden) pn  mozilla-cvs    <keine>       (keine
Beschreibung vorhanden) pn  mozilla-cvs-de <keine>       (keine
Beschreibung vorhanden) ii  mozilla-dev    1.0.0-1       Mozilla Web
Browser - development files un  mozilla-dmotif <keine>       (keine
Beschreibung vorhanden) ii  mozilla-dom-in 1.0.0-1       A tool for
inspecting the DOM of pages in Mo ii  mozilla-dom-in 0.0.20020610.0
A tool for inspecting the DOM of pages in Mo ii  mozilla-js-deb 1.0.0-1
JavaScript debugger for use with Mozilla pn  mozilla-js-deb <keine>
(keine Beschreibung vorhanden) rc  mozilla-locale 0.9.9-3       Mozilla
German Language/Region Package. pn  mozilla-locale <keine>
(keine Beschreibung vorhanden) pn  mozilla-locale <keine>       (keine
Beschreibung vorhanden) ii  mozilla-mailne 1.0.0-1       Mozilla Web
Browser - mail and news support pn  mozilla-mailne <keine>
(keine Beschreibung vorhanden) pn  mozilla-mailne <keine>       (keine
Beschreibung vorhanden) ii  mozilla-psm    1.0.0-1       Mozilla Web
Browser - Personal Security Mana pn  mozilla-psm-cv <keine>
(keine Beschreibung vorhanden) ii  mozilla-psm-sn 0.0.20020610.0
PSM - Personal Security Manager for Mozilla  un  mozilla-smotif
<keine>       (keine Beschreibung vorhanden) pn  mozilla-snapsh
<keine>       (keine Beschreibung vorhanden) pn  mozilla-snapsh
<keine>       (keine Beschreibung vorhanden) ii  mozilla-xmlter 1.0.0-1
Mozilla Web Browser - XML enabled pn  mozilla-xmlter <keine>
```

(keine Beschreibung vorhanden) pn mozilla-xmlter <keine> (keine Beschreibung vorhanden)

Problematisch ist hierbei, dass der Name des Pakets abgeschnitten wird; dies können Sie durch Setzen der Variablen COLUMNS ändern: COLUMNS=132 dpkg -l mozilla*.

Hiermit kann der Status eines Pakets ermittelt werden. Es werden u.a. Informationen zum Status der Installation, der Version, der Größe, der Section und der Abhängigkeiten ausgegeben. So können sehr schnell die Ursachen für eventuelle Probleme bei der Installation von weiteren Paketen ermittelt werden.

Package: bash Essential: yes Status: install ok installed Priority: required Section: base Installed-Size: 1288 Maintainer: Matthias Klose <doko@debian.org> Source: bash (2.05b-2-8.1) Version: 2.05b-8.1 Replaces: bash-doc (< 2.05-1), bash-completion Depends: base-files (> 2.1.12) Pre-Depends: libc6 (> 2.3.1-1), libncurses5 (> 5.3.20021109-1) Suggests: bash-doc Conflicts: bash-completion Conffiles: /etc/skel/.bashrc c5f1155761187900cbb3b6554c2b2533 /etc/skel/.bash_profile ee190fd94cb7bfe8a663447386c065e8 /etc/bash.bashrc e218a2979b01db4e9c3ae19c94294a57 /etc/bash_completion 14dde46ca4fb4af4d1f22f0f0fc2ef8c Description: The GNU Bourne Again SHell Bash is an sh-compatible command language interpreter that executes commands read from the standard input or from a file. Bash also incorporates useful features from the Korn and C shells (ksh and csh). . Bash is ultimately intended to be a conformant implementation of the IEEE POSIX Shell and Tools specification (IEEE Working Group 1003.2). . Included in the bash package is the Programmable Completion Code, by Ian Macdonald. . Statically linked.

Für den fortgeschrittenen Systemadministrator ist an dieser Stelle interessant, dass hier auch die MD5-Checksummen der Konfigurationsdateien ausgegeben werden.

Ermitteln von Checksummen

Um die Checksumme eines gesamten Pakets zu ermitteln, benutzt der Administrator das Programm debsums(siehe auch debsums).

```
fr@sushi:~$ debsums ssh    usr/bin/ssh                    OK usr/bin/scp                    OK
usr/bin/ssh-add                OK usr/bin/ssh-agent              OK usr/bin/ssh-keygen
OK usr/bin/ssh-keyscan           OK usr/bin/sftp                OK usr/bin/ssh-copy-id
OK usr/bin/ssh-argv0             OK usr/sbin/sshd               OK usr/lib/ssh-keysign
OK usr/lib/sftp-server           OK usr/share/man/man1/scp.1.gz        OK
usr/share/man/man1/ssh-agent.1.gz   OK usr/share/man/man1/ssh-keygen.1.gz  OK ...
```

Mit der Option -S kann ermittelt werden, zu welchem Paket eine Datei gehört. Dies trifft lediglich auf bereits installierte Pakete zu. Eine Suche in nicht installierten Paketen ist auf diesem Wege nicht möglich.

```
fr@surimi:~$ dpkg -S gdm.conf  gdm: /etc/gdm/factory-gdm.conf gdm:
/etc/gdm/gdm.conf  fr@surimi:~$ dpkg -S /etc/gdm/gdm.conf  gdm:
/etc/gdm/gdm.conf
```

Hierbei kann das Weglassen oder Hinzufügen des Pfades zu der gesuchten Datei zu unterschiedlichen Ergebnissen führen.

Durchsucht das System nach unvollständig installierten Paketen und versucht, sinnvolle Vorschläge zu unterbreiten, welche Schritte notwendig sind, um diese Pakete in einen funktionsfähigen Zustand zu versetzen.

Zeigt alle Dateien an, die zum angegebenen Paket gehören.

```
$ dpkg -L bash /. /bin /bin/bash /etc /etc/skel /etc/skel/.bashrc
/etc/skel/.bash_profile /etc/bash_completion.d /etc/bash.bashrc
/etc/bash_completion /usr /usr/share /usr/share/doc /usr/share/doc/bash
/usr/share/doc/bash/completion-contrib /usr/share/doc/bash/completion-
contrib/gnatmake /usr/share/doc/bash/completion-contrib/bitkeeper  ...
/usr/share/man/man7 /usr/share/man/man7/bash-builtins.7.gz /usr/bin
/usr/bin/bashbug /bin/rbash /bin/sh /usr/share/man/man1/sh.1.gz
```

Gibt, ohne weitere Parameter, eine Liste aller installierten Pakete aus. Wird zusätzlich ein „*“ angegeben, so werden alle Pakete mit dem aktuellen Zustand ausgegeben. Dies zeigt also auch zu löschende und nicht installierte Pakete an.

Diese Ausgabe kann in eine Datei geschrieben werden und mittels der Option `--set-selections` (beispielsweise auf einem anderen System oder aus einem Backup) wieder eingelesen werden. Somit können einzelne Systeme sehr leicht dupliziert werden. Auch kann auf diesem Wege ein System nach einem Hardware-Ausfall wiederhergestellt werden.

Beispiel:

```
dpkg --get-selections \* > /tmp/PAKETLISTE
```

Liest eine zuvor mit --get-selections erzeugte Paketliste wieder ein.

Beispiel:

```
dpkg --set-selections < /tmp/PAKETLISTE
```

Abhängig von den Informationen in der Datei PAKETLISTE werden so alle Pakete aus dieser Datei in der Paketdatenbank mit einem Status gekennzeichnet. Ein darauf folgendes apt-get dselect-upgrade wird die gewünschten Aktionen durchführen.

Aktualisiert die Informationen über verfügbare Pakete. Mit der Aktion --merge-avail werden die bestehenden Informationen mit den Informationen aus der angegebenen Packages-Datei zusammengeführt. Mit der Aktion --update-avail wird die alte Information durch die Information aus der Packages-Datei ersetzt. dpkg speichert alle Aufzeichnungen über die verfügbaren Pakete in /var/lib/dpkg/available.

Ein einfacheres Kommando, um die available-Datei in einem Aufruf abzurufen und zu aktualisieren, ist dselect update.

Sollte einmal aus Versehen eine der Konfigurationsdateien verschwunden sein oder eine andere wichtige Datei fehlen, so kann ein komplettes Paket einfach erneut installiert werden. Hierzu müssen Sie sich zunächst das betroffene Paket besorgen, beispielsweise mit apt-get -d install paket. Nun kann das Paket installiert werden. Die Option --force-confnew überschreibt dabei alle Konfigurationsdateien:

```
dpkg -i --force-confnew /var/cache/apt/archives/paket.deb
```

Diese Option kann zusammen mit jeder anderen Option von dpkg eingesetzt werden, führt jedoch im Allgemeinen zu einem Versionschaos. Hiermit werden alle Abhängigkeiten eines Pakets ignoriert und die gewünschte Aktion (beispielsweise „install“) ausgeführt. Alle Warnungen des Paketmanagements werden ignoriert.

Wird dpkg mit dieser Option eingesetzt, so ist es kaum mehr möglich, danach apt-get sinnvoll zu benutzen, da apt-get feststellen wird, dass bestimmte Abhängigkeiten nicht erfüllt werden können. In einem solchen Fall können Sie versuchen, mit apt-get -f install die Situation zu bereinigen. Dies wird aber das zuvor mit --force-depends behandelte Paket in einen unbrauchbaren Zustand versetzen bzw. löschen.

Pakete können im Debian Paketmanagement unterschiedliche Zustände haben; einer davon ist „hold“. Pakete mit diesem Zustand werden bei einem Update nicht aktualisiert. Um eine neue Version dieses Pakets zu installieren, muss das Paket gezielt installiert werden. Ein einzelnes Paket kann mit dem Kommando

```
echo <paketname> "hold" | dpkg --set-selections
```

auf „hold“ gesetzt werden.

Sollen mehrere Pakete auf einmal in den Zustand „hold“ gesetzt werden, so kann dies wie folgt geschehen:

```
dpkg --set-selections << EOF <paketname1> hold <paketname2> hold <paketname3> hold <paketname4> hold EOF
```

Natürlich kann der Status „hold“ auch wieder zurückgesetzt werden, dies geschieht wie folgt:

```
echo <paketname> "install" | dpkg --set-selections
```

Die Abfrage der Paketdatenbank nach allen Paketen die aktuell den Status „hold“ besitzen ist ebenso einfach:

```
dpkg --get-selections | grep hold
```

„hold“ mit Aptitude

dpkg und Aptitude nutzen unterschiedliche Informationen zu den Paketen. Dies führt dazu, dass ein mit Aptitude auf „hold“ gesetztes Paket mit apt-get upgrade dennoch aktualisiert wird. Werden also Pakete mittels Aptitude auf „hold“ gesetzt, so müssen auch Funktionen wie „upgrade“ und „dist-upgrade“ mittels Aptitude durchgeführt werden.

4.37 dpkg

4.38 dpkg-reconfigure

4.37 dpkg 4.39 dpkg-preconfigure

Mit diesem Kommando können Pakete neu konfiguriert werden, die „debconf“ (siehe debconf) zur Verwaltung der Konfigurationsdateien nutzen (dies sollte mittlerweile für den Großteil der Pakete der Fall sein). Hierzu ist neben dem Aufruf des Programmes nur noch der Paketname anzugeben.

Ein dpkg-reconfigure debconf konfiguriert debconf selbst neu. Hierbei kann zwischen verschiedenen interaktiven und vollautomatischen Modi gewählt werden.

Welche bereits installierten Pakete sich mit diesem Kommando neu konfigurieren lassen, ermittelt das Programm configure-debian .

4.38 dpkg-reconfigure

4.37 dpkg 4.39 dpkg-preconfigure

4.39 dpkg-preconfigure

4.38 dpkg-reconfigure 4.40 dpkg-scanpackages

dpkg-preconfigure kann Pakete vor der Installation auf einem System konfigurieren. dpkg-preconfigure konfiguriert eine Anzahl von Paketen und führt dazu die debconf-Skripte dieser Pakete aus. dpkg-preconfigure kennt dabei zwei Möglichkeiten der Parameterübergabe:

```
dpkg-preconfigure [options] package.deb  dpkg-preconfigure --apt
```

-ftype, --frontend=type

Setzt das zu verwendende debconf-Frontend.

-pvalue, priority=value

Gibt die minimale Priorität der debconf-Fragen an, die noch angezeigt werden sollen.

--apt

Aktiviert den „APT“-Modus. Dabei wird eine Liste von Paketen auf der Standard-Eingabe gelesen, statt diese auf der Kommandozeile zu übergeben. Normalerweise wird dies genutzt, um dpkg-preconfigure auf alle Pakete anzuwenden, bevor diese installiert werden. Hierzu ist Folgendes der Datei /etc/apt/apt.conf hinzuzufügen, falls es nicht schon vorhanden ist:

```
// Pre-configure all packages before // they are installed. DPkg::Pre-
Install-Pkgs {        "dpkg-preconfigure --apt --priority=low"; };
```

-h, --help

Zeigt eine Hilfe zum Programm dpkg-preconfigure an.

4.39 dpkg-preconfigure

4.38 dpkg-reconfigure 4.40 dpkg-scanpackages

4.40 dpkg-scanpackages

4.39 dpkg-preconfigure 4.41 dpkg-scansources

Pakete, die aus dem Internet geholt wurden und in irgendeinem Verzeichnis auf der Festplatte schlummern, lassen sich mit den bisher beschriebenen Methoden nur mühsam installieren. Mit dpkg lassen sich sehr einfach einzelne Pakete installieren; Abhängigkeiten werden jedoch nicht berücksichtigt. Eleganter wäre es, auch diese Pakete in die sources.list einzubinden und APT zu benutzen. Doch leider scheitert dies an der fehlenden Datei Packages.gz und einer entsprechenden Verzeichnisstruktur, in der die Pakete abgelegt sind. Die Datei Packages.gz kann mit dem Programm dpkg-scanpackages erzeugt werden.

Die Datei Packages.gz enthält Informationen darüber, welche Pakete sich in diesem Verzeichnis befinden. Mit dem Kommando

```
dpkg-scanpackages ./ /dev/null | gzip > Packages.gz
```

das direkt in dem Verzeichnis aufgerufen wird, in dem sich die Pakete befinden, lässt sich diese Datei erzeugen. Wenn weitere oder aktualisierte Pakete in diesem Verzeichnis abgelegt werden, muss das Kommando natürlich erneut aufgerufen werden.

Wenn die Pakete auf dem System im Verzeichnis /home/ftp/meinepakete/ abgelegt sind, lautet der Eintrag in der Datei /etc/apt/sources.list:

```
deb file:/home/ftp/meinepakete ./
```

Der Pfad ist den lokalen Gegebenheiten anzupassen.

Weitere Informationen zum Erstellen von Debian Repositories finden Sie in Debian Repositories.

4.40 dpkg-scanpackages

4.39 dpkg-preconfigure 4.41 dpkg-scansources

4.41 dpkg-scansources

4.40 dpkg-scanpackages 4.42 dpkg-checkbuilddeps

Natürlich ist es nicht nur möglich, für Binär-Pakete entsprechende Package-Dateien zu erzeugen, dies kann auch für die Source-Pakete geschehen.

Es werden zunächst die drei zum Binär-Paket gehörenden Sourcedateien mit den Endungen .orig.tar.gz, .dsc und .diff.gz benötigt. Diese werden in einem eigenen Verzeichnis abgelegt. Die Programme zur Verwaltung von Debian Paketen greifen bei Source-Paketen auf die Datei Sources.gz zu. Diese kann mit dem Kommando

```
dpkg-scansources ./ | gzip > Sources.gz
```

erzeugt werden. Hierbei ist zu beachten, dass dpkg-scansources kein Override-File benötigt. Die Angabe des zweiten Arguments „/dev/null“ muss hierbei also entfallen.

4.41 dpkg-scansources

◁ 4.40 dpkg-scanpackages △ 4.42 dpkg-checkbuilddeps ▷

4.42 dpkg-checkbuilddeps

4.41 dpkg-scansources 4.43 grep-dctrl

Mit diesem Programm lassen sich die zum Erzeugen eines Pakets notwendigen Abhängigkeiten überprüfen. Die Syntax lautet:

```
dpkg-checkbuilddeps -B [control-file]
```

Ohne weitere Parameter, versucht dpkg-checkbuilddeps die Datei debian/control auszuwerten. Es wird daraufhin geprüft ob, alle notwendigen Pakete auf dem System installiert sind, um das gewünschte Paket zu erzeugen. Sie können auch auf der Kommandozeile eine andere control-Datei angeben. Als einzige Option ist -B bekannt; die Verwendung dieser Option ignoriert Angaben in der Zeile Build-Depends-Indep.

4.42 dpkg-checkbuilddeps

◁ 4.41 dpkg-scansources △ 4.43 grep-dctrl ▷

4.43 grep-dctrl

4.42 dpkg-checkbuilddeps 4.44 dpkg-repack

Hiermit lassen sich die „control“-Dateien der Debian-Pakete durchsuchen. Dieses sehr spezialisierte grep-Kommando eignet sich, um alle Arten von Dateien zu durchsuchen, die dem Dateiformat folgen, das im „Debian Policy Manual“ (http://www.debian.org/doc/debian-policy/) beschrieben ist. Dies sind Dateien wie das „available“-File von dpkg, das „status“-File von dpkg sowie die „Packages“-Dateien einer Distribution, wie sie auf den CD-ROMs oder FTP-Servern zu finden sind.

Mit diesem Werkzeug lässt sich sehr schnell bestimmen, welche Pakete von einem bestimmten Maintainer betreut werden oder welche Version eines Pakets aktuell ist.

Die Syntax von grep-dctrl lautet:

```
Usage: grep-dctrl [OPTION...]
```

-!, --not

Negiert die folgende Suchanfrage.

-a, --and

Verknüpft Suchanfragen.

-c, --count

Zeigt nur die Anzahl der Suchergebnisse an.

--config-file=FILE

Benutzt die Datei FILE als Konfigurationsdatei.

-C, --copying

Zeigt die Copyright-Informationen zu diesem Programm an.

-d

Es wird nur die erste Zeile des Beschreibungsfeldes („Description“) der dem Suchmuster entsprechenden Pakete angezeigt.

--debug-optparse

Hilft bei der Fehlersuche.

-e, --eregex

Behandelt den Suchstring als POSIX-konformen, regulären Ausdruck.

-F, --field=FELDNAME,FELDNAME...

Zeigt nur die angegebenen Feldnamen aus der Paketbeschreibung an.

-i, --ignore-case

Ignoriert Groß- und Kleinschreibung im Suchbegriff.

-l, --errorlevel=NAME

Setzt den Errorlevel auf den gewünschten Wert.

-n, --no-field-names

Unterdrückt die Ausgabe der Feldnamen.

-o, --or

Verknüpft Suchbegriffe mit der ODER-Funktion.

-q, --quiet

Unterdrückt die Ausgaben auf der Standardausgabe.

-r, --regex

Behandelt den Suchbegriff als regulären Ausdruck nach POSIX-Standard.

-s, --show-field=FELD, FELD...

Zeigt nur den Inhalt der angegebenen Felder.

--silent

Unterdrückt Ausgaben auf der Standardausgabe.

-v, --invert-match

Zeigt nur Ergebnisse, die nicht auf den Suchbegriff zutreffen.

-X, --exact-match

Zeigt nur genau zutreffende Ergebnisse an.

-?, --help

Zeigt eine Hilfe zu diesem Programm an.

--usage

Zeigt eine kurze Erklärung der Optionen von grep-dctrl.

--version

Zeigt die Versionsnummer des Programms an.

4.43 grep-dctrl

4.42 dpkg-checkbuilddeps 4.44 dpkg-repack

4.44 dpkg-repack

4.43 grep-dctrl 4.45 dpkg-divert

Mit diesem Programm lassen sich bereits installierte Pakete wieder zu einem Debian Paket zusammenpacken. Wenn irgendwelche Veränderungen, beispielsweise an den Konfigurationsdateien, vorgenommen wurden, so werden diese in das Paket übernommen. Dies ermöglicht es, auf einfache Weise Pakete mit spezifischen Konfigurationen auf mehreren Systemen zu verteilen oder auch längst aus der Distribution verschwundene Pakete wiederzubeleben (solange noch auf einem anderen erreichbaren System die notwendige Software installiert ist). Auf diesem Wege kann auch der momentane Zustand eines Pakets gesichert werden, bevor Sie es upgraden.

dpkg-repack kennt die folgenden Optionen:

```
Usage: dpkg-repack [--root=dir] packagename [packagename ..]
--root=dir      Take package from filesystem rooted on <dir>.
--arch=arch     Force the arch to be built for architecture <arch>.
packagename     The name of the package to attempt to repack.
```

--root=dir

Erzeugt das Paket aus dem angegebenen Verzeichnis, beispielsweise eine gemountete Partition.

--arch=arch

Erzeugt das Paket für die angegebene Architektur.

Als abschließender Parameter ist lediglich der Name des zu packenden Pakets anzugeben.

4.44 dpkg-repack

4.43 grep-dctrl 4.45 dpkg-divert

4.45 dpkg-divert

4.44 dpkg-repack 4.46 dpkg-statoverride

Datei-„diversions“ - im Deutschen in etwa „Umleitungen“, „Ablenkungen“ oder auch „Diversifikationen“ - erlauben es dem Debian Paketmanagement, Dateien nicht an der vorgesehenen, sondern an einer „umgeleiteten“ Stelle im Dateisystem abzulegen. dpkg-divert kann innerhalb von Debian Paket-Skripten eingesetzt werden, um eine Datei aus einem anderen Paket zu überschreiben. Dabei wird, wie gleich noch gezeigt wird, die Datei nicht tatsächlich überschrieben, sondern nur umbenannt.

Während der Installation eines Pakets können so einzelne Dateien, beispielsweise Programme oder Manpages, durch andere Versionen ersetzt werden. Konfigurationsdateien sind von dieser Funktionalität ausgenommen. Um Konfigurationsdateien zu managen, ist entweder ein Paket zu erstellen, das das Ursprungspaket ersetzt oder mit diesem in Konflikt steht, oder aber es ist ein Paket zu erstellen, das Teile der Konfiguration enthält und diese mit dem Hauptpaket teilt (durch entsprechende Abhängigkeiten).

Beispielsweise kann ein Paket während der Installation ein Programm durch eine angepasste Version ersetzen. Hierzu ist sinngemäß Folgendes in das preinst-Skript des Pakets einzufügen:

```
if [ install = "$1" -o upgrade = "$1" ]; then     dpkg-divert --package
smailwrapper --add --rename \        --divert /usr/sbin/smail.real
/usr/sbin/smail  fi
```

In der Datei postrm des Pakets muss dieser Vorgang wieder rückgängig gemacht werden:

```
if [ remove = "$1" ]; then     dpkg-divert --package smailwrapper --
remove --rename \        --divert /usr/sbin/smail.real /usr/sbin/smail  fi
```

Die Dateien aus dem Debian Paket bleiben unangetastet mit Ausnahme der Umbenennung.

Mittels dpkg-divert „markierte“ Dateien werden bei einem späteren Update nicht von Dateien aus einem aktuelleren Paket überschrieben.

dpkg-divert kann ausschließlich auf Dateien und nicht auf Verzeichnisse angewendet werden.

Mit dpkg-divert können diese Umleitungen gesetzt, angezeigt oder auch wieder gelöscht werden. Weiterhin lassen sich die ursprünglichen Namen der umgeleiteten Dateien ermitteln. Die Syntax für das Kommando dpkg-divert lautet:

```
dpkg-divert [options] [--add] <file> dpkg-divert [options] --remove <file> dpkg-divert [options] --list <glob-pattern> dpkg-divert [options] --truename <file>
```

Die möglichen Optionen von dpkg-divert lauten:

--add (Optional)

Wird keine Option angegeben, so wird automatisch --add verwendet. Diese Option fügt die angegebene Datei in die Liste der zu diversifizierenden Dateien ein.

--admindir <Verzeichnis>

Setzt das Verzeichnis, in dem sich die Daten für dpkg befinden (normalerweise: /var/lib/dpkg/).

--divert <Umlenken nach>

Bestimmt einen neuen Namen für die Datei.

--help

Zeigt eine kurze Anleitung zu dpkg-divert an.

--local

Bewirkt, dass alle verfügbaren Versionen eines Paketes diversifiziert werden.

--list [<file>]

Zeigt die bestehenden Diversifikationen an. Wird ein Dateiname angegeben, so werden nur die entsprechenden Dateien angezeigt.

--package <Paketname>

Der Name eines Pakets, in dem die Kopie der Datei <Dateiname> nicht umgeleitet wird.

--quiet

Es werden keine Ausgaben auf dem Bildschirm erzeugt.

--rename

Verschiebt eine Datei. Existiert die Zieldatei bereits, so wird die Aktion abgebrochen.

--remove <file>

Löscht die angegebene Datei aus der Liste der diversifizierten Dateien und stellt den ursprünglichen Zustand wieder her.

--test

Testmodus; es werden keine Veränderungen durchgeführt.

--version

Zeigt die Versionsnummer von dpkg-divert an.

Beim Hinzufügen von Umleitungen werden, wenn nicht anders angegeben, immer die Option --divert <Ursprungsname>.distrib und die Option --local verwendet.

Eine Liste der umgeleiteten Dateien wird in der Datei /var/lib/dpkg/diversions gespeichert.

4.45 dpkg-divert

4.44 dpkg-repack 4.46 dpkg-statoverride

4.46 dpkg-statoverride

4.45 dpkg-divert 4.47 dpkg-query

Mittels dpkg-statoverride können einzelne Komponenten eines Pakets mit anderen Rechten, als vorgesehen waren, installiert werden. Dies kann bei Paketen sinnvoll sein, die Programme mit einem unerwünscht gesetzten setuid-Bit installieren oder wenn Programme nur von einer bestimmten Gruppe von Benutzern ausgeführt werden sollen. Eine Komponente kann dabei jedes Objekt im Dateisystem sein, Dateien, Verzeichnisse oder auch Gerätedateien.

dpkg-statoverride kennt verschiedene Optionen:

--add <user> <group> <mode> <file>

Überschreibt die Rechte für die Datei <file>. Dabei muss die Datei noch nicht existieren, wenn das Kommando ausgeführt wird. Die Informationen werden gespeichert und später bei der Installation der Datei auf dem System verändert. Benutzer- und Gruppennamen können sowohl im Klartext (also beispielsweise root oder nobody) oder auch als Nummern angegeben werden. Wird die Schreibweise mit Nummern bevorzugt, so ist das Zeichen # voranzustellen, also beispielsweise #0 oder #65534.

--remove <file>

Stellt die ursprünglich im Paket verwendeten Rechte wieder her.

--list [<glob-pattern>]

Zeigt alle veränderten Rechte an. Wird zusätzlich ein Suchbegriff angegeben, so wird die Ausgabe auf die Veränderungen beschränkt, auf die der Suchbegriff passt.

--force

Führt die gewünschte Aktion auch aus, wenn die internen Überprüfungen die Ausführung sonst nicht zulassen würden. Dies ist immer dann der Fall, wenn bereits Veränderungen an einer Datei vorliegen.

--update

Aktualisiert die Rechte sofort, wenn eine Datei existiert. Wird nur bei Verwendung der Option --add ausgeführt.

--quiet

Gibt weniger Informationen darüber aus, welche Veränderungen vorgenommen werden.

--help

Gibt einige Informationen zur Benutzung des Programms aus.

--admindir

Ändert das Verzeichnis, in dem die Datei statoverride gespeichert wird. Dies ist normalerweise das Verzeichnis /var/lib/dpkg/.

4.46 dpkg-statoverride

4.45 dpkg-divert 4.47 dpkg-query

4.47 dpkg-query

4.46 dpkg-statoverride 4.48 dpkg-architecture

Mit dpkg-query lässt sich die Paketdatenbank auf dem System abfragen. Dabei werden alle Pakete aus der Datei /var/lib/dpkg/available berücksichtigt. Die Abfrage ersteckt sich also sowohl auf installierte als auch auf nicht-installierte Pakete. dpkg-query kennt die folgenden Optionen:

-l oder --list Suchbegriff

Zeigt alle Pakete an, wenn kein Suchbegriff angegeben wird. Ansonsten wird die Ausgabe auf die dem Suchbegriff entsprechenden Pakete beschränkt. Es können in dem Suchbegriff auch Wildcards verwendet werden. Hierbei ist darauf zu achten, dass diese nicht von der Shell interpretiert werden. Mit dem Kommando dpkg-query -l 'apache*' würden beispielsweise alle Pakete ausgegeben werden, die mit „apache“ beginnen.

-w oder --show Suchbegriff

Wie die Option --list, jedoch kann die Ausgabe mit der Option --showformat angepasst werden.

-s oder --status Paketname

Zeigt den Status des angegebenen Pakets an. Dabei werden nur installierte Pakete berücksichtigt.

-L oder --listfiles Paketname

Zeigt alle installierten Dateien aus einem Paket an. Dateien, die während der Installation durch ein Skript erzeugt werden, sind dabei nicht berücksichtigt, da diese nicht in dem Paket enthalten sind.

-S oder --search Suchbegriff

Sucht nach einem Dateinamen in den installierten Paketen. Hierbei können wieder die aus der Shell bekannten Wildcards benutzt werden.

-p oder --print-avail Paketname

Zeigt Detailinformationen zu dem Paket an.

--license oder --license

Zeigt die Lizenz- und Copyright-Informationen zu diesem Programm an.

--version

Zeigt die Versionsnummer dieses Programms an.

--admindir=dir

Hiermit kann das Verzeichnis verändert werden, in dem die **dpkg**-Datenbank abgelegt wird. Dies sollte aber immer /var/lib/dpkg/ sein.

--showformat=format

Hiermit kann das Format der Ausgabe bei Verwendung von --show angepasst werden. Die Formatangabe ist eine Zeichenkette, die für jedes gefundene Paket ausgegeben wird. Paketinformationen können durch Variablen in der Form ${var[;width]} angegeben werden. Die Ausgabe ist dabei immer rechtsbündig ausgerichtet, eine linksbündige Ausgabe wird durch einen negativen Wert für die Breite erreicht.

Es können weiterhin die Escape-Sequencen \n (Newline), \r (Carriage Return) und \\ (Backslash) eingesetzt werden.

4.47 dpkg-query

◁ 4.46 dpkg-statoverride △ 4.48 dpkg-architecture ▷

4.48 dpkg-architecture

4.47 dpkg-query 4.49 debian-updates

dpkg-architecture ermittelt die Architektur des aktuellen Systems. Dies wird hauptsächlich beim Erzeugen von Debian Paketen benötigt, um so die Optionen für den Compiler entsprechend setzen zu können.

dpkg-architecture bietet dabei sowohl die Möglichkeit, die Architektur des Systems zu ermitteln, auf dem das Programm gestartet wird, als auch eine andere Architektur zu setzen.

Die Syntax von dpkg-architecture ist folgendermaßen aufgebaut:

```
Usage: dpkg-architecture [<option> ...] [<action> Options: -
a<debian-arch> set Debian architecture -t<gnu-system> set
GNU system type -f force flag (override variables set in
environment) Actions: -l list variables (default) -
q<variable> prints only the value of <variable> -s
print command to set environment variables -u print
command to unset environment variables -c <command> set
environment and run the command in it.
```

Die Ausgabe von dpkg-architecture auf einem Intel-System sieht ohne weitere Optionen wie folgt aus:

```
fr@wasabi:~$ dpkg-architecture DEB_BUILD_ARCH=i386
DEB_BUILD_GNU_CPU=i386 DEB_BUILD_GNU_SYSTEM=linux
DEB_BUILD_GNU_TYPE=i386-linux DEB_HOST_ARCH=i386
DEB_HOST_GNU_CPU=i386 DEB_HOST_GNU_SYSTEM=linux
DEB_HOST_GNU_TYPE=i386-linux
```

Abweichend hierzu noch ein Beispiel für ein System mit der PowerPC Architektur:

```
fr@inari:~$ dpkg-architecture DEB_BUILD_ARCH=powerpc
DEB_BUILD_GNU_CPU=powerpc
DEB_BUILD_GNU_SYSTEM=linux
DEB_BUILD_GNU_TYPE=powerpc-linux
DEB_HOST_ARCH=powerpc DEB_HOST_GNU_CPU=powerpc
DEB_HOST_GNU_SYSTEM=linux
DEB_HOST_GNU_TYPE=powerpc-linux
```

Aktuell kennt dpkg-architecture die Debian Architekturen: sh4eb, knetbsd-i386, netbsd-i386, hurd-i386, sh4, alpha, darwin-i386, ia64, mipsel, sparc, freebsd-i386, openbsd-i386, darwin-powerpc, arm, mips, sparc64, m68k, s390, sh3, hppa, s390x, kfreebsd-i386, powerpc, sh3eb, amd64. i386

Bekannte GNU System Typen sind zur Zeit: sh4eb-linux, i386-knetbsd-gnu, i386-netbsdelf-gnu, i386-gnu, sh4-linux, alpha-linux, i386-darwin, ia64-linux, mipsel-linux, sparc-linux, i386-freebsd, i386-openbsd, powerpc-darwin, arm-linux, mips-linux, sparc64-linux, m68k-linux, s390-linux, sh3-linux, hppa-linux, s390x-linux, i386-kfreebsd-gnu, powerpc-linux, sh3eb-linux, x86_64-linux, i386-linux.

4.48 dpkg-architecture

◁ 4.47 dpkg-query △ 4.49 debian-updates ▷

4.49 debian-updates

4.48 dpkg-architecture 4.50 configure-debian

Das Programm debian-updates erstellt einen Report über verfügbare Sicherheitsupdates für das System, auf dem es installiert ist. Die Informationen dazu werden von einem Debian Server bezogen, auf dem alle verfügbaren Updates gelistet sind. Für jedes in der Liste aufgeführte Paket wird geprüft, ob eine veraltete Version installiert ist. Ist dies der Fall, so wird eine Information auf der Konsole ausgegeben. Alternativ können diese Informationen auch per E-Mail verschickt werden.

Das Paket debian-updates ist leider nicht in den offiziellen Paketquellen enthalten. Steve Kemp pflegt das Paket unter http://www.steve.org.uk/Software/debian-updates/.

Die Syntax für dieses Kommando lautet

```
debian-updates [--options] ...
```

Hier ein Beispiel für den Aufruf von debian-updates ohne weitere Optionen:

```
sushi:~# debian-updates   DSA-730 bzip2 - race condition ----------
----------  <p>Imran Ghory discovered a race condition in bzip2, a
high-quality block-sorting file compressor and decompressor. When
decompressing a file in a directory an attacker has access to, bunzip2
could be tricked to set the file permissions to a different file the user has
permissions to.</p>       See the following URL for more details
http://www.debian.org/security/2005/dsa-730
```

debian-updates kennt die im Folgenden beschriebenen Optionen:

--file

Liest die Informationen über neue Pakete aus einer Datei, nicht von einem Server im Netz. Der Inhalt dieser Datei muss dem Inhalt, der von einem Server im Netz geliefert wird, entsprechen, die Informationen werden als RDF-Feed geliefert.

--help

Zeigt eine kurze Hilfe zu debian-updates an.

--mail

Sendet die Ergebnisse an die angegebene Mail-Adresse, es wird keinerlei Ausgabe auf der Konsole erzeugt.

--proxy

Die URL des zu verwendenden Proxy-Servers, in der Form http://hostname:port.

--user

Benutzername für den Proxyserver, falls notwendig.

--pass

Passwort für den Proxyserver, falls notwendig.

--packages

Die URL einer Packages-Datei mit der die DSA (Debian Security Announce - Sicherheitsgutachten, die auf der debian-security-announce Mailingliste veröffentlicht wurden) verglichen werden.

--url

Liest den RDF-Feed aus der angegebenen Adresse.

--verbose

Gibt mehr Informationen dazu aus, was das Programm an Aktionen ausführt. Ein Beispiel für den Aufruf mit dieser Option:

```
sushi:~# debian-updates --verbose Reading RDF from URL:
http://www.debian.org/security/dsa-long.en.rdf Downloading remote
URL: http://www.debian.org/security/dsa-long.en.rdf Downloaded
6427 bytes. Package bzip2 is installed and mentioned in an advisory.
Package bzip2 - has local version 1.0.2-1 Arch is : 'i386' Downloading
remote package details from :
http://security.debian.org/dists/woody/updates/main/binary-
```

i386/Packages.gz Downloading remote URL: http://security.debian.org/dists/woody/updates/main/binary-i386/Packages.gz Downloaded 236768 bytes. Package bzip2 - has remote version 1.0.2-1.woody2 Running: dpkg --compare-versions 1.0.2-1.woody2 le 1.0.2-1 Return code: 256 DSA-730 bzip2 - race condition ---------- Imran Ghory discovered a race condition in bzip2, a high-quality block-sorting file compressor and decompressor. When decompressing a file in a directory an attacker has access to, bunzip2 could be tricked to set the file permissions to a different file the user has permissions to. See the following URL for more details http://www.debian.org/security/2005/dsa-730 Package php4 is installed and mentioned in an advisory. Package php4 - has local version 4:4.3.10-2 Arch is : 'i386' Downloading remote package details from : http://security.debian.org/dists/woody/updates/main/binary-i386/Packages.gz Downloading remote URL: http://security.debian.org/dists/woody/updates/main/binary-i386/Packages.gz Downloaded 236768 bytes. Package php4 - has remote version 4:4.1.2-7.woody4 Running: dpkg --compare-versions 4:4.1.2-7.woody4 le 4:4.3.10-2 Return code: 0

--version

Zeigt die Versionsnummer von debian-updates an.

Zu beachten ist beim Einsatz von debian-updates, dass Sicherheitsupdates vom Debian Team nur für die jeweils aktuelle „stable“-Release zur Verfügung gestellt werden.

4.49 debian-updates

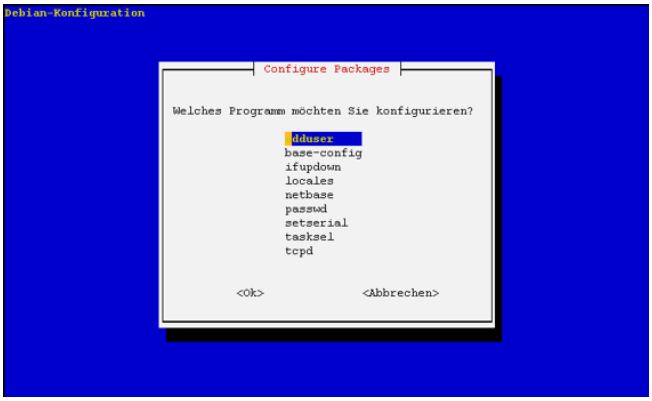
4.48 dpkg-architecture 4.50 configure-debian

4.50 configure-debian

4.49 debian-updates 4.51 debsums

Während der Installation von Paketen werden dem Administrator bei einigen Paketen Fragen zur Konfiguration gestellt. Diese werden in einer zentralen Datenbank (debconf, siehe debconf) gespeichert und können auch nachträglich mit dpkg-reconfigure (siehe dpkg-reconfigure) neu konfiguriert werden. Eine Frage bleibt dabei offen: Welche Pakete können so konfiguriert werden?

Mit dem Programm configure-debian steht ein Programm zur Verfügung, mit dem alle mittels debconf konfigurierbaren Pakete angezeigt und auch erneut konfiguriert werden können.



Zunächst müssen Sie den Abschnitt wählen, in dem sich das Programm befindet, beispielsweise admin, base, devel usw., danach kann das Paket ausgewählt und konfiguriert werden.

Rekonfigurierbare Pakete ermitteln

Alternativ zur Verwendung von configure-debian können natürlich auch die

Bordwerkzeuge eines Debian Systems eingesetzt werden. Die debconf-Vorlagen (Templates) eines Pakets befinden sich in der Datei paketname.config. Mit dem Befehl

```
fr@wasabi:~$ ls -1 /var/lib/dpkg/info/*.config /var/lib/dpkg/info/adduser.config
/var/lib/dpkg/info/apache-common.config /var/lib/dpkg/info/apache.config
/var/lib/dpkg/info/apache-ssl.config /var/lib/dpkg/info/bsdmainutils.config
/var/lib/dpkg/info/console-data.config /var/lib/dpkg/info/cupsys.config
/var/lib/dpkg/info/cvs.config /var/lib/dpkg/info/dash.config ...
```

lässt sich schnell eine Liste aller Pakete mit debconf-Unterstützung erstellen.

4.50 configure-debian

4.49 debian-updates 4.51 debsums

4.51 debsums

4.50 configure-debian 4.52 netselect

Mit dem Kommando debsums können die Checksummen von Dateien überprüft werden, die über Debian Pakete auf dem System installiert wurden. Mit diesem Werkzeug kann der Systemadministrator Veränderungen an den Dateien des Systems feststellen. Diese Veränderungen können gewollt sein (Veränderungen an Konfigurationsdateien, Datei temporär gegen eine selbst übersetzte Version getauscht... usw.) oder aber auch von einem Angreifer stammen, der Dateien im System gegen veränderte Binaries ausgetauscht hat (beispielsweise /bin/login).

```
fr@inari:~$ debsums --help debsums checks the MD5 sums of installed debian packages.  Usage: debsums [OPTIONS] [PACKAGE|DEB] ... Options:  -a, --all  check configuration files (normally excluded)  -c, --changed  report changed files (implies -s)  -l, --list-missing  list packages which don't have an md5sums file  -s, --silent  only report errors  -m, --md5sums=FILE  read list of deb checksums from FILE  -r, --root=DIR  root directory to check (default /)  -d, --admindir=DIR  dpkg admin directory (default /var/lib/dpkg)  -p, --deb-path=DIR[:DIR...]  search path for debs  -g, --generate=[all][,keep[,nocheck]]  generate md5sums from deb contents  --help  print this help, then exit  --version  print version number, then exit
```

Dem Programmaufruf können neben den Optionen noch Paketnamen übergeben werden, die gezielt überprüft werden sollen. Werden keine Paketnamen angegeben, so werden alle auf dem System installierten Pakete überprüft.

-a, --all

Überprüft auch die Checksummen der Konfigurationsdateien eines Pakets. Diese werden normalerweise nicht überprüft, da an diesen Dateien häufig Anpassungen vorgenommen werden.

-c, --changed

Zeigt veränderte Dateien an. Dies funktioniert nur zusammen mit der Option -s, da normalerweise alle Ergebnisse angezeigt werden.

-l, --list-missing

Zeigt Dateien an, für die keine MD5-Checksumme auf dem System vorhanden ist. Diese kann nachträglich mit der Option -g erzeugt werden. Dies sollte nur geschehen, wenn absolut sicher ist, dass die Dateien nicht zwischenzeitlich bereits von einem Angreifer verändert wurden!

-s, --silent

Zeigt ausschließlich Fehlermeldungen an. Informationen über korrekte Checksummen werden unterdrückt.

-m, --md5sums=FILE

Liest die Checksummen der Pakete aus der Datei FILE statt aus der Debian Datenbank.

-r, --root=DIR

Ändert das Startverzeichnis von „/“ auf den angegebenen Wert. Hiermit kann die Überprüfung der Checksummen auf Teilbereiche des Dateisystems eingegrenzt werden.

-d, --admindir=DIR

Ändert das Admin-Verzeichnis für dpkg (normalerweise /var/lib/dpkg/) auf den angegebenen Wert.

-p, --deb-path=DIR[:DIR...]

Ändert das Suchverzeichnis für Debian Pakete.

-g, --generate=[all][,keep[,nocheck]]

Erzeugt fehlende Checksummen.

4.51 debsums

4.50 configure-debian 4.52 netselect

4.52 netselect

4.51 debsums 4.53 deborphan

Debian kann sehr komfortabel mittels apt-get und entsprechenden Einträgen in der Datei sources.list über das Internet aktualisiert werden. Der Zugriff auf die via HTTP oder FTP verfügbaren Debian Server ist dabei unterschiedlich schnell. Natürlich möchte man einen netzwerktechnisch besonders gut erreichbaren Server zur Installation nutzen, doch welcher ist das? Hier hilft das Paket netselect.

netselect prüft aus der Liste der Debian Spiegel die Verfügbarkeit und die Übertragungsrate jedes einzelnen Servers. Abschließend wird der am besten erreichbare Server ausgegeben. Anhand dieser Angaben kann ein Eintrag in der Datei sources.list erfolgen.

Eine detaillierte Statistik aller angefragten Server kann mit der Option -vv ausgegeben werden. Weiterhin ist es möglich, auf der Kommandozeile nur eine Auswahl von Servern anzugeben; aus diesen wird dann der schnellste ermittelt.

```
fr@sushi:~$  netselect ftp.debian.org ftp.de.debian.org ftp.fr.debian.org
0 ftp.debian.org  fr@sushi:~$  netselect -vv ftp.debian.org
ftp.de.debian.org ftp.fr.debian.org Running netselect to choose 1 out of
3 addresses.        ..... ftp.debian.org                    0 ms
2 hops   90% ok ( 9/10) [ 0] ftp.de.debian.org                0 ms   2
hops   90% ok ( 9/10) [ 0] ftp.fr.debian.org                0 ms   2 hops
90% ok ( 9/10) [ 0] 0 ftp.debian.org
```

Etwas komfortabler lässt sich mit dem Programm netselect-apt aus dem gleichnamigen Paket arbeiten. Als Parameter kann das gewünschte Release („potato“„„woody“, „sid“ usw.) angegeben werden. netselect-apt holt zunächst die Liste aller Debian Spiegel, ermittelt den schnellsten Server und erzeugt (Achtung: im aktuellen Verzeichnis!) eine entsprechende Datei sources.list.

```
fr@sushi:~$ netselect-apt woody Retrieving the list of mirrors from
www.debian.org... --16:09:46--
```

```
http://www.debian.org/mirror/mirrors_full          => `mirrors_full'
...(weitere Aktivitäten)  Writing the sources.list in the current directory.
Done. fr@sushi:~$ ls -l sources.list  -rw-rw-r--    1 fr      fr          545
Mai 23  2002 sources.list
```

Die so entstandene Datei kann nun an die bestehende sources.list angehängt werden oder diese komplett ersetzen.

4.52 netselect

4.51 debsums 4.53 deborphan

4.53 deborphan

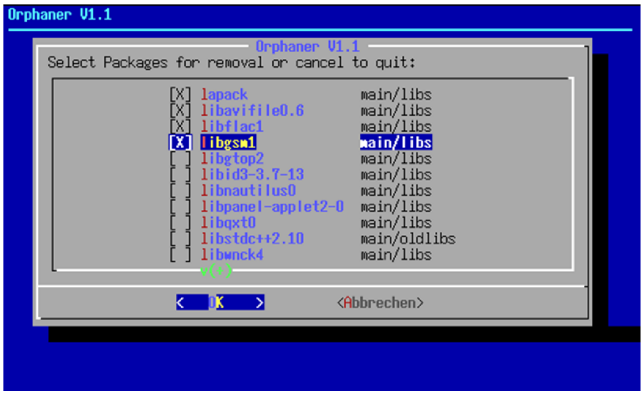
4.52 netselect 4.54 debfoster

deborphan ist ein Programm, das auf einem Debian System nach verwaisten Paketen sucht. Wird ein Paket gefunden, von dem keine anderen Pakete abhängig sind, so wird der Name ausgegeben. Dies ist hauptsächlich sinnvoll, um installierte Bibliotheken zu finden, die nicht mehr benötigt werden.

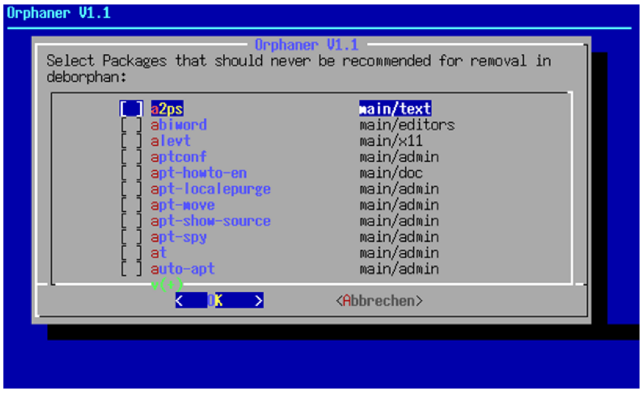
```
surimi:~# deborphan  libxosd0 libwww0 libgsm1 libqxt0 libwnck4
libavifile0.6 lapack libzvt2-0 libid3-3.7-13 libgtop2 libflac1 libnautilus0
libpanel-applet2-0 libstdc++2.10
```

Die angezeigten Pakete können nun mit den üblichen Tools entfernt werden. Wenn die Liste zunächst genau kontrolliert wurde (um zu verhindern, dass noch benötigte Pakete versehentlich gelöscht werden), können die nicht benötigten Pakete mit `dpkg --purge `deborphan`` oder `dpkg --purge `deborphan --guess-all`` oder auch `apt-get --purge remove `deborphan`` gelöscht werden.

Es stehen weiterhin zwei Frontends für deborphan zur Verfügung. Mit dem Programm orphaner können Sie „rekursiv“ Pakete entfernen.



editkeep kann verwendet werden, um eine Liste der Pakete zu erstellen, die nicht mehr von deborphan angezeigt werden sollen. So markierte Pakete werden auf dem System belassen bzw. in der Auflistung von deborphan nicht mehr angezeigt.



4.53 deborphan

4.52 netselect 4.54 debfoster

4.54 debfoster

4.53 deborphan 4.55 task-Pakete

debfoster entfernt automatisch nicht zwingend auf dem System benötigte Pakete. Hierzu wird eine Liste aller Pakete erstellt, die zur Installation ausgewählt wurden. In dieser Liste sind keine Pakete aufgeführt, die aufgrund einer Abhängigkeit zu einem Paket installiert wurden. Nur gezielt installierte Pakete werden aufgeführt. debfoster kann nach jedem Start von dpkg oder apt aufgerufen werden, um nicht länger benötigte Pakete aus dem System zu entfernen.

debfoster erfordert zunächst einen höheren Aufwand nach dem ersten Start. Es wird eine Liste aller Pakete definiert, und die Antworten werden gespeichert, so dass beim nächsten Start nur bei veränderten Paketen nachgefragt wird.

```
gnome-office is keeping the following 24 packages installed:   abiword-
gnome bonobo-conf dia-common dia-gnome evolution ganso gnucash
gnumeric gtkhtml guile1.4-slib libbonobo-conf0 libcamel0 libdate-
manip-perl   libfinance-quote-perl libgnome-pilot1 libgtkhtml-data
libguppi16   libgwrapguile1 libhtml-tableextract-perl libole2-0
libpisock4 libunicode0   mrproject slib Keep gnome-office?
[Ynpsiuqx?], [H]elp: Y  nautilus1.1 is keeping the following 10
packages installed:   libeel2-1 libeel2-data libgail-common libgail13
libgail15   libgnome-desktop-0 libgtkhtml2-0 libnautilus1.1-2
nautilus1.1-data   nautilus1.1-gtkhtml Keep nautilus1.1? [Ynpsiuqx?],
[H]elp: Y  aptconf is keeping the following 11 packages installed:
configlet-frontends gnome-sudo libxml-grove-perl libxml-perl   python-
configlet python-gdk-imlib python-glade python-gnome python-gtk
python-xml python2.1-xml Keep aptconf? [Ynpsiuqx?], [H]elp:
```

Weiterhin lässt sich debfoster über die Kommandozeile steuern. Somit können Sie viele Aktionen automatisieren.

Usage: debfoster [-ck FILE] [-adefhinopqrsvV] package1 package2- Installs package1, deinstalls package2 -v, --verbose Be a loudmouth -V, --version Show version and copyright information -h, --help Show this message -q, --quiet Silently build keeper file -f, --force Force system to conform to keeper file -m, --mark-only Do not install or delete packages -u, --upgrade Try to upgrade dependencies -c, --config FILE Specify configuration file -k, --keeperfile FILE Specify keeper file -n, --no-keeperfile Don't read keeper file -i, --ignore-default-rules Ignore default rules -a, --show-keepers Show packages on keeper list -s, --show-orphans Show orphaned packages -d, --show-depends PACKAGE Show all depends of PACKAGE -e, --show-dependents PACKAGE Show dependents of PACKAGE -p, --show-providers PACKAGE Show packages providing PACKAGE -r, --show-related PACKAGE Show packages brought in by PACKAGE -t, --use-tasks Make tasks visible as packages -o, --option OPT=VAL Override any configuration option See also: debfoster(8) debfoster 2.5 -- Copyright (C) 2000,2001 Wessel Dankers. Distributed under the GNU General Public License.

4.54 debfoster

4.53 deborphan 4.55 task-Pakete

4.55 task-Pakete

4.54 debfoster 4.56 Kernel-Pakete

4.55.1 tasksel

Die Auswahl aus einigen tausend Debian GNU/Linux-Paketen ist gerade für den Einsteiger nicht einfach. Aber auch Profis verlieren in der Masse der Pakete leicht den Überblick. Wer nicht zu jedem Paket die Beschreibung lesen und anhand dieser entscheiden will, ob ein Paket installiert werden soll oder nicht, kann sich in die Hände der Debian Entwickler begeben und bereits sorgfältig zusammengestellte Paketgruppen installieren. Es stehen mittlerweile eine ganze Reihe solcher task-Pakete zur Verfügung. Alle diese Pakete beginnen im Paketnamen mit task-, so dass Sie diese beispielsweise in dselect leicht heraussuchen (mit der Taste /) und installieren können.

task-Pakete stehen in verschiedenen Bereichen der Debian GNU/Linux-Distribution zur Verfügung. Anhand des Bereichs und des Namens lässt sich schon der Anwendungsbereich erkennen. Im Bereich net befinden sich die Pakete task-dialup, task-dialup-isdn, task-gnome-net und task-samba. Alle diese Pakete beinhalten eine Auswahl von nützlichen Paketen im Bereich Netzwerk. Hierzu gehören auch Netzwerkverbindungen über analoge oder digitale Telefonverbindungen (ISDN).

Wenn Sie ein solches Paket installieren, hier zum Beispiel das Paket task-dialup-isdn, werden eine ganze Reihe weiterer Pakete ausgewählt und einige andere zur Installation vorgeschlagen. Benutzen Sie hierzu zunächst einmal dselect:

```
dselect - recursive package listing          mark:+/=/- verbose:v help:?
EIOM Pri Section  Package     Description  _* Opt net      task-dialup-
Dialup utilities  _* Opt net      task-dialup  Dialup utilities  _* Xtr net
isdnutils    ISDN utilities  _* Xtr net      diald       dial on demand
daemon for PPP and SLIP.  _* Xtr admin    dialdcost   Cost estimation
and X Control panel for DI  _* Opt mail     fetchmail   POP2/3, APOP,
IMAP mail gatherer/forwarder  _* Opt web      junkbuster  The
Internet Junkbuster!  _* Xtr web      wwwoffle    World Wide Web
OFFline Explorer  __ Xtr news     leafnode    NNTP server for small
leaf sites  _* Opt news     inn         News transport system
`InterNetNews' by th  __ Xtr news     cnews       Simple News Server
for Usenet news.  __ Opt non-free diablo      News transport system
without reader suppor  _* Opt net      lftp        Sophisticated command-
line FTP/HTTP client  _* Opt admin    suidmanager  Manage File
Permissions  __ Xtr comm     mgetty      Smart Modem getty
replacement  __ Xtr net      isdnbutton  Start and Stop ISDN
```

connections and display __ Opt interpre python-tk Writing Tk applications with Python (Tkinter __ Opt web htdig WWW search system for an intranet or small i _* Opt news inewsinn NNTP client news injector, from InterNetNews __ Opt news innfeed This is the new INN feeder program `innfeed'.

Wenn Sie mit der Auswahl einverstanden sind, übernehmen Sie die Vorgaben einfach mit der Eingabetaste. Sie können aber auch weitere Pakete auswählen; nützlich ist beispielsweise das Paket isdnbutton.

Neben der Auswahl dieser task-Pakete via dselect oder apt steht mittlerweile auch ein spezielles Programm zur Verfügung, das sich speziell der task-Pakete annimmt: tasksel.

Mit dem Programm tasksel können Sie auf alle verfügbaren task-Pakete zugreifen. Ein task-Paket steht dabei jeweils für eine aufgabenbezogene Zusammenstellung von Paketen. Nach dem Start bekommen Sie eine Liste der verfügbaren Pakete angezeigt. Sie können mit den Cursortasten zwischen den Paketen wechseln und mit der Taste **RETURN** oder **SPACE** ein Paket zur Installation auswählen. Nochmaliges Drücken der Taste hebt diese Auswahl wieder auf.

Mit der Taste a wählen Sie alle verfügbaren Task-Pakete aus, mit der Taste n wird die Auswahl für alle bereits ausgewählten Pakete umgekehrt. Über die Taste i bekommen Sie weitere Informationen zu dem angewählten Paket; dort werden Ihnen auch die zu diesem Paket gehörenden weiteren Pakete angezeigt. Die Taste q beendet das Programm.

Platzbedarf von Task-Paketen

Die Basisinstallation eines Debian Systems nimmt circa 120 Megabyte Festplattenplatz in Anspruch, die zusätzlich empfohlenen Standard-Pakete beanspruchen circa 130 Megabyte, dabei werden rund 40 Megabyte Daten aus dem Netz geholt. Dies ergibt in der Summe 290 Megabyte benötigten Festplattenplatz für ein kleines Debian System.

Wenn Task-Pakete zur weiteren Installation eingesetzt werden sollen, ist es sinnvoll, vorab eine Abschätzung treffen zu können, wie viel Festplattenplatz benötigt wird. Im Folgenden sehen Sie eine Übersicht. Hierbei ist zu beachten, dass es Überschneidungen gibt. Werden zwei oder mehrere Task-Pakete installiert, so können die Summen nicht einfach addiert werden.

Task	Installed Size (MB)	Download Size (MB)	Space Needed To Install (MB)
desktop environment	345	118	463
X window system	78	36	114
games	49	14	63
Debian Jr.	340	124	464
dialup system	28	8	36
laptop system	3	1	4
scientific applications	110	30	140
C and C++	32	15	47
Python	103	30	133
Tcl/Tk	37	11	48
fortran	10	4	14
file server	1	-	1
mail server	4	3	7
usenet news server	6	2	8
print server	48	18	66
conventional unix server	55	19	74
web server	4	1	5
TeX/LaTeX environment	171	64	235
simplified Chinese environment	80	29	109
traditional Chinese environment	166	68	234
Cyrillic environment	29	13	42
French environment	60	18	78
German environment	31	9	40
Japanese environment	110	53	163
Korean environment	178	72	250
Polish environment	58	27	85
Russian environment	12	6	18
Spanish environment	15	4	19

tasksel ist sehr nützlich, wenn Sie sich nicht lange Gedanken über jedes einzelne zu installierende Paket machen wollen. Viele Anwender kennen sicher noch eine ähnliche Funktion aus der Installation von Debian 2.1. Dort konnte vor dem ersten Start von dselect auch eine solche Vorauswahl getroffen werden. Leider kam man zu einem späteren Zeitpunkt nie wieder an diese Stelle... Dieser Mangel wurde nun mit dem Programm tasksel behoben.

tasksel kennt auch einige Kommandozeilenoptionen:

```
debian:~# tasksel -h                                    Unknown
option: h                                               Usage:
tasksel install <Task> tasksel remove <Task> tasksel [Optionen];
Wobei Optionen eine Kombination von Folgendem sind:                -
t, --test          Test-Modus. Nichts ändern                    -r, --
required       Installiert Pakete mit der Priorität "required"          -i, --
important      Installiert Pakete mit der Priorität "important"          -s,
--standard       Installiert Pakete mit der Priorität "standard"          -
n, --no-ui         Keine Oberfläche zeigen, nur mit -r oder -i sinnvoll
--new-install    Installiert einige Tasks automatisch                --
list-tasks     Zeigt Tasks an und verlässt das Programm                 --
task-packages Zeigt vorhandene Pakete in einer Task                     --
task-desc      Gibt die Beschreibung einer Task aus
```

4.55 task-Pakete

4.54 debfoster 4.56 Kernel-Pakete

4.56 Kernel-Pakete

4.55 task-Pakete 4.57 base-config

Die Installation eines neuen Kernels kann nicht nur über den Weg eines aus den Sourcen selbst kompilierten Kernels erfolgen. Vom Debian Team werden auch bereits übersetzte Kernel bereitgestellt. Hierbei kann zwischen verschiedensten Kerneln, die jeweils für die entsprechenden Prozessoren optimiert sind, gewählt werden. Für den Kernel in der Version 2.6 stehen beispielsweise folgende Pakete zur Auswahl:

```
fr@surimi:~$ apt-cache search linux-image-2.6 linux-headers-2.6.18-4-486 - Header files for Linux 2.6.18 on x86 linux-headers-2.6.18-4-686 - Header files for Linux 2.6.18 on PPro/Celeron/PII/PIII/P4 linux-headers-2.6.18-4-686-bigmem - Header files for Linux 2.6.18 on PPro/Celeron/PII/PIII/P4 linux-headers-2.6.18-4-amd64 - Header files for Linux 2.6.18 on AMD64 linux-headers-2.6.18-4-k7 - Header files for Linux 2.6.18 on AMD K7 linux-headers-2.6.18-4-vserver-686 - Header files for Linux 2.6.18 on PPro/Celeron/PII/PIII/P4 linux-headers-2.6.18-4-vserver-k7 - Header files for Linux 2.6.18 on AMD K7 linux-headers-2.6.18-4-xen-686 - Header files for Linux 2.6.18 on i686 linux-headers-2.6.18-4-xen-vserver-686 - Header files for Linux 2.6.18 on i686 linux-headers-2.6.18-5-486 - Header files for Linux 2.6.18 on x86 linux-headers-2.6.18-5-686 - Header files for Linux 2.6.18 on PPro/Celeron/PII/PIII/P4 linux-headers-2.6.18-5-686-bigmem - Header files for Linux 2.6.18 on PPro/Celeron/PII/PIII/P4 linux-headers-2.6.18-5-amd64 - Header files for Linux 2.6.18 on AMD64 linux-headers-2.6.18-5-k7 - Header files for Linux 2.6.18 on AMD K7 linux-headers-2.6.18-5-vserver-686 - Header files for Linux 2.6.18 on PPro/Celeron/PII/PIII/P4 linux-headers-2.6.18-5-vserver-k7 - Header files for Linux 2.6.18 on AMD K7 linux-headers-2.6.18-5-xen-686 - Header files for Linux 2.6.18 on i686 linux-headers-2.6.18-5-xen-vserver-686 - Header files for Linux 2.6.18 on i686 linux-headers-2.6.18-6-486 - Header files for Linux 2.6.18 on x86 linux-headers-2.6.18-6-686 - Header files for Linux 2.6.18 on PPro/Celeron/PII/PIII/P4 linux-headers-2.6.18-6-686-bigmem - Header files for Linux 2.6.18 on PPro/Celeron/PII/PIII/P4 linux-headers-2.6.18-6-amd64 - Header files for Linux 2.6.18 on AMD64 linux-headers-2.6.18-6-k7 - Header files for Linux 2.6.18 on AMD K7 linux-headers-
```

2.6.18-6-vserver-686 - Header files for Linux 2.6.18 on PPro/Celeron/PII/PIII/P4 linux-headers-2.6.18-6-vserver-k7 - Header files for Linux 2.6.18 on AMD K7 linux-headers-2.6.18-6-xen-686 - Header files for Linux 2.6.18 on i686 linux-headers-2.6.18-6-xen-vserver-686 - Header files for Linux 2.6.18 on i686 linux-image-2.6-486 - Linux kernel 2.6 image on x86 linux-image-2.6-686 - Linux kernel 2.6 image on PPro/Celeron/PII/PIII/P4 linux-image-2.6-686-bigmem - Linux kernel 2.6 image on PPro/Celeron/PII/PIII/P4 linux-image-2.6-686-smp - Linux 2.6 image on PPro/Celeron/PII/PIII/P4 SMP - transition package linux-image-2.6-amd64 - Linux kernel 2.6 image on AMD64 linux-image-2.6-k7 - Linux kernel 2.6 image on AMD K7 linux-image-2.6-k7-smp - Linux 2.6 image on AMD K7 SMP - transition package linux-image-2.6-vserver-686 - Linux kernel 2.6 image on PPro/Celeron/PII/PIII/P4 linux-image-2.6-vserver-k7 - Linux kernel 2.6 image on AMD K7 linux-image-2.6-xen-686 - Linux kernel 2.6 image on i686 linux-image-2.6-xen-vserver-686 - Linux kernel 2.6 image on i686 linux-image-2.6.18-4-486 - Linux 2.6.18 image on x86 linux-image-2.6.18-4-686 - Linux 2.6.18 image on PPro/Celeron/PII/PIII/P4 linux-image-2.6.18-4-686-bigmem - Linux 2.6.18 image on PPro/Celeron/PII/PIII/P4 linux-image-2.6.18-4-amd64 - Linux 2.6.18 image on AMD64 linux-image-2.6.18-4-k7 - Linux 2.6.18 image on AMD K7 linux-image-2.6.18-4-vserver-686 - Linux 2.6.18 image on PPro/Celeron/PII/PIII/P4 linux-image-2.6.18-4-vserver-k7 - Linux 2.6.18 image on AMD K7 linux-image-2.6.18-4-xen-686 - Linux 2.6.18 image on i686 linux-image-2.6.18-4-xen-vserver-686 - Linux 2.6.18 image on i686 linux-image-2.6.18-5-486 - Linux 2.6.18 image on x86 linux-image-2.6.18-5-686 - Linux 2.6.18 image on PPro/Celeron/PII/PIII/P4 linux-image-2.6.18-5-686-bigmem - Linux 2.6.18 image on PPro/Celeron/PII/PIII/P4 linux-image-2.6.18-5-amd64 - Linux 2.6.18 image on AMD64 linux-image-2.6.18-5-k7 - Linux 2.6.18 image on AMD K7 linux-image-2.6.18-5-vserver-686 - Linux 2.6.18 image on PPro/Celeron/PII/PIII/P4 linux-image-2.6.18-5-vserver-k7 - Linux 2.6.18 image on AMD K7 linux-image-2.6.18-5-xen-686 - Linux 2.6.18 image on i686 linux-image-2.6.18-5-xen-vserver-686 - Linux 2.6.18 image on i686 linux-image-2.6.18-6-486 - Linux 2.6.18 image on x86 linux-image-2.6.18-6-686 - Linux 2.6.18 image on PPro/Celeron/PII/PIII/P4 linux-image-2.6.18-6-686-bigmem -

Linux 2.6.18 image on PPro/Celeron/PII/PIII/P4 linux-image-2.6.18-6-amd64 - Linux 2.6.18 image on AMD64 linux-image-2.6.18-6-k7 - Linux 2.6.18 image on AMD K7 linux-image-2.6.18-6-vserver-686 - Linux 2.6.18 image on PPro/Celeron/PII/PIII/P4 linux-image-2.6.18-6-vserver-k7 - Linux 2.6.18 image on AMD K7 linux-image-2.6.18-6-xen-686 - Linux 2.6.18 image on i686 linux-image-2.6.18-6-xen-vserver-686 - Linux 2.6.18 image on i686

Je nach vorhandener Hardware ist das entsprechende Paket zu wählen und beispielsweise mit apt-get vom Administrator zu installieren.

surimi:~# apt-get install linux-image-2.6-k7 Paketlisten werden gelesen... Fertig Abhängigkeitsbaum wird aufgebaut... Fertig Die folgenden zusätzlichen Pakete werden installiert: linux-image-2.6-k7 linux-image-2.6.18-4-k7 Vorgeschlagene Pakete: linux-doc-2.6.18 Die folgenden NEUEN Pakete werden installiert: linux-image-2.6-k7 linux-image-2.6.18-4-k7 0 aktualisiert, 2 neu installiert, 0 zu entfernen und 1 nicht aktualisiert. Es müssen 16,5MB Archive geholt werden. Nach dem Auspacken werden 48,9MB Plattenplatz zusätzlich benutzt. Möchten Sie fortfahren [J/n]? Hole:1 http://security.debian.org etch/updates/main linux-image-2.6.18-4-k7 2.6. 18.dfsg.1-12etch2 [16,5MB] Hole:2 http://ftp.de.debian.org etch/main linux-image-2.6-k7 2.6.18+6 [2030B] Es wurden 16,5MB in 2m19s geholt (118kB/s) Vorkonfiguration der Pakete ... Wähle vormals abgewähltes Paket linux-image-2.6.18-4-k7. (Lese Datenbank ... 66156 Dateien und Verzeichnisse sind derzeit installiert.) Entpacke linux-image-2.6.18-4-k7 (aus .../linux-image-2.6.18-4-k7_2.6.18.dfsg.1-12etch2_i386.deb) ... Done. Hole:1 http://security.debian.org etch/updates/main linux-image-2.6.18-4-k7 2.6. 18.dfsg.1-12etch2 [16,5MB] Hole:2 http://ftp.de.debian.org etch/main linux-image-2.6-k7 2.6.18+6 [2030B] Es wurden 16,5MB in 2m19s geholt (118kB/s) Vorkonfiguration der Pakete ... Wähle vormals abgewähltes Paket linux-image-2.6.18-4-k7. (Lese Datenbank ... 66156 Dateien und Verzeichnisse sind derzeit installiert.) Entpacke linux-image-2.6.18-4-k7 (aus .../linux-image-

2.6.18-4-k7_2.6.18.dfsg.1-12etch2_i386.deb) ... Done. Wähle vormals abgewähltes Paket linux-image-2.6-k7. Entpacke linux-image-2.6-k7 (aus .../linux-image-2.6-k7_2.6.18+6_i386.deb) ... Richte linux-image-2.6.18-4-k7 ein (2.6.18.dfsg.1-12etch2) ... Hmm. The package shipped with a symbolic link /lib/modules/2.6.18-4-k7/source However, I can not read the target: Datei oder Verzeichnis nicht gefunden Therefore, I am deleting /lib/modules/2.6.18-4-k7/source Running depmod. Finding valid ramdisk creators. Using mkinitramfs-kpkg to build the ramdisk. Running postinst hook script /sbin/update-grub. You shouldn't call /sbin/update-grub. Please call /usr/sbin/update-grub instead! Searching for GRUB installation directory ... found: /boot/grub Searching for default file ... found: /boot/grub/default Testing for an existing GRUB menu.lst file ... found: /boot/grub/menu.lst Searching for splash image ... none found, skipping ... Found kernel: /boot/vmlinuz-2.6.18-4-k7 Found kernel: /boot/vmlinuz-2.6.18-4-686 Updating /boot/grub/menu.lst ... done Richte linux-image-2.6-k7 ein (2.6.18+6) ...

Nach erfolgter Installation des Pakets befinden sich der eigentliche Kernel (vmlinuz-2.6.18-4-k7), die passende System.map (System.map-2.6.18-4-k7) sowie die Konfigurationsdatei, mit der das Kernel-Image erzeugt wurde (config-2.6.18-4-k7), im Verzeichnis /boot.

Weiterhin wurden die zum Kernel gehörenden Module unter /lib/modules/2.7.18-4-k7/ installiert.

Während der Installation des Kernel-Paketes wird eine passende Init-RAM Disk erzeugt und die Konfiguration des Bootloaders GRUB angepasst. Um den neuen Kernel zu aktivieren, muss das System nun neu gestartet werden.

Das Erstellen eines eigenen, an die individuellen Bedürfnisse angepassten Debian Kernel-Pakets ist im Abschnitt make-kpkg beschrieben.

4.56 Kernel-Pakete

4.55 task-Pakete 4.57 base-config

4.57 base-config

4.56 Kernel-Pakete 4.58 debootstrap

Mit base-config können die bereits bei der Installation vorgenommenen Einstellungen für die Zeitzone sowie die APT-Quellen zur Installation weiterer Pakete neu festgelegt werden. Abschließend kann direkt ein Frontend zur Paketauswahl (dselect, aptitude, tasksel o.Ä., je nachdem, welche Programme bereits installiert wurden) aufgerufen werden.

Alternativ können für diese beiden Schritte (Zeitzone, Paketquellen) auch die Programme tzconfig und apt-setup aufgerufen werden.

4.57 base-config

4.56 Kernel-Pakete 4.58 debootstrap

4.58 debootstrap

4.57 base-config 4.59 modconf

debootstrap kann verwendet werden, um ein Debian System von Grund auf neu zu installieren. Dies kann aus einem laufenden System heraus auf einer anderen Partition oder auch in einem Verzeichnis auf dem aktuellen System geschehen, beispielsweise um eine andere Release-Version von Debian zu testen. Auch ist es möglich, mittels debootstrap eine Installation von Debian von einer anderen Linux-Distribution aus vorzunehmen.

debootstrap nutzt die folgende Syntax:

```
debootstrap [OPTION]...  SUITE TARGET [MIRROR [SCRIPT]]
```

debootstrap benötigt neben den im Folgenden beschriebenen Optionen noch Informationen über die zu installierende Debian Version (SUITE) und das Verzeichnis (TARGET), in dem das neue System erzeugt werden soll. Weiterhin kann ein Debian Spiegel (MIRROR) angegeben werden, aus dem die notwendigen Pakete bezogen werden sollen. Optional kann auch ein Skript (SCRIPT) übergeben werden, dies wird während der Installation ausgeführt.

--arch ARCH

Architektur, für die das System erzeugt werden soll.

--download-only

Pakete werden nur vom Server geholt, aber nicht installiert.

--include=a,b,c...

Lädt die zusätzlich angegebenen Pakete vom Server und installiert diese. Beachten Sie, dass Abhängigkeiten von Anwender aufzulösen sind, es müssen also alle notwendigen Pakete aufgeführt werden.

--exclude=a,b,c...

Die aufgeführten Pakete werden nicht installiert und auch aus den internen Listen entfernt. Achtung: Dies wird wahrscheinlich essenziell wichtige Pakete löschen.

--variant=buildd

Installiert ein System mit allen für das Programm buildd notwendigen Paketen.

--verbose

Gibt mehr Informationen während der Installation aus.

--print-debs

Erzeugt eine Liste der zu installierenden Pakete und beendet dann das Programm.

--unpack-tarball DATEI

Entpackt die notwendigen Pakete aus dem Tar-Archiv DATEI. Es werden keine Pakete von einem Server geholt.

--boot-floppies

Nur für die internen Tests beim Erstellen von eigenen Boot-Disketten.

--debian-installer

Für interne Tests des Debian Installers.

Hier ein Beispiel für die Version „Sarge“: In dem Verzeichnis sarge-chroot/ im aktuellen Verzeichnis wird ein Basissystem installiert.

```
debootstrap sarge ./sarge-chroot http://ftp.debian.org/debian
```

4.58 debootstrap

4.57 base-config 4.59 modconf

4.59 modconf

4.58 debootstrap 4.60 shadowconfig

Mit modconf können zusätzliche Module in den Kernel geladen werden. modconf stellt ein einfaches Interface für die Kommandozeilen-Tools insmod bzw. modprobe dar. modconf wird ohne weitere Parameter aufgerufen.

Wie schon bei der Installation des Basissystems ist es notwendig, das zur Hardware passende Modul zu ermitteln und ggf. notwendige Parameter anzugeben.

4.59 modconf

4.58 debootstrap 4.60 shadowconfig

4.60 shadowconfig

4.59 modconf 4.61 tzconfig

Hiermit kann die Verwendung von Shadow-Passwörtern ein- und ausgeschaltet werden. Auch diese Option konnte bereits bei der Systeminstallation gewählt werden.

```
sushi:~# shadowconfig  Usage: /sbin/shadowconfig on | off
```

Als einzige Option kann diesem Kommando der Wert „on“ oder „off“ übergeben werden. Beachten Sie, dass beim Aus- und Wiedereinschalten dieser Option alle Informationen über den Gültigkeitszeitraum eines Passworts verloren gehen.

4.60 shadowconfig

4.59 modconf 4.61 tzconfig

4.61 tzconfig

4.60 shadowconfig 4.62 dlocate

Mit dem Kommando tzconfig kann die Zeitzone erneut eingestellt werden. Hier eine Beispielsitzung mit diesem Programm:

```
sushi:~# tzconfig  Your current time zone is set to Australia/Victoria
Do you want to change that? [n]: y  Please enter the number of the
geographic area in which you live:       1) Africa                  7)
Australia  2) America                    8) Europe       3) US time zones
9) Indian Ocean          4) Canada time zones           10) Pacific Ocean
       5) Asia                    11) Use System V style time zones       6)
Atlantic Ocean             12) None of the above   Then you will be
shown a list of cities which represent the time zone in which they are
located. You should choose a city in your time zone.  Number: 8
Amsterdam Andorra Athens Belfast Belgrade Berlin Bratislava
Brussels Bucharest Budapest Chisinau Copenhagen Dublin Gibraltar
Helsinki Istanbul Kaliningrad Kiev Lisbon Ljubljana London
Luxembourg Madrid Malta Minsk Monaco Moscow Nicosia Oslo Paris
Prague Riga Rome Samara San_Marino Sarajevo Simferopol Skopje
Sofia Stockholm Tallinn Tirane Tiraspol Uzhgorod Vaduz Vatican
Vienna Vilnius Warsaw Zagreb Zaporozhye Zurich  Please enter the
name of one of these cities or zones You just need to type enough
letters to resolve ambiguities Press Enter to view all of them again
Name: [] Berlin Your default time zone is set to 'Europe/Berlin'. Local
time is now:      Mit Okt 30 08:32:26 CET 2002. Universal Time is
now:  Mit Okt 30 07:32:26 UTC 2002. sushi:~#
```

4.61 tzconfig

4.60 shadowconfig 4.62 dlocate

4.62 dlocate

4.61 tzconfig 4.63 gpm

Wenn Sie sich ein wenig mit den Interna des Debian Paket-Systems auseinander gesetzt haben, sind Sie mit Sicherheit auch auf dpkg, das ursprüngliche Programm zum Paketmanagement, gestoßen. dpkg kann mit den Optionen -L und -S dazu benutzt werden, in den Paketdateien nach den Paketnamen zu suchen, in denen sich bestimmte Dateien befinden. Als schnelle Alternative dazu bietet sich das Programm dlocate an. Intern benutzt dlocate das GNU-Programm locate und kann über Parameter auch beispielsweise alle Dateien, die zu einem Paket gehören, anzeigen, den benötigten Festplattenplatz und Checksummen berechnen oder auch die passenden Manpages auflisten.

Wenn Sie dlocate ohne weitere Parameter aufrufen, bekommen Sie eine kurze Übersicht der Optionen:

```
linux:/home/fr/buch# dlocate Usage: dlocate [option] [string...]
Options:  (no option) string  list all records that match  -S  string
list records where files match  -L  package  list all files in
package  -l  package  almost-emulation of 'dpkg -l'  -s
package  print package's status  -ls  package  'ls -ldF' of all files
in package  -du  package  'du -sck' of all files in package  -conf
package  list conffiles in package  -lsconf  package  'ls -ldF' of
conffiles in package  -md5sum  package  list package's md5sums
(if any)  -md5check package  check package's md5sums (if any)  -
man  package  list package's man pages (if any)  The -L, -s, and -S
commands are roughly analagous to the  equivalent dpkg commands.
```

Suchen von nicht installierten Dateien

Die Programme dlocate oder auch dpkg -S können nur Dateien finden, die bereits auf Ihrem System installiert sind. Manchmal kommt es aber vor, dass Sie eine Datei oder ein Programm benötigen, von dem Sie den Namen kennen, aber nicht wissen, in welchem Paket es sich verbirgt. Sie können mit dem Kommando apt-cache search

exim.conf nach der gewünschten Datei suchen.

Die zweite Möglichkeit ist folgende: Datei Contents-i386.gz (auf anderen Architekturen als i386 setzen Sie bitte die entsprechende Architektur ein) enthält eine Liste aller Dateien mit den entsprechenden Paketnamen. Das Kommando zcat Contents-i386.gz | grep exim.conf sucht aus dieser Datei den Namen des Pakets heraus, zu dem die Datei exim.conf gehört.

4.62 dlocate

4.61 tzconfig 4.63 gpm

4.63 gpm

4.62 dlocate 4.64 mc (Midnight Commander)

gpm erlaubt es Ihnen, die Maus, sofern Sie eine an Ihrem Rechner angeschlossen haben, auch auf der textbasierten Konsole zu benutzen. Installieren Sie das Paket mit dselect oder apt. Während der Installation wird ein Konfigurationsskript aufgerufen, das nach der Schnittstelle der Maus, dem Typ sowie einigen anderen Parametern fragt.

Sie können dieses Konfigurationsskript auch später jederzeit verwenden, entweder wenn es bei der Erstkonfiguration Probleme gab oder wenn Sie eine andere Maus angeschlossen haben. Hierzu rufen Sie als Superuser (root) das Skript mit gpmconfig auf.

Am Beispiel einer Maus mit PS/2-Anschluss, wie sie an den meisten neueren Rechnern zu finden ist, werden wir die Konfiguration aufzeigen. Installieren Sie zunächst das Paket gpm, zum Beispiel mit apt-get install gpm:

```
Unpacking gpm (from .../misc/gpm_1.17.8-6.deb) ... Setting up gpm
(1.17.8-6) ... Configuring gpm (mouse event server):  Current
configuration: -m /dev/ttyS0 -t ms -l \     "a-zA-Z0-9_.:~/\300-\326\330-
\366\370-\377" Device: /dev/ttyS0 Type: ms Append: -l "a-zA-Z0-
9_.:~/\300-\326\330-\366\370-\377" Do you want to change anything
(Y/n)?
```

Die vorgegebenen Werte passen für eine ebenfalls recht verbreitete Maus an der seriellen Schnittstelle (COM1). Wenn Sie über eine solche Maus verfügen, können Sie die vorgegebenen Werte beibehalten. Bei einer PS/2-Maus sind aber einige andere Einstellungen nötig. Wählen Sie also Y, und fahren Sie mit der Installation wie folgt fort:

```
gpm has an experimental mouse test program which may help you
determine your mouse's type and which device it's attached to.  You
*must* not run any other software which needs the mouse, e.g. X,
selection or gpm, while running this program. Do you want to run
gpm's mouse-test program (Y/n)?
```

Wenn Sie sich sicher sind, dass Ihre Maus am PS/2-Anschluss angeschlossen ist, können Sie dieses Testprogramm überspringen; es schadet aber auch nicht, es zu benutzen. Probieren Sie es ruhig mal aus:

```
This program is designed to help you in detecting what type your mouse
is. Please follow the instructions of this program. If you're bored before
it is done, you can always press your 'Interrupt' key (usually Ctrl-C)
*** Remember: don't run any software which reads the mouse device
*** while making this test. This includes "gpm","selection", "X"  Note
that this program is by no means complete, and its main role is to detect
how does the middle button work on serial mice /dev/atibm: No such
device /dev/inportbm: No such device /dev/jbm: No such device
/dev/logibm: No such device ttyS1: No such file or directory  Trying
with 1200 baud  The possible device nodes are:     /dev/psaux
       /dev/rtc      /dev/ttyS0  /dev/ttyS1  /dev/ttyS2  /dev/ttyS3   I've still
6 devices which may be your mouse, Please move the mouse. Press any
key when done (You can specify your device name on cmdline, in order
to avoid this step. Different baud rates are tried at different times
```

Das Testprogramm hat auf unserem Beispielrechner sechs mögliche Gerätedateien festgestellt, an denen die Maus angeschlossen sein könnte. Für die PS/2-Maus ist gleich die erste, also /dev/psaux, zuständig. Bewegen Sie nun die Maus, und drücken Sie danach eine Taste.

```
removing "/dev/ttyS0" from the list removing "/dev/ttyS1" from the list
removing "/dev/ttyS2" from the list removing "/dev/ttyS3" from the list
Trying with 9600 baud   The possible device nodes are: /dev/psaux
Where is your mouse [/dev/psaux]?
```

Das Testprogramm hat nun alle Anschlüsse, an denen es keine Maus festgestellt hat, aus der Liste entfernt. Geben Sie nun das gefundene Device an:

```
> /dev/psaux What type is your mouse (or help) [ps2]?
```

Der Maustyp ist bei einer PS/2-Maus recht leicht zu bestimmen. Wenn Sie eine andere Maus benutzen, werfen Sie einfach mal mit help einen Blick in die Liste:

```
> help  gpm-Linux 1.17.8, $Date: 1999/01/03 21:02:51 $ Available
mouse types are:  name (synonym)       description  mman (Mouseman)
```

- The MouseMan protocol used by new Logitech mice. ms - For Microsoft mice (2 or 3 buttons). Some old 2 button mice send some spurious packets, which can be misunderstood as middle-button events. If this is happens to you, use the 'bare' mouse type. ms+ - Like 'ms', but allows dragging with the middle button. ms+lr - 'ms+', but you can reset m by pressing lr (see man page). bare (Microsoft) - For some 2 button Microsoft mice. Same as 'ms' except that gpm will not attempt to simulate a third button. msc (MouseSystems) - For most 3 button serial mice. sun - For Sparc mice. ... mtouch - For MicroTouch touch-screens (only button-1 events right now). acecad - For Acecad tablet in absolute mode (Sumagrapics MM-Series mode). kmiabps2 - For the Kensignton Mouse in a box on a PS/2 port (round connector with 6 pins), 3 buttons. Default for i386 is ms What type is your mouse (or help) [ps2]? > ps2

Der erkannte Maustyp ist für unser Beispiel aber korrekt und kann so beibehalten werden.

Set the responsiveness (normally not needed) []? > Repeat protocol (leave blank to turn repeating off) []? > Do you want to add any additional arguments [-l "a-zA-Z0-9_.:~/\300-\326\330-\366\370-\377"]? > Do you want to test this configuration (y/N)?

Diese vier Punkte können Sie erst einmal so belassen. Sie können u.a. die Empfindlichkeit der Maus einstellen; dies können Sie aber auch jederzeit später nach dem Studium der Manpage zu gpm detailliert erledigen.

Current configuration: -m ttyS1 -t ps2 -l "a-zA-Z0-9_.:~/\300-\326\330-\366\370-\377" Device: ttyS1 Type: ps2 Append: -l "a-zA-Z0-9_.:~/\300-\326\330-\366\370-\377"

Abschließend bekommen Sie noch einmal alle Daten angezeigt und können diese auch noch einmal ändern (dazu müssen Sie aber die ganze Konfiguration noch einmal durchgehen...).

```
Do you want to change anything (Y/n)? n Starting mouse interface server: gpm.
```

Wenn Sie mit den Einstellungen zufrieden sind, geben Sie ein `n` ein. Damit wird die Konfiguration gesichert und der `gpm`-Server gestartet. Sie können nun, wenn alles korrekt konfiguriert wurde, mit der Maus auf der Konsole Bereiche markieren und kopieren. Programme, die die Maus auch auf der Konsole unterstützen, wie zum Beispiel `mc`, können Sie auch mit der Maus benutzen.

4.63 gpm

4.62 dlocate 4.64 mc (Midnight Commander)

4.64 mc (Midnight Commander)

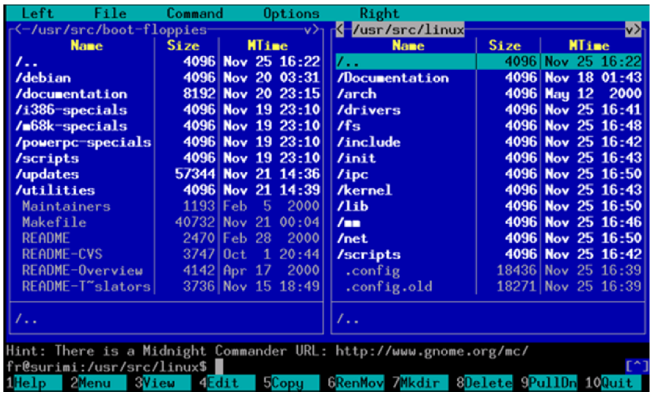
mc ähnelt stark dem aus der DOS-Welt bekannten Norton Commander. Soweit möglich, wird auch die Maus auf der textbasierten Oberfläche unterstützt.

Einige wichtige Funktionen des mc sind:

Arbeiten mit gepackten Dateien.

Wiederherstellen von gelöschten Dateien; dies unterliegt einigen Einschränkungen (benötigt mindestens Kernel 2.1.x).

Wenn Sie bereits mit dem nc unter DOS gearbeitet haben, werden Ihnen viele Dinge vertraut vorkommen.



4.64 mc (Midnight Commander)

4.63 gpm 4.65 screen

4.65 screen

4.64 mc (Midnight Commander) 4.66 ssh

screen ist ein sehr nützliches Programm im Zusammenhang mit virtuellen Konsolen. In der Urzeit der Unix-Geschichte waren ASCII-Terminals, die über die serielle Schnittstelle an den Rechner angeschlossen wurden, sehr verbreitet. Manchmal findet man solche Geräte auch noch heute im Einsatz oder kann diese günstig erstehen. Einem Einsatz an der seriellen Schnittstelle eines handelsüblichen PCs steht nichts im Wege, wenn man das passende Kabel hat. Leider stehen dann an einem solchen Terminal keine virtuellen Konsolen zur Verfügung.

Das gleiche Problem stellt sich beispielsweise, wenn man sich von außen über eine Modemverbindung an einem Rechner anmeldet; auch hier ist man auf ein einziges Terminal festgelegt.

Abhilfe schafft hier das Programm screen: Dieses „emuliert" mehrere virtuelle Terminals und kann auch die Sitzung nach dem Abbruch der Verbindung aufrechterhalten. Sie können, wenn Sie einen Account auf einem Rechner mit einer guten Internetanbindung haben, sich beispielsweise auf diesem einloggen, einen Download starten, sich abmelden und Stunden oder Tage später wieder anmelden und die Sitzung mit screen wieder aufnehmen: Der Download wird mittlerweile beendet sein, und Sie können sich die Daten mit voller Geschwindigkeit auf Ihren Rechner holen.

screen ist so etwas wie ein Full-screen-Windowmanager für die Konsole, der mehrere Prozesse, meist interaktive Shells, verwaltet. Jedes virtuelle Terminal hat die Funktionalität eines VT100-Terminals mit zusätzlichen Funktionen aus anderen Standards wie zum Beispiel ANSI X3.64 und ISO 2022 (hier wurden Funktionen wie das Einfügen und Löschen von Zeilen und die Unterstützung von verschiedenen Zeichensätzen entliehen). Jedes Terminal besitzt eine History-Funktion, und Daten können per Cut-and-Paste zwischen den Terminals kopiert werden.

Wenn Sie screen starten, wird ein Terminal mit einer Shell (oder einem anderen Programm) gestartet. Sie werden zunächst keinen Unterschied zu einer anderen Shell feststellen. Sie können aber nun zu jeder Zeit neue Terminals erzeugen - jedes mit einer neuen Shell, ebenso wie bei den „virtuellen" Terminals unter der Linux-Konsole. Sie können natürlich auch Terminals schließen, eine Liste der aktiven Terminals ansehen, die Ausgaben in einer Datei aufzeichnen und so weiter. Alle gestarteten Programme laufen weiter, auch wenn Sie zwischen den Terminals umschalten.

screen-Kommandos werden generell mit der Tastenkombination **STRG+a** eingeleitet. Mit der Kombination **STRG+a** bekommen Sie eine Hilfe zu screen in Form einer Befehlsübersicht. Hier eine kurze Übersicht der wichtigsten Kommandos:

? - Zeigt eine Übersicht der Kommandos.

c - Erzeugt ein neues Terminal.

n - Wechselt zum nächsten Terminal.

w - Zeigt an, wie viele Terminals aktiv sind.

Weitere Informationen zu screen finden Sie auf der Seite http://www.gnu.org/software/screen/.

4.65 screen

4.64 mc (Midnight Commander) 4.66 ssh

4.66 ssh

4.65 screen 4.67 Euro-Symbol

Secure Shell (ssh) ist ein Ersatz für rlogin und rsh und stellt eine verschlüsselte Verbindung in einem (unsicheren) Netzwerk - vor allem dem Internet - her. Sie sollten generell auf die Verwendung von Programmen verzichten, die Daten, insbesondere Passwörter, unverschlüsselt übermitteln. Vermeiden Sie möglichst auch die Benutzung von telnet. Die Telnet-Version, die von Debian GNU/Linux verwendet wird, unterstützt ebenfalls eine verschlüsselte Übertragung, aber gewöhnen Sie sich nicht daran; auf anderen Systemen ist dies kein Standard.

ssh verschlüsselt eine Verbindung bereits vor der Übertragung des Passworts. Sie können also einigermaßen sicher von jedem Ort aus sich auf einem Rechner einloggen. Ein Angreifer, der versucht, auf einem Rechner auf der Übertragungsstrecke Ihr Passwort aus dem Datenstrom zu filtern, hat so nur eine geringe Chance auf Erfolg. Er müsste genau zu dem Zeitpunkt das Datenpaket erwischen, zu dem Sie bei den ssh-Instanzen den Key austauschen!

Die einfachste Form der Benutzung von ssh ist der Aufruf mit der Angabe des gewünschten Zielrechners. Das Kommando

```
ssh hoshi.tischbahn.de
```

stellt eine verschlüsselte Verbindung zu diesem Rechner her und meldet Sie mit dem gleichen Benutzernamen wie auf dem lokalen Rechner an. Sie können zusätzlich einen anderen Benutzernamen mit der Option -l angeben:

```
ssh -l fr hoshi.tischbahn.de
```

oder auch:

```
ssh fr@hoshi.tischbahn.de
```

Wenn die beiden Benutzernamen identisch sind und der Rechner, von dem aus Sie sich einloggen, in der Datei /etc/hosts.equiv oder /etc/ssh/shosts.equiv aufgeführt ist, können Sie sich ohne Passwortabfrage anmelden.

Weitere Informationen finden Sie in der Manpage zu ssh (man ssh).

4.66 ssh

4.65 screen 4.67 Euro-Symbol

4.67 Euro-Symbol

4.66 ssh 4.68 Menüsystem

Um auf einem Debian System auch das Euro-Symbol auf der Konsole und unter X benutzen zu können, hat Javier Fernández-Sanguino Peña das „Debian Euro HOWTO“ (http://www.debian.org/doc/manuals/debian-euro-support/) geschrieben. Weiterhin stehen die beiden Pakete euro-support und tetex-eurosym zur Verfügung.

4.67 Euro-Symbol

4.66 ssh 4.68 Menüsystem

4.68 Menüsystem

4.67 Euro-Symbol 4.69 Paketmanagement für Umsteiger

Über das Debian GNU-Menüsystem werden alle auf dem System installierten Pakete automatisch in die Menüs und Menüleisten der Windowmanager oder auch des GNOME-Panels eingebunden. Neu installierte Pakete werden automatisch den Menüs hinzugefügt. Dies erledigen die Installationsskripte der einzelnen Pakete. Als Systemadministrator können Sie diese Einträge in den Menüs für alle Benutzer auf dem System ändern oder aber Ihre eigenen Menüs erzeugen bzw. die systemweit installierten verändern.

Die Debian GNU-Menüstruktur stellt sich wie folgt dar:

Apps -- Menü für Anwendungen Editors -- Programme zur Bearbeitung von Texten Net -- E-Mail, News, Webbrowser, IRC etc. Programming -- Debugger etc. Shells -- bash, ksh, zsh etc. Tools -- Diverse Tools: xclock, xmag, xman etc. Viewers -- Bildbetrachter, gs, xawtv etc. Math -- gnuplot, octave, oleo etc. Graphics -- xpaint, xfig, xtiff etc. Emulators -- dosemu etc. Sound -- TkMidity etc. System -- Systemverwaltung und -beobachtung Games -- Menü für Spiele Adventure -- Abenteuer, zork, MOO's etc. Arcade -- alles, was schnell ist... Board -- Brettspiele: Gnuchess, pente, gnugo Card -- Solitaire etc. Puzzles -- xpuzzles, ... Sports -- Sportliche Spiele Strategy -- lincity, freeciv Tetris-like -- Alles, was runterfällt... Toys -- oneko, xeyes etc. Screen -- Lock -- xlock etc. Screen-saver -- Bildschirmschoner Root-window -- Hintergründe Window-managers -- Umschalten zwischen verschiedenen Windowmanagern Modules -- fvwm modules etc. XShells -- shells (xterm, rxvt, ...)

Verändern Sie nichts an dieser Struktur. Die Paketbetreuer haben die Möglichkeit, Ergänzungen zu dieser Struktur einzubringen. Als Anwender sollten Sie dies aber so belassen.

Nach jeder Veränderung an den Menüeinträgen müssen Sie das Programm update-menus aufrufen. Beachten Sie hierbei, dass dies jeweils mit dem Benutzernamen

erfolgen muss, zu dem auch die Menüeinträge geändert wurden. Systemweite Änderungen werden vom Superuser (root) mittels update-menus dem System bekannt gegeben. Bei der Installation von Paketen wird update-menus automatisch aufgerufen.

Für jedes unter Debian GNU/Linux installierte Paket wird im Verzeichnis /usr/lib/menu/ mit dem jeweiligen Namen des Pakets (also zum Beispiel xterm) eine Datei angelegt. In dieser stehen Informationen zum Namen des Pakets, zum Text, der in den Menüs erscheinen soll, und zur Position des Eintrags in den Menüs. Ändern Sie nichts in diesen Dateien; diese können bei einem Update des Pakets überschrieben werden. Für systemweite Veränderungen steht das Verzeichnis /etc/menu/ zur Verfügung. In diesem kann der Systemadministrator eigene Dateien erzeugen oder Kopien einzelner Dateien aus /usr/lib/menu/ ablegen und diese verändern.

Hier eine Übersicht der Syntax dieser Dateien am Beispiel von GNUPlot:

```
?package(gnuplot):\          Der Name des Pakets      needs=text\
Benutzeroberfläche, die von diesem Programm
benötigt wird. Die möglichen Parameter dazu                sind:
needs=X11:              Dieses Programm benötigt X11      needs=text:
Für Programme, die nur im                Text-Modus laufen
(unter X11 muss                    der Windowmanager hierzu ein
xterm oder rxvt starten)     needs=vc:              Programm läuft nur
auf der Linux-Konsole      needs=wm:              Programm startet
einen anderen                    Windowmanager
section=Apps/Math\        Bezeichnet den Eintrag im Menübaum
title="Gnuplot"\          Der Text für den Eintrag (nicht zu lang)
command="/usr/bin/gnuplot" Der Pfad und Programmname, der
aufgerufen wird
```

Wenn Sie beispielsweise den Text, der für das Programm GNUPlot in den Menüs erscheint, ändern wollen, können Sie den Text in der Zeile title="Gnuplot"\ zwischen den Anführungszeichen ändern. Vergessen Sie nicht, danach update-menus aufzurufen.

Um einzelne Anwendungen an anderer Stelle im Menübaum erscheinen zu lassen, verändern Sie einfach die Zeile section=Apps/Math\.

Änderungen an den persönlichen Einstellungen für das Debian GNU/Linux-Menüsystem kann jeder Benutzer individuell in seinem Heimat-Verzeichnis unterhalb von ~/.menu vornehmen. Wenn dieses Verzeichnis noch nicht existiert, können Sie dieses mit mkdir .menu anlegen. Sie können nun in diesem Verzeichnis eigene Dateien anlegen oder

Dateien aus /usr/lib/menu/ kopieren und verändern. Diese Änderungen sind nicht systemweit gültig und werden nur für den jeweiligen Benutzer wirksam. Auch hier ist anschließend wieder das Programm update-menus aufzurufen, diesmal aber nicht als Superuser, sondern mit dem zugehörigen Account des Benutzers.

Neben dem Verändern von Einträgen besteht auch die Möglichkeit, Menüeinträge komplett auszublenden. Hierzu erzeugt man einfach eine leere Datei mit dem Namen des Pakets, das nicht mehr im Menübaum auftauchen soll. Das Kommando touch ist hierzu hervorragend geeignet:

```
cd ~/.menu touch gnuplot
```

... würde also beispielsweise den Eintrag für Gnuplot aus den Menüs entfernen. Auch dies funktioniert natürlich für jeden Benutzer im Verzeichnis ~/.menu/ oder systemweit unter /etc/menu/. Weitere Informationen zum Debian GNU/Linux-Menüsystem finden Sie in der Dokumentation zu dem Paket menu oder auf den Webseiten http://www.debian.org/doc/packaging-manuals/menu.html/. Dort sind auch weitere Interna beschrieben, wie zum Beispiel Informationen darüber, was zu tun ist, wenn man ein eigenes Debian Paket bei der Installation automatisch ins Menüsystem einbinden möchte.

4.68 Menüsystem

4.67 Euro-Symbol 4.69 Paketmanagement für Umsteiger

4.69 Paketmanagement für Umsteiger

4.68 Menüsystem 4.70 Installation von fremden Paketen

Dieser Abschnitt soll Umsteigern von anderen GNU-Linux-Distributionen das Paketmanagement erleichtern. Wenn Sie sich bereits in die Syntax von RPM eingearbeitet haben und mit den wichtigsten Kommandos vertraut sind, finden Sie hier schnell die entsprechenden Kommandos für Ihr neues Debian GNU-System.

RPM	DEB	Aktion
rpm -i *name*	dpkg -i *name*	Installiert das Paket *name*
rpm -i --nodeps*name*	dpkg --install --force-depends *name*	Installiert das Paket *name*, ohne Abhängigkeiten zu berücksichtigen
rpm -i --replacefiles*name*	dpkg --install --force-overwrite *name*	Installiert das Paket *name* und überschreibt bereits vorhandene Dateien
rpm -Va	debsums -a	Überprüft die Checksummen aller installierten Dateien auf dem System
rpm -qf /etc/syslog.conf	dpkg -S /etc/syslog.conf	Zeigt an, zu welchem Paket die Datei /etc/syslog.conf gehört
rpm -ql *name*	dpkg -L *name*	Listet alle Dateien des Paketes *name* auf
rpm -qpi *name.rpm*	dpkg -I *name.deb*	Zeigt die Informationen zum Paketarchiv *name.xxx*
rpm -qpl *name.rpm*	dpkg -c *name.deb*	Listet alle Dateien des Paketarchives *name.xxx* auf

rpm -qia	dpkg -l	Listet alle Pakete mit Informationen zu den Paketen. Das rpm-Kommando zeigt alle Paketinformationen, die dpkg-Variante nur die Kurzinformationen. Kurzinformationen können mit rpm über das Kommando rpm -qa --qf '%-20{NAME} %-10{VERSION} %{SUMMARY}\n' ausgegeben werden.
rpm: Funktion nicht implementiert	dpkg -r *name*	Entfernt das Paket *name*, Konfigurationsdateien werden nicht gelöscht
rpm -e *name*	dpkg --purge *name*	Entfernt das Paket *name* inklusive aller Konfigurationsdateien
rpm -qi *name*	dpkg -s *name*	Zeigt die Paketinformationen (Name, Status, Version, Abhängigkeiten usw.) zu dem Paket *name* an.
rpm -q --qf '%-20{NAME} %-10{VERSION} %{SUMMARY}\n'*name*	dpkg --list *name*	Zeigt eine Kurzbeschreibung zu dem Paket *name* an.
rpm -qa --qf '[%{=NAME}: %{FILENAMES}\n]' *string*	dpkg -S *string*	Sucht nach dem String *string* in den Paketinformationen/Dateilisten aller installierten Pakete
rpm -E *name*	dpkg --purge *name*	Löscht auch die Konfigurationsdateien des Paketes.

4.69 Paketmanagement für Umsteiger

4.68 Menüsystem 4.70 Installation von fremden Paketen

4.70 Installation von fremden Paketen

4.69 Paketmanagement für Umsteiger 4.71 Manuelles Entpacken von Debian Paketen

4.70.1 alien

Debian GNU/Linux verwendet ein eigenes Paketformat, das von keiner anderen Distribution (außer solchen, die auf Debian GNU/Linux basieren) verwendet wird. Natürlich sind aus diesem Grunde alle Programme zur Installation auf dieses Paketformat abgestimmt. Wir sind sicher, dass Sie alle benötigten Pakete in der Debian GNU/Linux-Distribution finden werden. Trotzdem zeigen wir Ihnen hier einen Weg, um fremde Pakete auf einem Debian GNU/Linux-System zu installieren. Dies kann auch zur Installation von kommerzieller Software nützlich sein, die häufig im rpm-Format vorliegt.

Das Programm alien dient dazu, Pakete in fremden Formaten auf einem Debian GNU/Linux-System zu installieren. alien ist natürlich auf allen Linux-Distributionen lauffähig und kann alle verbreiteten Paketformate entpacken und in das Format der verwendeten Distribution konvertieren.

Natürlich gibt es bei solchen Konvertierungen manchmal Probleme. Das zu installierende Paket wird in den meisten Fällen Bibliotheken verwenden, die nicht zwingend auf dem Zielsystem vorhanden sind, oder eventuell nicht in der passenden Version vorliegen. Gegebenenfalls müssen Sie auch diese Bibliotheken via alien installieren. In den meisten Fällen geht die Installation aber problemlos über die Bühne.

alien hat eine Reihe von Optionen; eine Übersicht bekommen Sie, wenn Sie alien ohne weitere Optionen aufrufen.

```
linux:/home/fr/rpm2deb# alien Usage: alien [options ...] file [...]   file [...]              Package file or files to convert.   -d, --to-deb
Generate a Debian deb package. (default)      Enables the following options:       --patch=<patch> Specify patch file to use instead of automatically looking for patch in                    /var/lib/alien.
--nopatch      Do not use patches.   -r, --to-rpm              Generate a RedHat rpm package.   -t, --to-tgz              Generate a Slackware tgz package.       --to-slp              Generate a Stampede .slp package.   -i, --install             Install generated package.   -g, --generate            Unpack, but do not generate a new package.   -s, --single              Like --generate, but do not create .orig                    directory.   -c, --scripts             Include scripts in package.   -k, --keep-version        Do
```

```
not change version of generated package.       --description=<desc>
Specify package description.   -h, --help                Display this help
message.   -v, --version                  Display alien's version number.
```

Die Benutzung von alien ist denkbar einfach. Im einfachsten Fall gibt man alien einfach den Namen des zu konvertierenden Pakets; alien erstellt dann daraus ein Paket mit der Endung .deb. Dieses können Sie dann wie jedes andere Paket unter Debian GNU/Linux installieren. Als Beispiel soll uns das Programm pi-address dienen. Wenn Sie über einen Palm-Pilot verfügen, sollten Sie das Debian Paket nach dem Versuch nicht löschen; wir kommen später noch einmal darauf zurück.

```
linux:/home/fr/rpm2deb# alien pilot-address-0.3.0-1.i386.rpm --
Examining pilot-address-0.3.0-1.i386.rpm -- Unpacking pilot-address-
0.3.0-1.i386.rpm 2813 blocks ---- -- Automatic package debianization --
Building the package pilot-address_0.3.0-2_i386.deb dh_testdir #
Nothing to do. dh_testdir dh_testroot dh_clean -k dh_installdirs cp -a `ls
|grep -v debian` debian/tmp dh_installdocs dh_installexamples
dh_installmenu dh_installcron dh_installchangelogs dh_compress
dh_suidregister dh_installdeb dh_shlibdeps dh_gencontrol
dh_makeshlibs dh_md5sums dh_builddeb dpkg-deb: building package
`pilot-address' in `../pilot-address_0.3.0-2_i386.deb'.  Generation of
pilot-address_0.3.0-2_i386.deb complete. -- Successfully finished
linux:/home/fr/rpm2deb# ls -l total 860 -rw-r--r-- 1 root root 437258
Feb 2 16:22 pilot-address-0.3.0-1.i386.rpm -rw-r--r-- 1 root root 435982
Feb 2 16:23 pilot-address_0.3.0-2_i386.deb
```

Sie können beispielsweise ein RedHat/Fedora-Paket mit dem Kommando alien paket.rpm konvertieren; Sie erhalten dann ein Debian Paket mit der Endung .deb. In diesem Beispiel würde die Datei also paket.deb heißen. Sie können dieses dann wie üblich mit dpkg -i paket.deb installieren.

4.70 Installation von fremden Paketen

4.69 Paketmanagement für Umsteiger 4.71 Manuelles Entpacken von Debian Paketen

4.71 Manuelles Entpacken von Debian Paketen

4.70 Installation von fremden Paketen | Kapitel 5. Debian Pakete im Detail

Um Einblick in ein Debian Paket zu erhalten, muss dieses nicht erst installiert werden. Debian Pakete basieren auf einem einfachen Archiv-Format und können mit dem Kommando ar entpackt werden. Beachten Sie dabei, dass das Paket nicht sinnvoll in das Debian Paketsystem integriert und installiert wird. Es wird lediglich in seine Bestandteile zerlegt.

Um das Paket in die Basiskomponenten aufzugliedern, ist folgendes Kommando anzuwenden:

```
ar -x paket.deb /tmp/paket
```

Alle Bestandteile sind danach im Verzeichnis /tmp/paket/ zu finden.

Um nur einen bestimmten Teil des Debian Pakets zu entpacken, kann dieser gezielt angegeben werden:

```
ar -x paket.deb data.tar.gz
```

Eine genaue Beschreibung der Bestandteile eines Debian Pakets finden Sie in Debian Paketformat.

4.71 Manuelles Entpacken von Debian Paketen

4.70 Installation von fremden Paketen | Kapitel 5. Debian Pakete im Detail

Kapitel 5. Debian Pakete im Detail

4.71 Manuelles Entpacken von Debian Paketen 5.2 Debian Kernel-Pakete erzeugen

Inhaltsverzeichnis

Wenn Sie sich schon einige Zeit intensiver mit Debian auseinander gesetzt haben, so kommt vielleicht der Wunsch auf, eigene Pakete für das lokale System zu erzeugen. Vielleicht wollen Sie aber auch Pakete für Freunde bereitstellen oder sogar an der Debian Entwicklung teilhaben. Auch wenn es bei einigen tausend Debian Paketen schwerfällt, für einen Anwendungsbereich keine passende Software zu finden, so kann es doch sinnvoll sein, bestehende Pakete anzupassen (beispielsweise mit einer entsprechenden Konfiguration für das lokale Netzwerk) oder gar komplett neue Pakete zu erzeugen, die noch nicht als Debian Paket verfügbar sind.

Für beide Probleme finden Sie in diesem Abschnitt eine Lösung. Doch zunächst zum Aufbau der Pakete.

5.1.1 Abhängigkeiten

5.1.2 (De-)Installationsprozess

Das Debian Paketmanagement benutzt ausschließlich Pakete im Format .deb. Das Format ist sehr einfach aufgebaut und basiert auf folgenden Prinzipien:

Das Paket ist mit Standard-Unix-Kommandos zu entpacken.

Das Format muss einfach zu erweitern sein.

Das Paketformat ist ein ar-Archiv und enthält die folgenden drei Dateien:

debian-binary enthält die Versionsnummer des verwendeten Debian-Paketformats, beispielsweise „2.0“.

control.tar.gz ist ein komprimiertes tar-Archiv mit den Metadaten des Pakets.

data.tar.gz ist ein komprimiertes tar-Archiv mit den eigentlichen Dateien, die vom Programm benötigt werden.

Pakete sind keine einsamen Objekte im Debian Paketsystem, sondern arbeiten als komplexes System in ihrer Gesamtheit zusammen. Um diese Zusammenarbeit zu steuern, gibt es eine Anzahl von Schlüsselwörtern, um die Abhängigkeiten zu beschreiben.

Pre-Depends hier werden andere Pakete aufgeführt, die vollständig installiert sein müssen, um dieses Paket zu installieren.

Depends führt Pakete auf, die vollständig installiert sein müssen, bevor dpkg dieses Paket konfigurieren kann.

Conflicts Pakete, die nicht zusammen mit diesem installiert sein können.

Provides liefert den Namen des von diesem Paket gelieferten virtuellen Pakets.

Replaces Pakete, die durch dieses Paket ersetzt werden.

Recommends, Suggests enthält eine Liste von Paket-Empfehlungen, die zu diesem Paket passen.

Die Installation und die Deinstallation von Debian Paketen läuft nach einem festen Schema ab. Die Installation eines neuen Pakets ist recht einfach:

<new-preinst> install

Entpacken des Pakets

<new-postinst> configure

Das Aktualisieren eines Pakets ist etwas komplizierter, da sowohl das bereits installierte als auch das neue Paket betroffen sind:

<old-prerm> upgrade <new-version>

<new-preinst> install <old-version>

Das Paket wird entpackt:

<old-postrm> upgrade <new-version>

<new-postinst> configure <old-version>

Beim Entfernen von Paketen ist zwischen dem Kommando remove und purge zu unterscheiden. Letzteres entfernt auch die Konfigurationsdateien eines Pakets aus dem System.

Löschen von Paketen:

prerm remove

Dateien werden gelöscht (Konfigurationsdateien bleiben erhalten!)

postrm remove

Die Hilfsskripte (mit Ausnahme von postrm) werden gelöscht.

Wird das Kommando purge benutzt, so werden zwei zusätzliche Schritte ausgeführt:

Konfigurationsdateien und eventuell vorhandene Kopien davon werden gelöscht

postrm purge

Das Paket wird aus der Paket-Datenbank entfernt.

Kapitel 5. Debian Pakete im Detail

4.71 Manuelles Entpacken von Debian Paketen 5.2 Debian Kernel-Pakete erzeugen

5.2 Debian Kernel-Pakete erzeugen

Kapitel 5. Debian Pakete im Detail 5.3 Debian Pakete anpassen

Auf einem Debian GNU/Linux-System gibt es grundsätzlich zwei verschiedene Verfahren, um einen individuellen Kernel zu erzeugen. Zunächst ist es natürlich möglich, alle Arbeitsschritte auf klassischem Wege „zu Fuß“, wie auf jedem anderen Linux-System, durchzuführen. Die elegantere Methode ist es jedoch, ein eigenes Debian-Kernel-Paket zu erzeugen und dieses zu installieren.

Sehen wir uns zunächst die etwas elegantere Debian Methode an; die klassische Methode werden wir später besprechen.

Das Paket kernel-package unterstützt den Administrator bei der Verwaltung von individuellen Linux-Kerneln auf seinem System.

Sicherlich haben Sie schon die bei Debian GNU/Linux mitgelieferten Kernel-Pakete entdeckt. Diese sind so konfiguriert, dass sie auf den meisten Systemen funktionsfähig sind. Trotzdem ist es manchmal notwendig, einen eigenen Kernel zu übersetzen, um neue Hardware einzubinden oder um Hardware-Konflikte zu umgehen. Sie können hierbei das Paket kernel-package zu Hilfe nehmen. Dieses erzeugt ein Debian Paket mit einem individuellen Kernel und allen Modulen für Ihr System.

Zunächst benötigen Sie, neben dem Paket kernel-package, das Sie schon installieren können, ein Paket mit dem gewünschten Kernel-Quellcode (Source). Es stehen einerseits Debian Pakete mit dem Quellcode des Kernels zur Verfügung (Paketname linux-source-2.x.xx), welche mit apt-get installiert werden können. Es können aber auch die Original Kernel-Quellen verwendet werden. Sie finden die Archive unter ftp://ftp.kernel.org oder auf den Spiegelservern in Deutschland: ftp://ftp.de.kernel.org oder ftp://ftp2.de.kernel.org.

Aktuelle Kernel-Version herausfinden

Wenn Sie sich nicht per FTP durch die Verzeichnisse auf dem Kernel-Server „wühlen“ möchten, um nachzusehen, ob es eine neue Version gibt, können Sie auf der Webseite www.kernel.org/finger_banner die aktuellen Versionen direkt abrufen.

The latest mainline 3 version of the Linux kernel is: 3.11-rc5 The latest stable 3.10 version of the Linux kernel is: 3.10.7 The latest stable 3.9 version of the Linux kernel is: 3.9.11 (EOL) The latest longterm 3.4 version of the Linux kernel is: 3.4.58 The latest longterm 3.2 version of the Linux kernel is: 3.2.50 The latest longterm 3.0 version of the Linux kernel is: 3.0.91 The latest longterm 2.6.34 version of the Linux kernel is: 2.6.34.14 The latest longterm 2.6.32 version of the Linux kernel is: 2.6.32.61 The latest linux-next version of the Linux kernel is: next-20130816

Weiter benötigen Sie einige zusätzliche Pakete, um einen neuen Kernel zu übersetzen, dies sind: gcc, libc5-dev oder besser (weil aktueller) libc6-dev, binutils, make, gawk oder mawk, gzip, shellutils, grep sowie bin86 auf der i386-Plattform. Wenn Sie das Kommando make menuconfig zur Kernel-Konfiguration benutzen möchten, muss das Paket libncurses5-dev installiert sein. Aber sicher haben Sie einige davon bereits installiert.

Wenn die notwendigen Pakete installiert sind, entpacken Sie die Kernel-Quellen; üblicherweise geschieht dies unter /usr/src/.

Wenn Debian Kernel-Source-Pakete verwendet werden, so befindet sich das zu entpackende Archiv bereits im Verzeichnis /usr/src/. Natürlich kann auch jeder andere Linux-Kernel-Source verwendet werden.

Beim Entpacken der Sourcen ist darauf zu achten, dass das Verzeichnis /usr/src/linux/ nicht durch Inhalte aus dem Archiv überschrieben wird. Wenn Sie verschiedene Kernel-Versionen verwalten wollen, empfiehlt es sich, das Verzeichnis /usr/src/linux/ umzubenennen, beispielsweise in linux-source-2.4.31.

Sorgen Sie auch dafür, dass das Verzeichnis /usr/src/ auf einer Partition mit ausreichend Platz angelegt ist. Die aktuellen Kernel-Archive in der Version 2.6 sind in gepackter Form ca. 40-50 MByte (je nach verwendetem Komprimierungsprogramm) groß und nehmen im entpackten Zustand ca. 220 MByte ein, wenn alles übersetzt ist - und zwar pro entpackter Kernel-Version!

Wechseln Sie nun in das Verzeichnis, in dem die Kernel-Quellcodes liegen.

Der Name eines Debian GNU/Linux-Kernel-Pakets besteht immer aus dem Basisnamen (hier: linux-image), der Versionsnummer des Kernels (zum Beispiel 2.6.18, diese wird aus dem Kernel-Makefile ermittelt) und der so genannten Revisionsnummer; Letztere können Sie individuell vergeben (über die Option --revision, die Sie dem Programm make-kpkg übergeben können). Sie sollten diese Revisionsnummer eindeutig wählen, um zu verhindern, dass ein bereits installierter Kernel überschrieben wird. Weiterhin darf das Zeichen „_“ (Unterstrich) nicht verwendet werden. Alternativ können Sie auch die Umgebungsvariable DEBIAN_REVISION auf den gewünschten Wert setzen.

Sie sollten die Revisionsnummer bei jedem neuen Kernel erhöhen; das Debian Paketsystem kann so automatisch ein Update durchführen.

Auch für das Paket kernel-package gibt es natürlich eine Konfigurationsdatei; diese finden Sie wie üblich im Verzeichnis /etc/ als kernel-pkg.conf. Üblicherweise sollten Sie dort mindestens Ihren Namen sowie die E-Mail-Adresse angeben. Sie können so immer feststellen, dass dieses Paket kein Original Debian Paket ist. Sie können, wenn nötig, noch weitere Variablen in dieser Datei benutzen. Momentan werden folgende Optionen unterstützt:

maintainer: Der „ Betreuer“ dieses Kernel-Pakets. Wenn Sie hier einen Apostroph (') verwenden möchten, so müssen Sie dieses wie folgt angeben: John O'\''Brien.

email: Ihre E-Mail-Adresse.

pgp: Der Name, der in der PGP-Datenbank gesucht werden soll. Normalerweise wird hier der Maintainer automatisch eingesetzt. Sie können dies auch mit der Umgebungsvariablen PGP_SIGNATURE überschreiben.

debian: Die Revisionsnummer des Pakets; der Standardwert ist 1.00. Sie können die Umgebungsvariable DEBIAN_REVISION benutzen.

image_in_boot: Wenn Sie diese Variable auf TRUE setzen, wird der Kernel im Verzeichnis /boot/ abgelegt und ein entsprechender symbolischer Link angelegt, anstatt wie sonst üblich den Kernel direkt in das „ root“-Verzeichnis (/) zu kopieren. Dies kann auch über die Umgebungsvariable IMAGE_IN_BOOT gesetzt werden.

kimage: Typ des Kernel-Images, zum Beispiel zImage oder bzImage; der Standardwert ist bzImage. Dieser Wert kann über die Umgebungsvariable IMAGE_TYPE gesetzt werden.

no_symlink: Kann nicht zusammen mit reverse_symlink verwendet werden. Sinnvoll kann diese Option im Zusammenspiel mit image_in_boot verwendet werden. Bei Verwendung der Option no_symlink wird das Kernel-Image immer als Datei vmlinuz abgelegt (und nicht als /boot/vmlinuz-x.x.xx). Ein bereits existierendes Kernel-Image wird in jedem Fall (und nicht nur, wenn es sich vom neuen Kernel-Image unterscheidet) in

vmlinuz.old umbenannt. Dieses bringt eine Beschränkung auf zwei Kernel-Images mit sich; weitere Versionen müssen dann von Hand eingepflegt werden. Diese Option kann auf Systemen eingesetzt werden, die keine symbolischen Links unterstützen, beispielsweise wenn loadlin eingesetzt wird. Diese Option ist aber eher als Hack zu betrachten.

reverse_symlinks: Beschreibt, ob für das Kernel-Image ohne Versionsnummer ein symbolischer Link (mit Versionsnummer) angelegt werden soll. Diese Option wird von einer gesetzten Umgebungsvariablen REVERSE_SYMLINK überschrieben. Weiterhin kann diese Option nicht zusammen mit no_symlinks verwendet werden, wohl aber mit link_in_boot. Wie bei der Option no_symlink besteht hierbei eine Beschränkung auf zwei Kernel-Images, wenn nicht weitere Kernel-Images von Hand gepflegt werden. Diese Option sollte nur genutzt werden, wenn es zwingend erforderlich ist, beispielsweise bei der Verwendung eines umsdos-Dateisystems auf der /boot-Partition.

patch_the_kernel: Diese Variable ist nur für Experten gedacht. Wird diese Variable auf den Wert YES gesetzt (diese Variable kann durch die Umgebungsvariable PATCH_THE_KERNEL überschrieben werden), so wird beim Erzeugen des Debian Pakets das Skript /usr/src/kernel-patches/$(architecture)/apply ausgeführt und alle installierten Kernel-Patches auf den Kernel-Source-Code angewendet. Dieser Vorgang wird bei einem Aufruf von make-kpkg clean wieder rückgängig gemacht, indem /usr/src/kernel-patches/$(architecture)/unpatch ausgeführt wird. Um Architektur-unabhängige Patches zu verwenden, kann als Architektur „all“ angegeben werden.

config_target: Beschreibt, welcher Schritt der Konfiguration durchlaufen werden soll. Der Vorgabewert ist hierbei oldconfig, der sich besonders gut für das Erstellen eines Kernels ohne viele Nachfragen eignet. Wenn die Option patch_the_kernel gesetzt ist, so werden wahrscheinlich weitere Anpassungen für die neuen Funktionen notwendig. Dann sollte diese Option auf menuconfig oder xconfig gesetzt werden.

Die Umgebungsvariable CONFIG_TARGET überschreibt den Wert in der Konfigurationsdatei. Entspricht der Wert von config_target nicht einem der Werte config, oldconfig, menuconfig oder xconfig, so wird der Wert auf oldconfig gesetzt.

use_saved_config: Wird diese Variable auf NO gesetzt, so wird die Datei .config.save ignoriert.

root_cmd: Mit dieser Variablen kann ein Kommando bestimmt werden, das es erlaubt, als Benutzer temporär die Rechte des Administrators zu erlangen. Kommandos, die dies ermöglichen, sind beispielsweise sudo oder fakeroot. Mit der Umgebungsvariablen ROOT_CMD kann dieser Wert überschrieben werden.

delete_build_link: Wird diese Variable auf YES gesetzt, so wird der Link auf die Datei /lib/modules/$VERSION/build aus dem erzeugten Debian Paket entfernt. Das Setzen der Umgebungsvariablen DELETE_BUILD_LINK hat die gleichen Auswirkungen.

do_clean: Das Setzen dieser Variable auf den Wert YES bewirkt den Aufruf von make clean, nachdem das Kernel-Paket erzeugt wurde. Alle anderen Werte verhindern den Aufruf. Die entsprechende Umgebungsvariable dafür lautet CLEAN_SOURCE.

install_vmlinuz: Mit dem Setzen dieser Variable auf YES wird zusätzlich zu dem komprimierten Kernel-Image (vmlinuz) auch ein unkomprimiertes Kernel-Image installiert.

source_clean_hook: An dieser Stelle kann vor dem Packen der Kernel-Sourcen zu einem Archiv ein Programm aufgerufen werden, das beispielsweise nicht benötigte Dateien löscht. Dies können Verzeichnisse sein, die zur Versionskontrolle benötigt werden, oder es können Dateien gelöscht werden, die zu einer nicht benötigten Architektur gehören.

Die Aktion wird nur auf den temporären Kernel-Source-Baum innerhalb von ./debian/tmp-source/usr/src/linux-source-X.X.XX angewendet.

header_clean_hook: Wie zuvor bei source_clean_hook beschrieben; diesmal jedoch ausschließlich mit Auswirkungen auf die entsprechenden Kernel-Header.

doc_clean_hook: Wie zuvor bei source_clean_hook beschrieben; diesmal jedoch ausschließlich mit Auswirkungen auf die entsprechende Kernel-Dokumentation.

extra_docs: Kann auf ein Verzeichnis zeigen, in dem zusätzliche Dokumentation liegt, die im Verzeichnis /usr/share/doc/linux-image-X.X.XX/ abgelegt werden soll. Die Dateien werden dabei nicht komprimiert, und es wird auch nicht geprüft, ob Dateien mit gleichen Namen bereits vorhanden sind. Der Pfad kann auch mit der Umgebungsvariablen EXTRA_DOCS übergeben werden.

kpkg_follow_symlinks_in_src: Nützlich für Entwickler, die Symlinks intensiv nutzen, um Kernel zu erzeugen. In den erzeugten Paketen werden die Links umgewandelt, so dass auch tatsächlich alle Dateien enthalten sind. Die entsprechende Umgebungsvariable KPKG_FOLLOW_SYM-LINKS_IN_SRC kann ebenfalls genutzt werden.

make_libc_headers: Diese Variable sollte nur vom Maintainer der libc eingesetzt werden, wenn die libc6 neu übersetzt wird. Nutzen Sie diese Variable nicht! Die entsprechende Umgebungsvariable ist MAKE_LIBC_HEADERS.

CONCURRENCY_LEVEL: Hiermit kann beeinflusst werden, wie viele Compiler-Instanzen von make über das Flag -j aufgerufen werden. Der Wert muss eine (kleine) Ganzzahl sein.

ARCH_IN_NAME: Wird diese Variable benutzt, so wird ein zusätzlicher Name für das Kernel-Paket in der Subarchitektur hinzugefügt. So können über ein kleines Skript unterschiedliche Kernel für verschiedene Architekturen erzeugt werden. Beachten Sie, dass lediglich der Paketname mit dieser Variablen verändert wird, nicht die Anordnung der Dateien in Dateisystem.

CONFDIR: Diese Variable kann den Pfad zu einem Verzeichnis enthalten, in dem sich Konfigurationsdateien (.config) für die jeweilige Architektur befinden. Die Voreinstellung zeigt auf das Verzeichnis /usr/share/kernel-package/Config.

IMAGEDIR: Wenn die erzeugten Kernel-Images nicht im Verzeichnis /boot gespeichert werden sollen, so kann mit dieser Variablen das Verzeichnis verändert werden. Benutzer von loadlin werden dies zu schätzen wissen.

MODULE_LOC: Zeigt auf ein Verzeichnis, in dem zusätzliche Module abgelegt sind. Die Voreinstellung ist /usr/src/modules.

PATCH_DIR: Zeigt auf ein Verzeichnis, in dem sich Kernel-Patches befinden, die vor dem Übersetzen des Kernels auf den Source-Code angewendet werden sollen. Diese Variable zeigt in der Voreinstellung auf /usr/src/kernel-patches/ARCHITECTURE.

ALL_PATCH_DIR: Hiermit kann ein Verzeichnis angegeben werden, in dem sich Architektur-unabhängige Kernel-Patches befinden. Die Vorgabe ist hier das Verzeichnis /usr/src/kernel-patches/all.

Die Priorisierung erfolgt in dieser Reihenfolge:

Eine Variable ist in der Datei rules gesetzt. Dies ist die Voreinstellung.

Eine Variable ist in der Datei /etc/kernel-pkg.conf gesetzt und hat damit eine höhere Wertigkeit.

Variablen können auch über die entsprechenden Umgebungsvariablen gesetzt werden. Umgebungsvariablen überschreiben die Angaben in den Konfigurationsdateien und den Voreinstellungen.

Werden make-kpkg-Optionen auf der Kommandozeile verwendet, so überschreiben diese alle vorhergehenden Definitionen.

Abschließend wird das Debian Kernel-Paket mittels make-kpkg und den gewünschten Optionen erzeugt.

Mit make-kpkg lassen sich Debian Kernel-Pakete aus dem Kernel-Quellcode (Source) erzeugen. Dabei kann sowohl die Konfiguration des Kernels während des Einsatzes von make-kpkg vorgenommen werden, als auch der eigentliche Übersetzungsvorgang (Kompilieren) des Quellcodes angestoßen werden.

make-kpkg muss immer innerhalb des Verzeichnisses aufgerufen werden, in dem sich der Quellcode des Kernels befindet. Weiterhin sind administrative Rechte notwendig. Dies kann dadurch erreicht werden, dass make-kpkg als Administrator (root) ausgeführt wird. Es ist aber auch ausreichend, das Programm fakeroot einzusetzen.

Die Syntax von make-kpkg ist recht einfach:

```
make-kpkg [options] [target [target ...]]
```

Als einfaches Beispiel für den Einsatz von make-kpkg kann folgendes Beispiel dienen, das auch als normaler Benutzer aufgerufen werden kann:

```
make-kpkg --rootcmd fakeroot kernel_image
```

Es wird ein Debian Kernel-Paket erzeugt und in dem Verzeichnis abgelegt, das dem aktuellen Verzeichnis übergeordnet ist.

```
```

--help

Gibt eine kurze Hilfe zu den Optionen von make-kpkg aus.

--revision number

Setzt die Revisionsnummer des Debian Pakets auf das angegebene Argument.

Diese Option bewirkt nur eine Änderung der Revisionsnummer während der Konfigurationsphase (also wenn die Datei stamp-configure nicht existiert). Es empfiehlt sich, in jedem Fall vor der Benutzung dieser Option make-kpkg clean

aufzurufen. Den gleichen Effekt hat das Löschen der Dateien stamp-configure und stamp-debian.

Nach den Informationen im Debian Policy Manual darf die Zeichenkette für die Version ausschließlich aus Zahlen und den Zeichen + und . bestehen und muss mindestens eine Zahl beinhalten. Die offiziellen Debian Pakete mit Kerneln weichen davon ab, diese beinhalten auch das Zeichen -.

Der voreingestellte Wert für diesen Parameter ist 1.00.Custom. Mit der Umgebungsvariablen DEBIAN_REVISION_MANDATORY kann eine Warnung ausgegeben werden, falls die Revisionsnummer weder auf der Kommandozeile noch in der Konfigurationsdatei gesetzt wird.

--append-to-version foo, --append_to_version foo

Das angegebene Argument „foo“ wird an den Wert der Variablen EXTRAVERSION angehängt, die aus den Angaben im Kernel-Makefile generiert wird. Da EXTRAVERSION ein Bestandteil der Kernel-Version ist, wird diese Angabe auch im Namen des Debian Kernel-Pakets erscheinen und muss somit auch den Richtlinien für Paketnamen entsprechen. Das bedeutet, dass ausschließlich Kleinbuchstaben sowie die Zeichen -, + und . erlaubt sind.

Wird diese Option verwendet, so wird die Umgebungsvariable APPEND_TO_VERSION überschrieben. Bei Verwendung dieser Option muss make-kpkg clean nach der Konfiguration des Kernels aufgerufen werden, da hiermit die Datei include/linux/version.h mit den gewünschten Werten erzeugt wird. Ein einfacher Aufruf von make-kpkg würde eine bestehende Datei include/linux/version.h nicht verändern und so dazu führen, dass der Kernel die Module nicht findet, da diese in einem Verzeichnis abgelegt werden, das sich von der im Kernel festgelegten Syntax unterscheidet.

Wird später noch ein Paket mit den Kernel-Modulen erzeugt, so ist dort auch die verwendete Option anzugeben, damit die Module zum Kernel passen.

--flavor foo

Dies ist eine mittlerweile überflüssige Option, die noch aus Gründen der Kompatibilität mitgeführt wird. Zukünftig sollte nur noch --append_to_version benutzt werden.

--added-modules foo, --added_modules foo

Das Argument zu dieser Option ist ein einzelnes Modul oder eine kommaseparierte Liste von Modulen, die nicht im Standard-Kernel enthalten sind. Dabei muss der komplette Pfad zu den Modulen angegeben werden, falls diese nicht im üblichen Pfad liegen (/usr/src/modules, der auch über die Variable MODULE_LOC gesetzt werden kann).

--added-patches foo, --added_patches foo

Das Argument zu dieser Option ist eine kommaseparierte Liste von Patches, die dem Kernel-Quellcode hinzugefügt werden sollen. Die Verwendung dieser Option setzt automatisch auch die Option patch_the_kernel auf YES.

Abweichend zur Behandlung von Modulen ist es bei Patches ausreichend, den Basisnamen des Patches anzugeben. Der volle Pfad ist nicht erforderlich. Für jeden Namen eines Patches in der Liste wird wie folgt nach der passenden Datei gesucht: Wird die Datei in einem der Verzeichnisse ALL_PATCH_DIR/{apply,unpatch}/ gefunden, so wird die Datei ALL_PATCH_DIR/apply/<patch_name> während des Laufs von „configure“ ausgeführt. Passend dazu wird die Datei ALL_PATCH_DIR/unpatch/<patch_name> ausgeführt, wenn ein „clean“ ausgeführt wird.

Voreingestellt ist, dass alle vorhandenen Patches angewendet werden, in dem die ausführbaren Dateien in ALL_PATCH_DIR/apply/ bearbeitet werden. Dies geschieht durch Setzen der Konfigurationsvariablen patch_the_kernel oder der Umgebungsvariablen PATCH_THE_KERNEL auf den Wert YES. Zu beachten ist, dass alle Patches wieder aus dem Quellcode entfernt werden, wenn clean aufgerufen wird. Dieses Verhalten kann verhindert werden, indem die Umgebungsvariable NO_UNPATCH_BY_DEFAULT gesetzt wird.

Für das bisher Beschriebene gilt, dass die Variable ALL_PATCH_DIR auf das Verzeichnis /usr/src/kernel-patches/ verweist.

Wird später make-kpkg clean aufgerufen, so werden alle Patches wieder entfernt. Dies kann durch Setzen der Umgebungsvariablen NO_UNPATCH_BY_DEFAULT verhindert werden.

--arch foo

Setzt die Hardware-Architektur auf den angegebenen Wert. Diese Option wird nur benötigt, wenn der Kernel nicht auf der Zielplattform erzeugt wird (Cross-Compiling). Derselbe Effekt wird durch Setzen der Variablen KPKG_ARCH auf die gewünschte Zielarchitektur erreicht. Normalerweise wird die Architektur automatisch ermittelt.

--cross-compile foo, --cross_compile foo

Sollte ebenfalls gesetzt werden, wenn der Kernel für eine fremde Hardware-Architektur erzeugt wird. Die Umgebungsvariable hierfür lautet CROSS_COMPILE.

--subarch foo

Auf einigen Hardware-Plattformen (beispielsweise Alpha und m68k) müssen unterschiedliche Kernel für die verschiedenen Prozessoren erzeugt werden. Die dieser Option entsprechende Umgebungsvariable lautet KPKG_SUBARCH.

--arch-in-name, --arch_in_name

Diese Option vergibt einen erweiterten Namen für das Kernel-Image-Paket, so dass in einem Skript mehrere Images für Sub-Architekturen erzeugt werden können. Dies kann auch durch Setzen der Variablen ARCH_IN_NAME erreicht werden. Dies beeinflusst lediglich den Namen des Pakets, nicht das Verzeichnis, in dem die passenden Module abgelegt werden.

--pgpsign name

Setzt die Zeichenkette, mit der die Changes-Dateien für externe Module digital signiert werden. Diese Option überschreibt die Werte, die in /etc/kernel-pkg.conf oder auch ~/.kernel-pkg.conf angegeben sind.

--config target

Ändert die Einstellung für die Kernel-Konfiguration. Normalerweise wird „oldconfig“ benutzt, alternativ kann hier auch „config“, „menuconfig“, „xconfig“, „old“, „menu“ oder auch „x“ angegeben werden.

Sinnvoll ist diese Option im Zusammenhang mit PATCH_THE_KERNEL, wenn von den eingesetzten Patches neue Konfigurationsoptionen hinzugefügt werden.

--targets

Zeigt eine Liste der bekannten „Targets“ (Ziele, hier als Zielpakete zu verstehen) an. Mehr zu den Targets später.

--noexec

Übergibt die Option -n an das Kommando make, so dass keine Kommandos ausgeführt werden. Es wird lediglich angezeigt, was passieren würde.

--initrd

Wenn make-kpkg die Option linux-image übergeben wurde, so werden durch diese Option alle zusätzlichen Aktionen aufgerufen, die notwendig sind, um eine passende Init-RAM-Disk zu erzeugen.

Hierbei ist zu beachten, dass ein zusätzlicher Patch benötigt wird, der bereits in den Debian-Kernel-Sourcen enthalten ist. Wird dieser „cramfs“-Patch nicht verwendet, so muss die Konfiguration von mkinird so angepasst werden, dass cramfs nicht benutzt wird.

--zimage

Erzeugt einen zImage-Kernel statt eines bzImage-Kernels.

--bzimage

Erzeugt einen bzImage-Kernel.

--rootcmd foo

Kommando, mit dem auf dem System besondere Zugriffsrechte erreicht werden können. Dies können beispielsweise sudo oder fakeroot sein. Diese Option wird an das Programm dpkg-buildpackage weitergereicht.

--us

Diese Option wird an dpkg-buildpackage weitergereicht; der Quellcode wird nicht digital signiert.

--uc

Diese Option wird an dpkg-buildpackage weitergereicht; die Datei changelog wird nicht digital signiert.

Alle genannten Optionen können so lange verkürzt werden, bis diese nicht mehr eindeutig sind. Weiterhin ist es erlaubt, ein oder zwei Minuszeichen (- oder --) voranzustellen. Argumente können von den Optionen durch ein Leerzeichen oder ein Gleichheitszeichen getrennt sein.

clean

Löscht alle nicht benötigten Dateien aus dem Kernel-Quellcode, die zuvor über das Target „build“ erzeugt worden sind, und führt dazu das Kommando make distclean aus.

buildpackage

Führt die Targets „clean“ und „binary“ aus und erzeugt so ein Kernel-Paket.

binary

Erzeugt alle vier möglichen Debian Kernel-Pakete, indem die Targets kernel_source, kernel_headers, kernel_doc und kernel_image ausgeführt werden.

kernel_source

Erzeugt ein Debian Paket mit dem Kernel-Quellcode. Zeigt die Umgebungsvariable SOURCE_CLEAN_HOOK auf eine ausführbare Datei, so wird diese im Quellcode-Verzeichnis des Kernels ausgeführt, bevor das Paket erstellt wird. Auf diesem Weg können nicht benötigte Dateien und Verzeichnisse entfernt werden (beispielsweise Architektur-abhängige Dateien oder auch Verzeichnisse, die der Versionskontrolle dienen (find . -type d -name CVS -prune -exec rm -rf {})).

Diese Option wirkt sich ausschließlich auf den Kernel-Quellcode aus, der zu einem Debian-Paket gepackt wird. Ähnliche Auswirkungen haben die Umgebungsvariablen HEADER_CLEAN_HOOK und DOC_CLEAN_HOOK: Es wird ein Programm ausgeführt, das die Kernel-Header- bzw. die Kernel-Dokumentationsverzeichnisse bearbeitet.

kernel_headers

Erzeugt ein Paket mit den Kernel-Headern.

kernel_doc

Erzeugt ein Paket mit der Dokumentation für den Kern.

kernel_image

Erzeugt ein Debian Paket mit dem Linux-Kernel, der aus dem Quellcode übersetzt wird. Ist noch keine Konfigurationsdatei für den Kern (.config) vorhanden, so wird eine Konfiguration benutzt, die der der Debian Bootfloppies sehr ähnlich ist.

Das Paket wird so erstellt dass, abhängig von der Hardware-Plattform, der entsprechende Bootloader aufgerufen wird, so dass der neue Kernel auch zuverlässig gestartet wird. Als Bootloader kommt dabei LILO, SILO, QUIK, VMELILO, ZIPL, yaboot, PALO oder auch GRUB zum Einsatz. Während der Installation des Pakets wird dem Administrator angeboten, eine Bootdiskette zu erzeugen. Diese wird formatiert, falls dies notwendig ist.

build

Hiermit wird lediglich der Compiler gestartet und der Kernel aus dem Source-Code übersetzt. Dieses Target wird vom Target kernel_image benutzt.

modules

Erzeugt alle Module und die entsprechenden Debian Pakete, die sehr eng von einer passenden Kernel-Version abhängen. Hierbei wird vorausgesetzt, dass alle Module im Verzeichnis /usr/src/modules abgelegt sind. Es wird ein Debian Modul-Paket erzeugt, eine entsprechende komprimierte tar-Datei, eine komprimierte diff-Datei. Weiterhin werden die MD5-Checksummen in der changes-Datei aktualisiert (mittels dpkg-genchanges). Die Datei wird mit der gleichen Identität digital signiert, mit der auch das

Kernel-Paket signiert wurde. Diese Option erlaubt einem Maintainer, das Paket direkt in die Debian Archive zu kopieren.

modules_config

Konfiguriert alle Module, die sich unter /usr/src/modules befinden.

modules_image

Erzeugt Pakete zu den Modulen unter /usr/src/modules, es werden aber keine Quellcode-Pakete oder Diff-Dateien erzeugt. Weiterhin wird die Changes-Datei nicht digital signiert.

modules_clean

Löscht alle nicht benötigten Dateien im Verzeichnis /usr/src/modules.

configure

Diese Option führt ein make configure aus, genauer genommen das mittels --config angegebene config_target. Es ist voreingestellt auf oldconfig. Dies bietet die Möglichkeit, von make config erzeugte Dateien anzupassen, so dass diese später nicht von make-kpkg verändert werden.

debian

Erzeugt das Verzeichnis debian/ innerhalb des Quellcode-Verzeichnisses und wendet vorhandene Patches auf den Quellcode an.

libc-kheaders

Eine spezielle Option für den Betreuer der C-Bibliotheken.

make-kpkg verwendet die folgenden Umgebungsvariablen, falls diese gesetzt sind: DEBIAN_REVISION_MANDATORY, APPEND_TO_VERSION, VERSION_H_OK, PATCH_THE_KERNEL, NO_UNPATCH_BY_DEFAULT, KPKG_ARCH, CROSS_COMPILE, KPKG_SUBARCH, ARCH_IN_NAME, INITRD, SOURCE_CLEAN_HOOK, MODULE_LOC, INITRD_OK

Neben den Optionen, die zur Laufzeit übergeben werden, kann make-kpkg auch über die Konfigurationsdateien /etc/kernel-pkg.conf (systemweite Konfiguration) oder ~/.kernel-pkg.conf (benutzerbezogene Konfiguration) gesteuert werden.

Die dem Debian Paket mitgelieferte Konfiguration erlaubt es, den Namen und die E-Mail-Adresse zu überschreiben. Die Konfigurationsdateien sind als Teile eines Makefiles zu

verstehen, alle in einem solchen Makefile erlaubten Funktionen können hier aufgerufen werden.

Mit der Debian Distribution werden neben den Sourcen für die in Debian selbst verwendeten Kernel auch eine Reihe Kernel-Patches geliefert, die in eigene Kernel integriert werden können. Ein guter Einstieg für die ersten Versuche ist der kernel-patch-debianlogo; dieser Patch ersetzt den beim Systemstart angezeigten Pinguin durch ein Debian Logo. Bei der ersten Installation eines solchen Debian Kernel-Patches wird in /usr/src/ das Verzeichnis kernel-patches angelegt.

Die Installation eines solchen Kernel-Patch-Pakets wendet den Patch noch nicht an; dies geschieht erst mit dem Aufruf von make-kpkg:

```
bash$ fakeroot make-kpkg --append-to-version=030825 \  --added-patches=debianlogo kernel_image modules_image
```

Das Kommando fakeroot wird dabei benutzt, um beschränkte administrative Rechte auch als normaler Benutzer zu erlangen.

Wenn weitere Patches in den Kernel eingespielt werden sollen, können diese durch Kommata getrennt an der Option --added-patches angegeben werden.

Normalerweise benutzt make-kpkg „make oldconfig“, nachdem die Patches angewendet wurden. Wenn von den Patches neue Kernel-Optionen benötigt werden, so werden diese abgefragt. Alternativ kann auch die Option --config=menuconfig benutzt werden, um die komplette Konfiguration zu überprüfen.

In einigen Fällen kann es notwendig sein, den Linux-Kernel an die eigenen Bedürfnisse anzupassen. Debian GNU/Linux installiert einen Standard-Kernel, der in den meisten Fällen ausreichend ist. Zusätzliche Treiber können über Module im laufenden Betrieb hinzugeladen werden.

Um aus den Kernel-Sourcen einen lauffähigen Kernel zu erzeugen, müssen die Quelltexte mittels eines passenden Compilers übersetzt werden. Der komplette Linux-Kernel wurde in der Programmiersprache C geschrieben - mit Ausnahme einiger ganz weniger Zeilen, die aus Geschwindigkeitsgründen in der Maschinensprache Assembler geschrieben wurden. Übrigens steht unter Linux der gcc (GNU Compiler Collection) - eine freie Implementierung eines C-Compilers - zur Verfügung.

Es werden folgende Pakete benötigt, um einen Kernel selbst zu erzeugen:

binutils - Der GNU-Assembler, -Linker und einige Zusatzprogramme.

libc6-dev - GNU C-Bibliothek, Entwicklerpaket.

gcc - Der eigentliche GNU (EGCS) C-Compiler.

make - GNU-Version von „make“.

bin86 - Ein 16-Bit-Assembler.

Nützlich, aber nicht zwingend erforderlich, sind weiterhin folgende Pakete:

libncurses5-dev - Entwickler-Bibliotheken und Dokumentation für ncurses.

tkstep8.0-dev - NeXTStep ähnliche Version des Tk-Toolkits. (oder tk8.0-dev oder tkstep4.2-dev oder tk4.2-dev).

kernel-package - Debian Linux-Kernel-Paket-Skripte.

Soll ein individuell auf das System angepasster Kernel installiert werden, so ist es zuerst nötig, die Kernel-Sourcen (Quellcode) zu installieren. Für den ersten Versuch ist es ratsam, die gleiche Kernel-Version zu installieren, die auch schon vom Installationsprogramm installiert wurde. Der Befehl

```
cat /proc/version
```

gibt Ihnen die installierte Version aus. Installieren Sie nun (mit dselect, dpkg oder apt) die Sourcen zu diesem Kernel.

Die Kernel-Sourcen werden vom Installationsprogramm unter /usr/src/ als Datei linux-source-2.x.x.tgz abgelegt und müssen noch von Hand entpackt werden. Dies wird mit dem Befehl

```
tar xvfz linux-source-2.x.x.tgz
```

erreicht. Die entpackten Sourcen finden sich nun in dem Verzeichnis linux-source-2.x.x/.

Dass die Debian Quellcodes des Kernels als ein Archiv installiert werden, mag auf den ersten Blick seltsam erscheinen, da ja durch das Entpacken sogar noch mehr Speicherplatz belegt wird. Dies hat aber bei näherer Betrachtung einen guten Grund. Das Debian Paketmanagement führt eine Datenbank über alle installierten Pakete und die dazugehörigen Dateien. Würde nun ein Kernel-Quellcode-Paket entpackt installiert werden, so würden die Anwender direkt mit diesen Dateien arbeiten und diese eventuell gar verändern. Damit wäre die zentrale Kontrolle umgangen.

Auf den Servern finden sich die aktuellen Archive des Linux-Kernels auch in einer mit bzip2 (anstatt mit gzip) komprimierten Form. Diese Archive sind noch einmal ein wenig kleiner, da bzip2 bessere Kompressionsalgorithmen verwendet. Die so gepackten Archive enden auf .bz2. Diese Archive können ebenfalls mit tar entpackt werden; hierzu ist aber die Option -j statt -z anzugeben. Beachten Sie: Es sind zwischenzeitlich einige Versionen von tar im Umlauf gewesen, bei denen für bzip2-komprimierte Archive die Option -I zu verwenden ist.

Wechseln Sie nun in das Verzeichnis, in dem Sie vorher die Kernel-Sourcen entpackt haben. Mit dem Kommando make config erzeugen Sie die benötigte Konfigurationsdatei. Sie können hier den Kernel individuell an Ihre eigene Hardware anpassen. Bei vielen Optionen haben Sie die Möglichkeit, zwischen fest im Kernel integrierten Treibern „ * “ oder Modulen „ M “ zu wählen. Achten Sie darauf, nur die nötigen Treiber fest in den Kernel einzubinden, da sonst der Kernel zu groß wird und nicht mehr von lilo geladen werden kann. Gute Kandidaten auf der Liste der Module sind alle Treiber, die nicht zum unmittelbaren Systemstart benötigt werden (Netzwerk, Bandlaufwerke).

Neben make config stehen Ihnen alternativ die Befehle make menuconfig mit einer textbasierten Oberfläche ähnlich wie bei dselect sowie make xconfig unter X11 zur Verfügung. Diese erleichtern kleine Änderungen am Kernel sehr, da die gewünschten Einstellungen direkt anzuwählen sind.

Wenn Sie die Option menuconfig benutzen wollen, muss das Paket libncurses5-dev installiert sein; dieses stellt die notwendigen Funktionen für die textbasierte Oberfläche zur Verfügung.

Nachdem Sie die gewünschten Einstellungen gemacht haben, werden mit dem Befehl make dep die Abhängigkeiten geprüft. Ein make clean räumt noch übrig gebliebene Dateien von der Festplatte. Einen neuen Kernel erzeugen Sie mit dem Befehl make bzImage. Zum Übersetzen des Kernels müssen neben einem C-Compiler auch die Pakete bin86 und natürlich make (das haben Sie aber sicher vorher schon bemerkt) installiert sein. Nach einiger Zeit (wenn alles ohne Fehlermeldungen über die Bühne gegangen ist) finden Sie den neuen Kernel unter /usr/src/linux-source-2.x.x/arch/i386/boot/. Dieser muss nun noch an die passende Stelle (bei Debian üblicherweise /boot/) kopiert werden. Abschließend ist die Konfiguration von lilo zu prüfen und ggf. anzupassen.

Statt make bzImage können Sie auch make bzlilo verwenden: Dieser Befehl kopiert den Kernel nach /vmlinuz und benennt vorher den schon vorhandenen Kernel in vmlinuz.old um. Danach wird automatisch lilo aufgerufen, und somit steht der Kernel dann ab dem nächsten Neustart zur Verfügung.

Sie können nun die Treiber, die Sie bei der Konfiguration als Module ausgewählt haben, übersetzen. Dies geschieht mit dem Befehl make modules. Nach dem Übersetzen der Module werden diese mit make modules_install an den richtigen Ort kopiert.

Benutzen Sie das Zeichen „ & “, um mehrere Kommandos nacheinander auszuführen. Sie müssen so nicht die einzelnen Schritte beim Übersetzen eines neuen Kernels abwarten. make dep && make clean && make bzlilo && make modules && make modules_install erledigt einen kompletten Durchlauf ohne Pause.

Probieren Sie einfach einmal, den Kernel mit der Option -s zu übersetzen, also: make -s zImage oder make -s bzImage. Bei einem so übersetzten Kernel werden lediglich Warnungen und Fehlermeldungen des Kernels beim Systemstart ausgegeben. Alle Ausgaben der Treiber werden unterdrückt.

Sie können das Übersetzen des Kernels beschleunigen, indem Sie den Parameter -j # einfügen, wobei # für eine (fast beliebige) Zahl steht. Mit diesem Parameter werden, entsprechend der angegebenen Zahl, mehrere Prozesse gestartet und Teile des Kernels gleichzeitig übersetzt. Sinnvolle Werte für die Anzahl der Prozesse sind in erster Linie vom Ausbau des Hauptspeichers (RAM) abhängig. Auf Systemen mit mehreren Prozessoren wirkt sich dies natürlich auch positiv aus. Bedenken Sie bitte, dass zu hoch gewählte Werte zum Auslagern (Swappen) führen und den Vorgang merklich verlangsamen.

Um festzustellen, welcher Wert sinnvoll ist, benutzen Sie das Kommando time make -j 10 bzImage und variieren den Wert für die Anzahl der Prozesse.

Sollten Sie nach dem Übersetzen der Module mit make modules; make modules_install Probleme haben, diese zu laden, liegt dies wahrscheinlich daran, dass die Datei modules.dep, in der die Abhängigkeiten (dependencies) beschrieben sind, nicht aktuell ist. Es ist nicht nötig, in dieser Datei irgendetwas von Hand zu ändern: Der Befehl depmod -a 2.6.x erstellt eine aktuelle Datei für Sie, wobei 2.6.x der neuen Kernel-Version entspricht.

Wenn Sie viele verschiedene Kernel-Versionen auf der Festplatte halten, kann es vorkommen, dass Fehlermeldungen in der Form Warning: /boot/System.map has an incorrect kernel version. erscheinen. Neben der Möglichkeit, je eine Version der System.map in /boot/ und eine weitere in /usr/src/linux/ zu halten (was maximal zwei Versionen erlaubt), bietet Debian GNU/Linux sozusagen eine „Komplettlösung“.

Das Skript /etc/init.d/klogd startet beim Systemstart den klogd. Sie können am Anfang dieses Skriptes in der Variablen KLOGD als Parameter -k /boot/System.map-$(uname -r) angeben. Eine entsprechende Zeile findet sich

bereits vorbereitet in dem Skript. So wird je nach verwendeter Kernel-Version eine passende System.map aus /boot/ geladen. Diese müssen Sie nach dem Übersetzen des Kernels in das Verzeichnis /boot/ kopieren und passend zur Kernel-Version benennen. Am einfachsten können Sie das mit folgendem Kommando erledigen: cp /usr/src/linux/System.map /boot/System.map-`uname -r`.

Hier sehen Sie ein kleines Skript, das in /boot/ nach Kerneln sucht und eine passende lilo.conf erstellt, mit dem neuesten Kernel als Standardkernel.

```
#!/bin/bash  umask 772 kernel_dir=/boot  # lilo assumes the default image is the first one in lilo.conf, so # we sort the kernel images backwards, hence the highest-version'd kernel # will be the default. images=`cd $kernel_dir >> ls -1 vmlinuz-* \  | egrep "vmlinuz-([0-9]+).([0-9]+).([0-9]+)[^-]*$" \  | sort -rn`  cp -f /etc/lilo.conf.static /tmp/lilo.conf  # three lines per entry, 3 x 19 images = 57 ( for img in $images ; do    label=`echo $img | sed 's/vmlinuz/linux/ ; s/-//g ; s/\.//g'` echo "image=$kernel_dir/$img"    echo "label=$label"    echo ""  done ) | head -57 >> /tmp/lilo.conf  if /sbin/lilo -C /tmp/lilo.conf ; then   mv -f /etc/lilo.conf /etc/lilo.conf.last   cp -f /tmp/lilo.conf /etc/lilo.conf   echo successfully installed new bootloader.   rm -f /tmp/lilo.conf   exit 0 else echo eek, lilo barfed   rm -f /tmp/lilo.conf   exit 1 fi
```

Wenn Sie einen der neueren Kernel (2.3.x oder 2.4.x) mit Debian GNU/Linux 2.2 verwenden möchten, sollten Sie folgende Zeile in die Datei /etc/fstab einfügen:

```
none/var/shm    shm  defaults     0       0
```

Ab der Kernel-Version 2.3.51 wurde das „Shared-Memory“-Dateisystem eingeführt.

5.2 Debian Kernel-Pakete erzeugen

5.3 Debian Pakete anpassen

5.2 Debian Kernel-Pakete erzeugen 5.4 Debian Pakete - Aufbau der Sourcen

5.3.1 apt-build

Das Anpassen von vorhandenen Debian Paketen kann aus verschiedenen Gründen sinnvoll sein. Es kann vorkommen, dass ein Paket für die verwendete Release-Version von Debian (noch) nicht verfügbar ist, dass andere Optionen zur Übersetzung mit dem Compiler verwendet werden sollen oder dass das Paket auf einer anderen Hardware-Architektur benutzt werden soll. Auch Anpassungen an den Init-Skripten sind unter Umständen sinnvoll.

Zunächst sind von dem gewünschten Paket die entsprechenden Dateien mit den Endungen .dsc (Description/Beschreibung), .diff.gz (die für Debian vorgenommenen Änderungen) und .tar.gz (die eigentlichen Sourcen des Programms) vom Debian FTP-Server zu holen. Für das Paket gato „gato“ finden Sie auf dem Debian FTP-Server (ftp://ftp.de.debian.org/) im Verzeichnis debian/pool/main/g/gato die Dateien gato_0.6.4-2.diff.gz, gato_0.6.4-2.dsc und gato_0.6.4.orig.tar.gz.

Im nächsten Schritt sind die Sourcen des Pakets zu entpacken. Dies geschieht nicht (wie sonst bei Source-Paketen üblich) mit dem Kommando tar (dieses würde lediglich die Original-Sourcen entpacken), sondern mit dem Kommando dpkg-source -x *.dsc. Hierbei werden zusätzlich die für das Debian Paket vorgenommenen Veränderungen (Patches) und alle Dateien, die sich im Verzeichnis debian/ befinden, entpackt. Die Sourcen befinden sich in einem neuen Verzeichnis, das sich aus dem Paketnamen und der Programmversion zusammensetzt. Wechseln Sie in dieses Verzeichnis.

Nun können die gewünschten Veränderungen an den Sourcen vorgenommen werden. Wenn Sie Veränderungen oder Ergänzungen an Dateien im Verzeichnis debian/ vornehmen, so ist darauf zu achten, dass unter Umständen auch Veränderungen oder Anpassungen in der Datei debian/rules notwendig sind. Mehr zu den Funktionen der einzelnen Dateien erfahren Sie im nächsten Abschnitt. Abschließend kann das Paket mit dpkg-buildpackage -us -uc -rfakeroot erstellt werden. Weitere Optionen von dpkg-buildpackage sind ebenfalls im nächsten Abschnitt beschrieben.

Der im vorigen Abschnitt beschriebene Weg, ein einzelnes Paket zu entpacken, anzupassen und zu übersetzen, ist sicherlich sinnvoll, wenn nur wenige Pakete so optimiert werden sollen. Bei der Gentoo Linux Distribution ist es üblich, jedes Software-Paket selbst zu übersetzen. Dort wird bereits von Anfang an festgelegt, auf welchem Prozessor das zukünftige System laufen soll, und alle Pakete werden mit den entsprechenden Optimierungen übersetzt.

Kritiker führen immer wieder gerne an, dass eine solche Optimierung auf einem Debian System mit einem hohen Aufwand verbunden ist. Schauen wir uns doch einmal an, wie hoch dieser Aufwand tatsächlich ist.

Um bereits für Debian bereitgestellte Software-Pakete erneut auf einem System zu übersetzen, dient das Programm apt-build aus dem gleichnamigen Paket. Dieses sollte spätestens jetzt installiert werden. apt-build nutzt die Konfigurationsdatei /etc/apt/apt-build.conf. Diese hat folgenden Inhalt, die Werte werden während der Installation des Paketes abgefragt:

```
build-dir = /var/cache/apt-build/build repository-dir = /var/cache/apt-build/repository Olevel = -O3 march = -march=pentium2 mcpu = -mcpu=pentium2 options =
```

Die in der Konfigurationsdatei enthaltenen Parameter gliedern sich in die von apt-build verwendeten Verzeichnisse und die Optionen, die dem GNU C Compiler übergeben werden. Hier sind die für den eingesetzten Prozessortyp passenden Optionen anzugeben, diese werden für alle Pakete, die mittels apt-build erzeugt werden, eingesetzt.

Des Weiteren wird ein Eintrag in der Datei /etc/apt/sources.list benötigt, welcher auf ein entsprechendes Repository zeigt, aus dem die Quellen der Debian Pakete bezogen werden können. Ein solcher Eintrag kann beispielhaft wie folgt aussehen:

```
deb-src ftp://ftp.debian.org/debian/ stable main contrib
```

Ein Eintrag für das apt-build Repository mit den optimierten Paketen auf dem lokalen System ist während der Installation des Paketes apt-build auf Wunsch bereits der Datei /etc/apt/sources.list hinzugefügt worden. Diese sieht folgendermaßen aus:

```
deb file:/var/cache/apt-build/repository apt-build main
```

Damit ist die Konfiguration von apt-build abgeschlossen. Der Aufruf von apt-build muss mit Administratorrechten erfolgen. apt-build verwendet die folgenden Optionen:

update

Aktualisiert die Liste der installierbaren Pakete.

upgrade

Aktualisiert die bereits installierten Pakete.

install

Holt die Debian Quellcode-Pakete zu dem gewünschten Paket, übersetzt die Software mit den eingestellten Optionen, erzeugt ein Debian Paket und installiert das erzeugte Paket. Ein Beispiel:

```
apt-build install memstat -----> Installing build dependencies (for memstat) <----- Reading Package Lists... Done Building Dependency Tree... Done 0 packages upgraded, 0 newly installed, 0 to remove and 0 not upgraded. -----> Downloading memstat source (memstat) <----- Reading Package Lists... Done Building Dependency Tree... Done Need to get 22.4kB of source archives. Get:1 ftp://ftp.debian.org/main memstat 0.4-1 (dsc) [482B] Get:2 ftp://ftp.debian.org/main memstat 0.4-1 (tar) [21.9kB] Fetched 22.4kB in 0s (322kB/s) dpkg-source: extracting memstat in memstat-0.4 -----> Building memstat <----- ... dpkg-genchanges: binary-only upload - not including any source code dpkg-buildpackage: binary only upload (no source included) -----> Moving packages to repository <----- -----> Updating repository <----- Using: -O3 -mcpu=pentium2 -march=pentium2 ... Reading Package Lists... Done Building Dependency Tree... Done Reading Package Lists... Done Building Dependency Tree... Done The following NEW packages will be installed:   memstat ...
```

source

Holt und entpackt die Quellcode-Dateien zu einem Paket.

apt-build wird also eingesetzt, um eigene, angepasste Pakete aus bestehenden Debian Paketen zu erzeugen. Wird jedoch ein Upgrade des Systems durchgeführt (apt-get dist-upgrade), so wird das mittels apt-build erzeugte, angepasste Paket durch das offizielle Debian Paket ersetzt. Dies kann verhindert werden, indem die Priorität von Paketen, die mit apt-build erzeugt wurden, erhöht wird.

Um die Priorität solcher mittels apt-build erzeugter Pakete zu erhöhen, muss die Datei /etc/apt/preferences wie folgt angepasst werden:

```
Package: *    Pin: release o=apt-build       Pin-Priority: 990
```

Durch diese Änderung werden selbst erzeugte Pakete bei einem Upgrade nicht mehr durch neuere, offizielle Debian Pakete überschrieben. Mit dem Kommando apt-cache policy lässt sich dies überprüfen.

```
fr@wasabi]~ $ apt-cache policy  Package Files:  100
/var/lib/dpkg/status      release a=now  500 http://ftp2.de.debian.org
sid/non-free Packages      release o=Debian,a=unstable,l=Debian,c=non-
free      origin ftp2.de.debian.org  500 http://ftp2.de.debian.org
sid/contrib Packages      release
o=Debian,a=unstable,l=Debian,c=contrib      origin ftp2.de.debian.org
500 http://ftp2.de.debian.org sid/main Packages      release
o=Debian,a=unstable,l=Debian,c=main      origin ftp2.de.debian.org
990 file: apt-build/main Packages      release o=apt-build,a=apt-
build,l=apt-build,c=main Pinned Packages:
```

Dies ist eine einfache und mit dem Debian Paketmanagement kompatible Methode, eigene Pakete zu verwalten. Zu beachten ist dabei, dass Updates solcher Pakete auf neue Versionen oder auf Versionen, in denen Sicherheitslücken beseitigt wurden, selbst

durchzuführen sind. Werden Bibliotheken auf diese Weise selbst verwaltet, so kann dies dazuführen, dass Pakete, die auf diesen Bibliotheken aufsetzen, eventuell nicht aktualisiert werden können. In diesem Fall sind die Bibliotheken auf den neuesten Versionsstand nachzuziehen.

5.3 Debian Pakete anpassen

5.2 Debian Kernel-Pakete erzeugen 5.4 Debian Pakete - Aufbau der Sourcen

5.4 Debian Pakete - Aufbau der Sourcen

5.3 Debian Pakete anpassen 5.5 Erstellen, prüfen und verwalten von Debian Paketen

Wenn Sie ein komplett neues Debian Paket erzeugen wollen, sei es, weil zu den Sourcen noch kein Paket vorhanden ist, oder sei es, weil Sie selbst eine Software entwickelt haben, die Sie als Debian Paket zur Verfügung stellen wollen, so stehen Ihnen auch dafür unter Debian einige Hilfsmittel zur Verfügung.

Zunächst sollten Sie sich die Sourcen des gewünschten Pakets besorgen und ein Unterverzeichnis für das neue Projekt erzeugen. Am Beispiel des Programms dtcltiny, einem Client zur Steuerung von digitalen Modellbahnen, sollen die einzelnen Schritte vorgestellt werden. Die Sourcen dieses Programms finden Sie unter sourceforge.net/projects/dtcltiny/. Natürlich können Sie auch ein beliebiges anderes Programm benutzen, von dem Sie ein Debian Paket erzeugen möchten. Sie müssen dann aber ggf. an einigen Stellen etwas Kreativität einsetzen, da die hier gezeigten Beispiele sich nicht 1:1 übertragen lassen. Es ist auch nicht zwingend notwendig, dass die Sourcen zu einem Programm verfügbar sind. Selbstverständlich können Sie auch Debian Pakete von Programmen im Binär-Format erstellen oder auch Dokumentation, beispielsweise als PDF, zu einem Paket packen.

Wechseln Sie nun in das Unterverzeichnis und entpacken Sie die Sourcen des Programms:

```
fr@debian:~/Daten/debian-pakete$ mkdir dtcltiny
fr@debian:~/Daten/debian-pakete$ cd dtcltiny
fr@debian:~/Daten/debian-pakete/dtcltiny$ tar xvfz ../../../dtcltiny-
0.3.4.tgz  dtcltiny-0.3.4/ dtcltiny-0.3.4/Makefile.am dtcltiny-
0.3.4/acinclude.m4 dtcltiny-0.3.4/Makefile.dist  ...  dtcltiny-
0.3.4/dtcltiny/InfoPort.moc.o dtcltiny-0.3.4/dtcltiny/.libs/ dtcltiny-
0.3.4/configure dtcltiny-0.3.4/stamp-h
```

Wichtig ist hierbei, dass das entstehende Unterverzeichnis schon aus dem Programmnamen und der Versionsnummer besteht. Sollte dies nicht so sein, muss das Verzeichnis entsprechend umbenannt werden. Im nächsten Schritt wechseln Sie in das Verzeichnis und rufen das Programm dh_make mit der Option -f auf:

```
fr@debian:~/Daten/debian-pakete/dtcltiny$ cd dtcltiny-0.3.4/
fr@debian:~/Daten/debian-pakete/dtcltiny/dtcltiny-0.3.4$ dh_make -f
Type of package: single binary, multiple binary, or library? [s/m/l] s
Maintainer name : Frank Ronneburg Email-Address    :
fr@debiananwenderhandbuch.de  Date           : Tue, 13 Nov 2001
21:56:17 +0100 Package Name    : dtcltiny Version        : 0.3.4 Type of
Package : Single Hit <enter> to confirm:  Done. Please edit the files in
the debian/ subdirectory now. dtcltiny uses a configure script, so you
probably don't have to edit the Makefiles.
```

dh_make erzeugt ein Unterverzeichnis debian/, in dem sich Vorlagen für alle notwendigen Dateien zur Erzeugung eines Debian Pakets befinden. Wenn noch kein Archiv mit den Original-Quellcodes (.tar.gz) vorliegt, beispielsweise weil nur ein einfaches Skript in ein Paket verpackt werden soll, so muss die Option --native angegeben werden. dh_make bietet die Möglichkeit, verschiedene Pakettypen zu erzeugen; für ein einfaches Paket wie in diesem Beispiel ist „single binary“ die richtige Wahl. Abschließend wird noch der Hinweis ausgegeben, dass sich im Unterverzeichnis debian/ einige Dateien befinden, auf die Sie noch einen Blick werfen sollten. Weiterhin wird darauf hingewiesen, dass in den Sourcen von dtcltiny ein configure-Skript enthalten ist und somit Änderungen am Makefile wahrscheinlich unnötig sind. Neben dem Verzeichnis debian/ wurde im darüber liegenden Verzeichnis noch eine Kopie des gesamten Verzeichnisbaums angelegt. Dieser wurde um die Endung .orig ergänzt.

Bei genauer Betrachtung des Beispiels ist vielleicht aufgefallen, dass die E-Mail-Adresse nicht korrekt erkannt wurde; ein nettes Beispiel also, so kommen wir nicht um einen Blick ins Verzeichnis debian/ herum ;-).

Die Datei README.Debian enthält Informationen zu den Besonderheiten dieses Debian Pakets. Wenn Veränderungen oder Ergänzungen gegenüber dem Source-Paket vorgenommen werden, so sollten diese hier vermerkt werden. Achten Sie darauf, in der letzten Zeile die E-Mail-Adresse anzupassen. Eine bereits veränderte Version könnte wie folgt aussehen:

dtcltiny for Debian ********** Ergänzungen: - dtcltiny in dtcltiny.bin umbenannt. - dtcltiny ist ein wrapper, wenn vorhanden, wird eine Konfigdatei .dtcltiny.data aus dem Homeverzeichnis des Benutzers geladen. - In /usr/share/doc/dtcltiny/ findet sich eine Beispielkonfigurationsdatei (SAMPLE.dtcltiny.data) -- Frank Ronneburg <fr@debiananwenderhandbuch.de>, Tue, 6 Nov 2001 22:29:53 +0100

Wenn Sie ein offizielles Debian Paket erstellen oder auch nur das Paket im Netz verfügbar machen wollen, so sollte die komplette Dokumentation, also auch Kommentare, in englischer Sprache erstellt werden. Bei Paketen für den privaten Gebrauch bleibt Ihnen die Wahl der Sprache natürlich überlassen.

Wenn keine weiteren Veränderungen an den Sourcen, der Dokumentation oder den Skripten des Programms notwendig waren, so kann diese Datei auch einfach gelöscht werden.

Diese Datei enthält lediglich den Paketnamen, den Bereich sowie die Priorität des Pakets. Neuere Versionen von dh_make erzeugen diese Datei nicht mehr.

dclttiny_0.3.4-1_i386.deb misc optional

In dieser Datei werden Veränderungen zwischen den einzelnen Paketversionen (changes) dokumentiert. Sie sollten hier bei einer aktualisierten Version des Pakets die Veränderungen gegenüber der vorhergehenden Version dokumentieren.

dtcltiny (0.3.4-1) unstable; urgency=low * Initial Release. -- Frank Ronneburg <fr@debiananwenderhandbuch.de> Tue, 6 Nov 2001 22:29:53 +0100

In dieser Datei finden Sie Hinweise zum Copyright dieses Pakets. Es können die gesamten Copyright-Informationen hier aufgeführt werden. Bei Paketen, die einer weit verbreiteten Lizenz unterliegen, reicht auch ein kurzer Hinweis, wo die Lizenz zu finden ist.

Zu diesem Punkt gibt es immer wieder Diskussionen unter den Debian Entwicklern. Insbesondere bei Paketen, die der GPL unterliegen, gehen die Meinungen auseinander. Grundsätzlich sollte, um Missverständnisse zu vermeiden, zu jedem Paket die Lizenz mitgeliefert werden. Die GPL hat in der aktuellen Version 2 eine Größe von ca. 18 Kbyte. Das ist nicht sehr viel; wenn man die aber mit einigen tausend Paketen multipliziert, so kommt doch eine beachtliche Menge an Speicherplatz zusammen. Es wurde vereinbart, im Verzeichnis /usr/share/common-licenses/ jeweils eine Version der gebräuchlichsten Lizenzen abzulegen und in jedem Paket nur auf die entsprechende Stelle zu verweisen. Somit ist sichergestellt, dass die Lizenz auf jedem Debian System installiert ist.

Das folgende Beispiel zeigt eine solche Datei mit einem Hinweis auf die GPL:

```
This package was debianized by Frank Ronneburg
<fr@debiananwenderhandbuch.de> on Tue,  6 Nov 2001 22:29:53
+0100.  It was downloaded from http://www.markus-pfeiffer.de/
Upstream Author(s): Markus Pfeiffer <dtcltiny@markus-pfeiffer.de>
This code is released under the terms of the GPL license.  See
/usr/share/common-licenses/GPL for the full license.
```

Die Datei control enthält verschiedenste Angaben zu dem Paket. Wichtig ist hier die Zeile Section:, dies ist der Bereich, in dem das Paket in der Debian Paketstruktur erscheint. Die Zeile Description: enthält in einer Zeile eine kurze Beschreibung des Pakets. Alle weiteren Zeilen müssen mit einem Leerzeichen beginnen und beschreiben das Paket ausführlich. Auch diese Texte sollten in englischer Sprache verfasst werden (das hier gezeigte Beispiel spiegelt lediglich den Zeitmangel des Autors wieder.).

```
Source: dtcltiny Section: x11 Priority: optional Maintainer: Frank
Ronneburg <fr@debiananwenderhandbuch.de> Build-Depends:
debhelper (>> 3.0.0) Standards-Version: 3.5.2  Package: dtcltiny
Architecture: any Depends: ${shlibs:Depends} Description: Control
```

Locomotives on a Model Railroad, needs SRCP Server dtcltiny ist ein Programm, mit dem man Loks auf einer Modellbahn steuern kann, die mit einem Digitaldecoder ausgerüstet sind. Es stellt für jede Lok einen Regler zur Verfügung, mit dem man die Geschwindigkeit, Fahrtrichtung und Sonderfunktionen einstellen kann. Nebenbei bietet es noch die Möglichkeit, Decoder nach dem NMRA-DCC-Standard und Uhlenbrock-Decoder zu programmieren.

Die meisten der verwendeten Schlüsselwörter sollten selbsterklärend sein. Source: beschreibt den Namen (der Sourcen) des Pakets; mit Priority: kann bestimmt werden, ob auf das Paket verzichtet werden kann (optional) oder ob das Paket zwingend notwendig für das System ist. Die Zeile Maintainer enthält den Namen und die E-Mail-Adresse des Paketbetreuers; diese Angaben sind in der gezeigten Schreibweise anzugeben. Build-Depends: beschreibt die Abhängigkeiten des Pakets. Mit dem Schlüsselwort Package: wird der Paketname des Binär-Pakets beschrieben; die folgende Zeile Architecture: gibt an, für welche Architekturen das Paket verwendet werden kann. Dies kann beispielsweise i386, powerpc, alpha usw. sein oder, wie hier gezeigt, das Schlüsselwort any für alle Architekturen.

In dieser Datei werden die Regeln zum Erstellen des Pakets definiert. Sollte der von Ihnen verwendete Source-Code kein Makefile enthalten oder auch kein configure-Script verwenden, so müssen Sie in dieser Datei die notwendigen Schritte zum erfolgreichen Übersetzen des Pakets beschreiben. Im hier gezeigten Beispiel wurde make ($(MAKE)) mit der Option -j 6 ergänzt, um die Übersetzung etwas zu beschleunigen.

Eventuell kann es, beispielsweise bei Paketen, die ausschließlich Dokumentation enthalten, notwendig sein, alle Zeilen, die make aufrufen, auszukommentieren. Sie müssen dann durch geeignete Einträge in der Datei selbst dafür sorgen, dass die Dateien an die richtige Stelle im Dateisystem kopiert werden. Die hier gezeigte Version wurde bereits an einigen Stellen angepasst.

Im Abschnitt install: build sind einige Regeln definiert worden, mit denen das ausführbare Programm durch ein Skript ersetzt wird. Weiterhin wird eine Konfigurationsdatei als Beispiel im Dokumentationsverzeichnis abgelegt.

#!/usr/bin/make -f # Sample debian/rules that uses debhelper. # GNU copyright 1997 to 1999 by Joey Hess. # Uncomment this to turn on verbose mode. #export DH_VERBOSE=1 # This is the debhelper

```
compatability version to use. export DH_COMPAT=3  configure:
configure-stamp configure-stamp:        dh_testdir  # Add here
commands to configure the package.    ./configure --prefix=/usr --
program-suffix=.bin   touch configure-stamp  build: configure-stamp
build-stamp build-stamp:    dh_testdir  # Add here commands to
compile the package.  $(MAKE) -j 6    #/usr/bin/docbook-to-man
debian/dtcltiny.sgml > dtcltiny.1          touch build-stamp  clean:
      dh_testdir  dh_testroot         rm -f build-stamp configure-stamp
      # Add here commands to clean up after the build process.         -
$(MAKE) clean         dh_clean  install: build         dh_testdir
      dh_testroot         dh_clean -k         dh_installdirs     # Add here
commands to install the package into debian/dtcltiny.     mkdir -p
$(CURDIR)/debian/dtcltiny/usr   $(MAKE) install
prefix=$(CURDIR)/debian/dtcltiny/usr                cp
$(CURDIR)/debian/dtcltiny.sh
$(CURDIR)/debian/dtcltiny/usr/bin/dtcltiny  chmod a+x
$(CURDIR)/debian/dtcltiny/usr/bin/dtcltiny          # Demo Konfigfile
      mkdir -p $(CURDIR)/debian/dtcltiny/usr/share/doc/dtcltiny/    cp
/home/fr/.dtcltiny.data \
      $(CURDIR)/debian/dtcltiny/usr/share/doc/dtcltiny/SAMPLE.dtclti
ny.data   # Build architecture-independent files here. binary-indep:
build install # We have nothing to do by default.  # Build architecture-
dependent files here. binary-arch: build install       dh_testdir
      dh_testroot #       dh_installdebconf              dh_installdocs
      dh_installexamples      dh_installmenu #         dh_installlogrotate #
      dh_installemacsen #    dh_installpam #  dh_installmime #
      dh_installinit      dh_installcron     dh_installman     dh_installinfo #
      dh_undocumented       dh_installchangelogs ChangeLog  dh_link
      dh_strip     dh_compress       dh_fixperms #     dh_makeshlibs
      dh_installdeb #   dh_perl       dh_shlibdeps       dh_gencontrol
      dh_md5sums       dh_builddeb  binary: binary-indep binary-arch
.PHONY: build clean binary-indep binary-arch binary install configure
```

Mit dem Debian Menüsystem können Programme, unabhängig von der verwendeten Benutzeroberfläche, in die Menüs der Benutzeroberfläche eingebunden werden. Bei der Verwendung eines Icons ist darauf zu achten, dass dieses auch tatsächlich vorhanden ist.

Gegebenenfalls muss das Icon mit dem Paket zusammen installiert werden. Das Beispiel-Paket dtcltiny verwendet ein Icon aus dem Gnome-Paket.

```
?package(dtcltiny):needs=X11 section=Apps/Tools \
icon="/usr/share/pixmaps/gnome-diskfree.png"\   title="dtcltiny"
command="/usr/bin/dtcltiny"
```

Neben dem Paketnamen finden sich in dieser Datei weitere Angaben dazu, welche Benutzeroberfläche das Programm benötigt und in welchem Menüzweig es erscheinen soll. Das Schlüsselwort title beschreibt den Text, der im Menü angezeigt wird; command schließlich enthält den kompletten Pfad und den Programmnamen.

Diese vier Dateien enthalten Informationen, die vor (pre) bzw. nach (post) dem Installieren bzw. dem Entfernen des Pakets ausgeführt werden sollen. Hier nur ein Beispiel:

Alle in der Datei postinst enthaltenen Kommandos werden nach (post) der Installation (inst) des Pakets ausgeführt. Durch dh_make wurde im Verzeichnis debian/ eine Vorlage erzeugt (postinst.ex), die umbenannt und mit Inhalten gefüllt werden kann.

```
#! /bin/sh # postinst script for erddcd # # see: dh_installdeb(1)  set -e  # summary of how this script can be called: #        * <postinst> 'configure' <most-recently-configured-version> #        * <old-postinst> 'abort-upgrade' <new version> #        * <conflictor's-postinst> 'abort-remove' 'in-favour' <package> #          <new-version> #        * <deconfigured's-postinst> 'abort-deconfigure' 'in-favour' #          <failed-install-package> <version> 'removing' #          <conflicting-package> <version> # for details, see /usr/share/doc/packaging-manual/ # # quoting from the policy: #     Any necessary prompting should almost always be confined to the #     post-installation script, and should be protected with a conditional #     so that unnecessary prompting doesn't happen if a package's #     installation fails and the 'postinst' is called with 'abort-upgrade', #     'abort-remove' or 'abort-deconfigure'.  case "$1" in configure)     ;;     abort-upgrade|abort-remove|abort-deconfigure)     ;;
```

```
*)        echo "postinst called with unknown argument \`$1'" >&2
exit 0     ;; esac  # dh_installdeb will replace this with shell code
automatically # generated by other debhelper scripts.  #DEBHELPER#
exit 0
```

Das Skript kann während der Installation mit verschiedenen Parametern aufgerufen werden. Entsprechende Abschnitte in der Vorlage können mit den gewünschten Kommandos gefüllt werden.

Wenn alle Anpassungen an den Dateien im Verzeichnis `debian/` vorgenommen wurden, so kann versucht werden, mit dem Kommando `dpkg-buildpackage` ein Debian Paket zu erzeugen. Hierbei ist zu beachten, dass das Kommando im Verzeichnis mit den entpackten Quelldateien ausgeführt wird.

5.4 Debian Pakete - Aufbau der Sourcen

5.3 Debian Pakete anpassen 5.5 Erstellen, prüfen und verwalten von Debian Paketen

5.5 Erstellen, prüfen und verwalten von Debian Paketen

5.4 Debian Pakete - Aufbau der Sourcen 5.6 Package-Dateien

Nachdem die Vorbereitungen an den Dateien im Verzeichnis debian/ abgeschlossen sind, kann mit dem Programm dpkg-buildpackage das Paket erstellt werden.

dpkg-buildpackage wird ohne weitere Parameter versuchen, die Sourcen und die Changes-Datei des Pakets digital zu signieren. Hierzu wird ein vorhandener GnuPG-Key benutzt. Ist kein solcher Key erzeugt worden, so kann das Signieren mit den Optionen -us (unsigned source) und -uc (unsigned changes) verhindert werden. Im einfachsten Fall genügt also das Kommando dpkg-buildpackage -us -uc (als „root“), um das Paket zu erzeugen.

dpkg-buildpackage verfügt natürlich noch über viele weitere Optionen:

debian:~# dpkg-buildpackage -h Debian dpkg-buildpackage . Copyright (C) 1996 Ian Jackson. Copyright (C) 2000 Wichert Akkerman This is free software; see the GNU General Public Licence version 2 or later for copying conditions. There is NO warranty. Usage: dpkg-buildpackage [options] Options: -r<gain-root-command> -p<sign-command> -d do not check build dependencies and conflicts -D check build dependencies and conflicts -k<keyid> the key to use for signing -sgpg the sign-command is called like GPG -spgp the sign-command is called like PGP -us unsigned source -uc unsigned changes -a<arch> Debian architecture we build for (implies -d) -b binary-only, do not build source } also passed to -B binary-only, no arch-indep files } dpkg-genchanges -S source only, no binary files } -t<system> set GNU system type } passed to dpkg-architecture -v<version> changes since version <version> } -m<maint> maintainer for package is <maint> } -e<maint> maintainer for release is <maint> } only passed -C<descfile> changes are described in <descfile> } to dpkg- -si (default) src includes orig for rev. 0 or 1 } genchanges -sa uploaded src always includes orig } -sd uploaded src is diff and .dsc only } -nc do not clean source tree (implies -b) -tc clean source tree when finished -ap add pause before starting signature process -h print this message -W Turn certain errors into warnings. } passed to -E When -W is turned on, -E turned it off. } dpkg-source -i[<regex>] ignore diffs of files matching regex } only passed to dpkg-source

dpkg-buildpackage benötigt eine Umgebung, die der späteren Installationsumgebung ähnlich ist. Das Paket kann als Superuser (root) erzeugt werden, um dies zu gewährleisten. Nun macht es wenig Sinn, einen GnuPG-Key für den Superuser zu erzeugen. Um dieses Problem zu umgehen, dient die Option -r, unmittelbar gefolgt von einem Kommando, das es dem Benutzer erlaubt, Kommandos als Superuser auszuführen bzw. das eine virtuelle root-Umgebung schafft. Dies kann beispielsweise das

Kommando fakeroot oder sudo sein, ggf. ist natürlich die Konfiguration des Programms (sudo) für diesen Benutzer anzupassen.

Viele der weiteren Optionen können und sollten in den Konfigurationsdateien im Verzeichnis debian/ definiert werden; bei Bedarf können diese aber auch auf der Kommandozeile angepasst werden.

cvs-buildpackage unterstützt das Erzeugen von Debian Paketen, bei denen die Sourcen des Pakets mit CVS verwaltet werden. cvs-buildpackage ist ein Skript, das dpkg-buildpackage nutzt. Dabei wird zunächst die Datei debian/changelog interpretiert und die aktuelle Paketversion exportiert. Es wird überprüft, ob alle Änderungen bereits in den CVS-Baum importiert wurden, sollte dies nicht der Fall sein, so wird der Vorgang abgebrochen und dem Benutzer wird die Möglichkeit gegeben, die veränderten Dateien in das CVS zu importieren.

Beachten Sie, dass das Arbeitsverzeichnis, aus dem heraus cvs-buildpackage aufgerufen wird, die spätere Entwicklungsumgebung enthält. In diesem Verzeichnis werden die aus dem CVS exportierten Dateien gespeichert, und cvd-buildpackage hat die volle Kontrolle über dieses Verzeichnis. Das bedeutet, dass alle Dateien, mit Ausnahme der originalen Sourcen, gelöscht werden können. Stellen Sie sicher, dass Sie nicht in Ihrem aktuellen Entwicklungszweig mit diesem Programm arbeiten; es können alle Daten verloren gehen!

Zusammen mit den Programmen cvs-inject und cvs-upgrade bildet dieses Programm die Infrastruktur, die es Debian Entwicklern erlaubt, CVS als zentrales Entwicklungswerkzeug einzusetzen. Hierbei wird insbesondere die Verwendung verschiedener Release-Stände eines Pakets, also beispielsweise „stable“, „unstable“ oder auch „Experimental“, unterstützt.

-h

Zeigt eine kurze Hilfe zu diesem Program an.

-m<module>

Der Name des Moduls im CVS.

-P<paketname>

Gibt den Namen des Pakets an. Dies ist sinnvoll, wenn das Programm nicht im CVS-Baum ausgeführt wird.

-V<version>

Bestimmt die Versionsnummer des Pakets. Zusammen mit dem Paketnamen kann diese Option außerhalb eines CVS-Baums verwendet werden.

-T<tag>

Der CVS-Tag zum Exportieren der Sourcen des Pakets. Hierbei wird nicht versucht, das Tag aus der Versionsnummer heraus zu ermitteln. Dies setzt zwingend voraus, dass Sie wissen, was dies bedeutet...

-U<tag>

CVS-Tag, aus dem die Sourcen exportiert werden sollen. Diesmal wird nicht versucht, aus der Versionsnummer zu extrahieren. Auch hier: Überlegen Sie genau, ob dies gewünscht ist.

-C<Build Command>

Name des Programms, um die Debian Pakete zu erzeugen; normalerweise wird dies dpkg-buildpackage sein. Der Benutzer kann aber auch ein beliebiges anderes Programm oder Skript verwenden. Beispielsweise kann das Kommando chroot /opt/root dpkg-buildpackage benutzt werden, um das Paket in einer abgeschirmten „chroot"-Umgebung zu erzeugen. Ein anderes Beispiel wäre die Verwendung von pdebuild --auto-debsign --buildresult ../, um die Pakete mittels pbuilder zu erzeugen. Dieses Argument überschreibt die Umgebungsvariable CVSDEB_BUILDPACKAGE und auch die Variable conf_buildpackage in der Konfigurationsdatei.

-G<get method>

Mit dieser Option kann ein Archiv des aktuellen Quellcodes aus dem Netz geladen werden. Dies überschreibt ggf. eine bereits bestehende Datei mit gleichem Namen. Üblicherweise werden hier Programme wie wget oder curl eingesetzt werden. Diese Option überschreibt die Umgebungsvariable CVSDEB_GET_ORIG und die Option conf_get_orig in der Konfigurationsdatei.

-A

Diese Option verwendet apt-get source, um ein aktuelles Archiv des Quellcodes zu beschaffen. Ist bereits ein aktuelles Archiv vorhanden, so hat diese Option keinerlei Wirkung. Werden die Optionen -A und -G zugleich angegeben, so wird zuerst -G beachtet. Hiermit wird die Umgebungsvariable CVSDEB_USE_APT sowie die Variable conf_use_apt in der Konfigurationsdatei überschrieben.

-R<root directory>

Das Wurzel-Verzeichnis des ursprünglichen Quellcodes. Es wird erwartet, dass das Archiv mit dem Quellcode im Verzeichnispfad <Wurzelverzeichnis><Paketname>/ zu finden ist, falls nicht das Arbeitsverzeichnis auf einen anderen Pfad gesetzt ist oder der Quellcode aus dem CVS exportiert wird. Wird das Arbeitsverzeichnis von cvs-buildpackage gesetzt (auf der Kommandozeile, in der Konfigurationsdatei oder durch Setzen der Umgebungsvariablen), so wird die Angabe für das Wurzelverzeichnis ignoriert.

-W<Arbeitsverzeichnis>

Diese Option gibt den vollen Pfad zum Arbeitsverzeichnis von cvs-buildpackage an, in dem der Quellcode aus dem CVS exportiert wird und in dem die Datei <package name>_<version>.orig.tar.gz liegt. Es ist nicht zwingend notwendig, dass ein Archiv mit dem originalen Quellcode vorhanden ist, da dies auch aus den Quellen aus dem CVS-Baum erzeugt werden kann. Hierbei kann es jedoch zu Unterschieden kommen, wenn bereits Veränderungen im CVS vorgenommen wurden. Bei Verwendung der Version aus dem CVS wird als Dateiname für das Archiv upstream_version_<version> verwendet, und zwar ohne Angabe der Debian Revision. Die Verwendung dieser Option überschreibt die Angaben in der Umgebungsvariablen CVSDEB_WORKDIR und in der Variablen conf_workdir in der Konfigurationsdatei.

-F

Diese Option, „Force“, hat nur Auswirkungen, wenn sie im Quellcode-Verzeichnis benutzt wird. Dies bewirkt, dass cvs tag -F aufgerufen wird, bevor die Quellen aus dem CVS exportiert werden. Es werden hierbei die Umgebungsvariable CVSDEB_FORCETAG sowie die Konfigurationsvariable conf_forcetag überschrieben.

-E

Diese Option bewirkt einen „Full Export“. Normalerweise exportiert cvs-buildpackage alle Daten aus dem CVS-Baum. Ist eine Datei orig.tar.gz nicht vorhanden, so wird der komplette CVS-Baum aus dem CVS exportiert, unabhängig davon, ob diese Option angegeben wird oder nicht. Dies überschreibt die Umgebungsvariable CVSDEB_FULLEXPORT und die Konfigurationsvariable conf_fullexport.

-op

Das Gegenteil von „Full Export“. Mit dieser Option wird eine Datei orig.tar.gz im aktuellen Arbeitsverzeichnis von cvs-buildpackage entpackt. Danach wird cvs rdiff benutzt, um den Quellcode-Baum mit dem aktuellen Stand aus dem CVS abzugleichen.

-ctp

Diese Option setzt den String package_ vor jedes CVS-Tag. Dies überschreibt den Konfigurationsdatei-Eintrag conf_forcetag und die Umgebungsvariable CVSDEB_PACKAGEINTAG. Normalerweise wird dieser String nicht vor das CVS-Tag geschrieben.

-n

Diese „No-Exec-Option“ bewirkt einen kompletten Testlauf von cvs-buildpackage, ohne tatsächlich Kommandos auszuführen; eine reine Simulation.

-f<fix_skript>

Mit dieser Option kann ein Skript angegeben werden, das im Quellcode-Verzeichnis ausgeführt wird, um Zugriffsrechte zu korrigieren. Dies ist notwendig, wenn neue Skripte von zusätzlichen Patches angelegt wurden.

-H<hook_script>

Mit dieser Option wird ein Skript vor dem Aufruf von dpkg-buildpackage aufgerufen. Dies überschreibt die Umgebungsvariable CVSDEB_HOOK, die wiederrum die Konfigurationsvariable conf_hook_script überschreibt.

-x<prefix>

Setzt das CVS-Präfix für das voreingestellte Modul. Dies überschreibt die Umgebungsvariable CVSDEB_PREFIX.

Alle weiteren Kommandozeilenargumente werden nicht interpretiert und vollständig an das Programm dpkg-buildpackage durchgereicht.

Neben den Kommandozeilenoptionen versucht cvs-buildpackage zunächst in der Datei /etc/cvsdeb.conf, systemweite Einstellungen zu finden. Danach wird, wenn vorhanden, eine benutzerbezogene Datei ~/.cvsdeb.conf gelesen und interpretiert.

cvs-autoreleasedeb erlaubt die Übertragung von Paketen auf einen Server, welche von cvs-buildpackage aus einem CVS-Modul erzeugt wurden. CVS

cvs-autoreleasedeb verwaltet Informationen zu allen Paketen, die automatisch auf einen Server übertragen werden sollen. Wird die Datei debian/changelog im CVS aktualisiert und die Debian Versionsnummer des Paketes dabei vergrößert, so wird dieses Paket automatisch auf den Server übertragen.

Alle Einstellungen dieses Programmes werden in der Konfigurationsdatei cvs-autoreleasedeb.conf verwaltet. cvs-autoreleasedeb kennt keinerlei Kommandozeilenparameter.

cvs-autoreleasedeb kann auf zwei Arten eingesetzt werden. Zunächst ist es möglich, als Benutzer cvs-autoreleasedeb das Skript über einen Cron-Job aufzurufen. Dies ist sinnvoll, wenn regelmäßig neue Versionen der Pakete (beispielsweise im Rahmen eines Software-Entwicklungsprojektes) zur Verfügung gestellt werden sollen. In diesem Fall wird die Konfigurationsdatei /etc/cvs-autoreleasedeb.conf gelesen und ausgewertet, die Pakete werden im Verzeichnis /var/lib/cvs-autoreleasedeb/ erzeugt. Sollcvs-autoreleasedeb auf diese Weise genutzt werden, so ist die Datei /etc/default/cvs-autoreleasedeb anzupassen. Die Ausgabe des Programmes cvs-autoreleasedeb wird in die Datei /var/log/cvs-autoreleasedeb/run.log geschrieben.

Alternativ kann cvs-autoreleasedeb durch den Benutzer direkt aufgerufen werden. In diesem Fall wird als Konfigurationsdatei die Datei $HOME/.cvs-autoreleasedeb/conf eingesetzt. Die erzeugten Pakete landen im Verzeichnis $HOME/.cvs-autoreleasedeb/. cvs-autoreleasedeb setzt hierbei keine Vorgabewerte ein, es muss also zwingend eine sinnvolle Konfigurationsdatei vorhanden sein.

5.5.4.1 Überprüfung von Verzeichnisnamen

Mit debchange kann die Datei debian/changelog in einem Debian Source-Verzeichnisbaum verwaltet werden. Vereinfacht kann auch der Alias dch verwendet werden. Das Kommando debchange muss immer innerhalb des Verzeichnisses aufgerufen werden, in dem sich die Sourcen zu dem zu erstellenden Debian Paket befinden. Es wird ein Texteditor gestartet und automatisch ein neuer Eintrag dem changelog hinzugefügt. (Es werden die Umgebungsvariablen VISUAL und EDITOR benutzt, um den gewünschten Editor zu bestimmen.) Wird der gewünschte Text bereits auf der Kommandozeile angegeben, so kann debchange auch in einem Skript ohne weitere Interaktion des Benutzers ablaufen. Der Eintrag wird an der entsprechenden Stelle hinzugefügt, inklusive aller notwendigen Umbrüche.

Die Syntax für dieses Kommando lautet:

```
debchange [options] [text ...] dch [options] [text ...]
```

Wird der Editor ohne Änderungen an der Datei verlassen, so wird auch von debchange keine weitere Änderung an der Datei vorgenommen. Wenn ein Editor benutzt wird, der mit der Option +n an eine bestimmte Stelle in einem Dokument springen kann, so wird die Schreibmarke direkt an die passende Stelle in der changelog-Datei gesetzt.

Beachten Sie, dass changelog-Dateien immer mit einer UTF-8-Kodierung gespeichert werden. Sollte dies nicht der Fall sein, so können Sie das Programm iconv verwenden, um die Datei umzuwandeln.

debchange kann darüber hinaus mit der Option --closes Einträge erzeugen, die dazu führen, dass ein Bug im Debian Bug Tracking System geschlossen wird. Hierzu wird das Debian BTS (Debian Bug Tracking System, bugs.debian.org/) nach dem Titel und dem Paketnamen durchsucht. Mit der Option --noquery oder der Umgebungsvariablen DEBCHANGE_QUERY_BTS, die auf den Wert no gesetzt sein muss, kann diese Abfrage verhindert werden.

Meistens wird eine der Optionen --append, --increment oder --newversion, wie weiter unten beschrieben, eingesetzt. Werden keine Optionen angegeben, so versucht debchange eine Logdatei von dupload oder dput im darüberliegenden Verzeichnis zu finden, um zu ermitteln, ob die bestehende Version eines Paketes bereits erfolgreich auf einen Server übertragen wurde. Die Ausgabe einer Warnung erfolgt, falls eine Logdatei gefunden wurde, in dieser jedoch kein erfolgreicher Upload verzeichnet ist.

Werden die Optionen --increment und --newversion benutzt, so wird der Name und die E-Mail- Adresse für die neue Version wie folgt ermittelt. Ist die Umgebungsvariable DEBFULLNAME gesetzt, so wird der Inhalt als Name des Maintainers (Paketbetreuer) verwendet. Weiterhin wird die Umgebungsvariable DEBEMAIL ausgewertet. Ist diese gesetzt, so wird der Inhalt als Mail-Adresse verwendet. Hat diese Variable einen Wert, der der Syntax name <email> entspricht, so wird auch der Name aus dieser Variablen verwendet, falls DEBFULLNAME nicht gesetzt ist.

Ist die Variable DEBEMAIL nicht gesetzt, so wird versucht, die Information aus der Variablen EMAIL zu ermitteln. Konnte der Name bisher nicht ermittelt werden, so wird der Systemaufruf getpwuid(3) verwendet, um den Namen aus der Datei /etc/passwd zu ermitteln. Schlägt auch dies fehl, so wird der letzte Eintrag in der changelog-Datei ausgewertet.

Konnte die E-Mail-Adresse nicht ermittelt werden, so wird versucht, diese aus der Datei /etc/mailname zu ermitteln. Danach wird der Benutzername zusammen mit dem FQDN getestet. Gibt auch dies keinen sinnvollen Wert, so wird auch hier die Datei changelog ausgewertet. Abschließend ist festzuhalten, dass die Variablen DEBFULLNAME und DEBEMAIL gesetzt werden sollten, wenn dieses Skript eingesetzt wird.

Besteht der Verzeichnisname, der den Quellcode enthält, aus dem Paketnamen und der Versionsnummer, so versucht debchange dieses Verzeichnis umzubenennen, falls die Upstream-Versionsnummer verändert wurde. Dies kann mit der Option --preserve auf der Kommandozeile oder auch in der Konfigurationsdatei verhindert werden, wie weiter unten beschrieben.

Wie viele andere Skripte aus dem Paket devscripts versucht auch debchange, rekursiv im Verzeichnisbaum die Datei debian/changelog zu finden. Wird diese Datei gefunden, so wird überprüft, ob das darüber liegende Verzeichnis dem Paketnamen entspricht. Wie dies genau durchgeführt wird, kann durch die beiden Konfigurationsvariablen DEVSCRIPTS_CHECK_DIRNAME_LEVEL und DEVSCRIPTS_CHECK_DIRNAME_REGEX sowie deren Kommandozeilen-Gegenstücke --check-dirname-level bzw. --check-dirname-regex beeinflusst werden.

Die Variable DEVSCRIPTS_CHECK_DIRNAME_LEVEL kann dabei die folgenden Werte annehmen:

0

Der Verzeichnisname wird nicht überprüft.

1

Überprüft den Verzeichnisnamen nur, wenn bei der Suche nach der Datei debian/changelog ein Wechsel des Verzeichnisses notwendig ist.

2

Der Verzeichnisname wird in jedem Fall überprüft.

--increment, -i

Erhöht die Versionsnummer des Pakets um eins. Dies erzeugt einen neuen Eintrag am Anfang des changelog mit den entsprechenden Kopf- und Fußzeilen. Handelt es sich um ein neues Debian Paket, so wird auch der Verzeichnisname angepasst.

--append, -a

Fügt einen neuen Eintag am Ende der aktuellen Version ein.

--newversion, -v

Gibt eine spezifische Versionsnummer an (inklusive des Debian Release) und verhält sich ansonsten wie die Option --increment. Auch hier wird der Verzeichnisname angepasst, wenn dies notwendig ist.

--fromdirname, -d

Diese Option interpretiert die Versionsnummer aus dem aktuellen Verzeichnis, diese muss in der gebräuchlichen Form paketname-versionsnummer vorliegen.

--closesnnnnn,[nnnnn,...]

Erzeugt changelog-Einträge, die die angegebenen Bug-Nummern im BTS schließen. Es wird eine Warnung ausgegeben, falls das Debian BTS nicht erreichbar ist und die Option --noquery nicht angegeben wurde.

--[no]query

Legt fest, ob das BTS benutzt werden soll, wenn entsprechende Einträge im changelog vorgenommen werden.

--preserve, -p

Lässt den Verzeichnisnamen des Source-Baums unverändert, auch wenn sich die Versionsnummer des Pakets verändert.

--no-preserve

Ändert den Verzeichnisnamen des Source-Baums nicht (die Voreinstellung).

--distribution dist, -D dist

Setzt die angegebene Distribution in den changelog-Eintrag.

--urgency urgency, -u urgency

Setzt den angegebenen Eintrag für die Dringlichkeit in die changelog-Datei. Der Standardwert ist „low“.

--no-conf, --noconf

Benutzt keinerlei Voreinstellungen aus der Konfigurationsdatei. Dies muss als erst Option angegeben werden!

--check-dirname-level N

Wie vorab beschrieben, beeinflusst diese Option das Durchsuchen von Verzeichnissen.

--check-dirname-regex regex

Wie vorab beschrieben, beeinflusst diese Option das Durchsuchen von Verzeichnissen.

--help, -h

Zeigt eine kurze Information als Hilfe an.

--version

Zeigt Informationen zur Version und zum Copyright dieses Programms an.

debchange versucht beim Start, die Konfigurationsdateien /etc/devscripts.conf und ~/.devscripts in dieser Reihenfolge einzulesen. Werte in den Konfigurationsdateien können durch Optionen auf der Kommandozeile überschrieben werden. Folgende Variablen werden in den Konfigurationsdateien unterstützt:

DEBCHANGE_PRESERVE

Wird diese Variable auf den Wert yes gesetzt, so hat dies die gleiche Wirkung wie die Kommandozeilenoption --preserve.

DEBCHANGE_QUERY_BTS

Auf den Wert no gesetzt, bewirkt diese Variable das Gleiche wie die Kommandozeilenoption --noquery.

DEVSCRIPTS_CHECK_DIRNAME_LEVEL, DEVSCRIPTS_CHECK_DIRNAME_REGEX

Wie zuvor beschrieben, beeinflussen diese beiden Variablen die Verzeichnisse, die nach einer Datei debian/changelog durchsucht werden sollen. Bei der Verwendung dieser Variablen in den globalen Konfigurationsdateien ist zu beachten, dass dies Einfluss auf alle Debian Pakete hat, die auf diesem System erstellt werden.

Weiterhin werden von debchange die Umgebungsvariablen DEBEMAIL, EMAIL und DEBFULLNAME ausgewertet. Mit der Umgebungsvariable VISUAL oder EDITOR kann auch der bevorzugte Texteditor bestimmt werden.

debdiff vergleicht die Dateien in zwei einzelnen oder auch in Gruppen von Debian Paketen. Die Syntax für dieses Kommando lautet:

debdiff [options] ... deb1 deb2 debdiff [options] ... changes1 changes2 debdiff [options] ... --from deb1a deb1b ... --to deb2a deb2b ... debdiff [options] ... dsc1 dsc2

Beim Vergleich der Debian Pakete werden alle Dateien verglichen, die direkt zum Paket gehören, also nicht die Hilfsskripte, die vor oder nach der Installation der Dateien aufgerufen werden. Angezeigt werden dabei alle neu hinzugekommenen oder aus dem Paket entfernten Dateien zwischen den beiden Versionen des Pakets. Weiterhin wird mit dem Programm wdiff ein Vergleich der control-Dateien in den Paketen vorgenommen und das Ergebnis angezeigt.

Es können mit debdiff auch Gruppen von Paketen verglichen werden, debdiff unterstützt dies auf zwei Arten: Zunächst können zwei .changes-Dateien angegeben werden. Die in diesen Dateien aufgeführten Debian Pakete werden daraufhin verglichen, indem der Inhalt aller Pakete zusammengeführt wird. Hierzu müssen sich die Debian Pakete im Verzeichnis der .changes-Datei befinden.

Alternativ können die gewünschten Debian Pakete auf der Kommandozeile mittels der Optionen --from und --to angegeben werden. Hiermit kann festgestellt werden, ob bei einem Aufsplitten des Pakets Dateien verloren gegangen sind. control-Dateien werden in beiden Fällen nur beachtet, wenn lediglich zwei Debian Pakete miteinander verglichen werden.

debdiff verwendet die Konfigurationsdateien aus dem Paket devscripts. Die Einstellungen in den Konfigurationsdateien können dabei immer mit den Kommandozeilenoptionen überschrieben werden.

Werden debdiff zwei Quellcode-Pakete (.dsc-Dateien) beim Aufruf übergeben, so werden die Inhalte dieser Quellcode-Pakete verglichen. Unterscheiden sich die beiden Pakete nur in der Debian Versionsnummer (die Dateien .orig.tar.gz und .dsc sind gleich), dann wird das Programm interdiff benutzt, um die „Patches“ zu vergleichen. Ist dieses Programm nicht vorhanden, so wird diff verwendet, um die beiden Quellcode-Bäume miteinander zu vergleichen.

--dirs, -d

Normalerweise werden Verzeichnisnamen beim Vergleich von Paketen ignoriert. Mit dieser Option kann debdiff dazu veranlasst werden, den Vergleich der Pakete auch auf Verzeichnisse anzuwenden.

--nodirs

Ignoriert Verzeichnisse beim Vergleich. Dies ist die Voreinstellung; diese Option kann aber benutzt werden, wenn in den Konfigurationsdateien etwas anderes definiert wurde.

--move FROM TO, -m FROM TO

Es kann vorkommen, dass Dateien oder Verzeichnisse zwischen Paketversionen verschoben werden. Mit dieser Option kann debdiff davon in Kenntnis gesetzt werden. Dabei ist zunächst (als FROM) der Name der Datei oder des Verzeichnisses in dem älteren Paket anzugeben. Als zweiter Parameter folgt dann die neue Position der Datei oder des Verzeichnisses.

Es kann eine beliebige Anzahl von verschobenen Dateien oder Verzeichnissen angegeben werden, diese werden alle der Reihe nach abgearbeitet.

--move-regex FROM TO

Wie die vorher beschriebene Option, jedoch wird die Angabe als regulärer Ausdruck in Perl-Schreibweise (s/^FROM/TO/) interpretiert.

--nocontrol

Werden nur zwei Debian Pakete verglichen, so werden normalerweise die control-Dateien mittels wdiff verglichen. Diese Option unterdrückt dieses Verhalten.

--control

Vergleicht die control-Dateien. Dies ist die Voreinstellung, kann aber benutzt werden, um Informationen in den Konfigurationsdateien zu überschreiben.

--wp, --wl, --wt

Reicht die Optionen -p, -l oder -t an das Programm wdiff weiter. Dies führt zu einer ausführlicheren Ausgabe von wdiff.

--show-moved

Wenn mehrere Debian Pakete auf der Kommandozeile angegeben wurden (über die .changes-Dateien oder mittels der --from/ --to-Syntax), dann zeigt diese Option auch die Dateien an, die zwischen den Paketen verschoben wurden.

--noshow-moved

Dies ist die Voreinstellung (siehe vorhergehende Option). Hiermit kann eine Einstellung in den Konfigurationsdateien überschrieben werden.

--no-conf, --noconf

Verhindert das Einlesen von Konfigurationsdateien. Dies muss als erste Option auf der Kommandozeile angegeben werden.

--renamed FROM TO

Wird die Option --show-moved verwendet und ein Paket somit umbenannt, so wird debdiff eingesetzt, um die Pakete zu vergleichen.

--help, -h

Zeigt eine kurze Übersicht aller Optionen.

--version, -v

Zeigt Informationen zur Version und zum Copyright dieses Programms.

debdiff benutzt die Konfigurationsdateien /etc/devscripts.conf und ~/.devscripts. Alle in diesen Konfigurationsdateien verwendeten Optionen können auf der Kommandozeile überschrieben werden. debdiff benutzt die folgenden Variablen in den Konfigurationsdateien:

DEBDIFF_DIRS

Wird diese Variable auf yes gesetzt, so wirkt dies wie die Kommandozeilenoption --dirs.

DEBDIFF_CONTROL

Auf den Wert no gesetzt, bewirkt diese Option das Gleiche wie die die Kommandozeilenoption --nocontrol. Der Vorgabewert ist yes.

DEBDIFF_SHOW_MOVED

Auf yes gesetzt, hat dies die Wirkung der Kommandozeilenoption --show-moved.

DEBDIFF_WDIFF_OPT

Diese Variable wird zu wdiff weitergereicht und kann den Wert -p, -l oder -t annehmen.

wdiff vergleicht zwei Dateien und gibt Informationen darüber aus, welche ganzen Wörter sich in diesen beiden Dateien unterscheiden. Ein Wort ist dabei eine Zeichenkette zwischen zwei Leerzeichen.

dpkg-depcheck ermittelt die Abhängigkeiten eines Programms. Hierbei wird das Programm mittels strace gestartet und die verwendeten Programme ermittelt. Die ausgegebene Liste kann mit verschiedenen Optionen den Bedürfnissen angepasst werden.

Zur Veranschaulichung hier ein Beispiel:

```
fr@wasabi:~$ dpkg-depcheck -ab gpg --list-key 887EB817  gpg: Hinweis: Alte voreingestellte Optionendatei '/home/fr/.gnupg/options' wurde ignoriert gpg: WARNUNG: Sensible Daten könnten auf Platte ausgelagert werden. gpg: siehe http://www.gnupg.org/faq.html für weitere Informationen pub  1024D/887EB817 2000-12-12 Frank Ronneburg <fr@tischbahn.de> uid                       Frank Ronneburg <fr@debiananwenderhandbuch.de> sub  1024g/B91B2CCC 2000-12-12    ------------------------------------------------------------------- The following files did not appear to belong to any package: /usr/lib/locale/locale-archive Packages needed:   locales   gnupg   libbz2-1.0  (Build-)Essential packages used:   zlib1g   libc6
```

Zunächst wird die Ausgabe des Programms, hier gpg --list-key 887EB817, ausgegeben. Danach folgen die Auswertungen von dpkg-depcheck. Die Optionen -ab zeigen die Abhängigkeiten zur Laufzeit (-a) und zur Erzeugung (-b) des Pakets an.

Die Syntax von dpkg-depcheck lautet:

```
dpkg-depcheck [options] command
```

-a, --all

Zeigt alle Pakete an, die zur Ausführung des Kommandos benötigt werden.

-b, --build-depends

Zeigt auch Pakete an, die zum Übersetzen dieses Programms benötigt werden. Berücksichtigt direkte und indirekte Anhängigkeiten.

-d, --ignore-dev-deps

Zeigt keine Pakete an, die zur Entwicklung (*-dev*) benötigt werden.

-m, --min-deps

Zeigt nur die minimal benötigten Pakete an.

-C, --C-locale

Führt das Programm mit der angegebenen Sprachumgebung aus.

--no-C-locale

Verändert die Sprachumgebung bei der Ausführung des Programms nicht.

-l, --list-files

Zeigt die zu den Paketen gehörenden Dateien mit an.

--no-list-files

Zeigt die zu den Paketen gehörenden Dateien nicht mit an.

-o, --output=DATEI

Schreibt die Ausgabe in die angegebene Datei, anstatt diese auf der Standardausgabe anzuzeigen.

-O, --strace-output=DATEI

Schreibt die Ausgabe von strace in die angegebene Datei, anstatt diese auf der Standardausgabe anzuzeigen.

-f, --features=LIST

Aktiviert oder deaktiviert Funktionen, die als kommaseparierte Liste angegeben werden. Eine Funktion wird dabei mit dem vorangestellten Zeichen + aktiviert; zur Vereinfachung kann dieses Zeichen auch weggelassen werden. Funktionen werden durch das vorangestellte Zeichen - deaktiviert. Die möglichen Funktionen sind:

warn-local

Gibt eine Warnung aus wenn, Dateien in den Verzeichnissen /usr/local/ oder /var/local/ benutzt werden. Diese Funktion ist normalerweise aktiviert.

discard-check-version

Ignoriert den Aufruf von Programmen mit der Option --version. Dies wird häufig in Configure-Skripten benutzt. Diese Funktion ist normalerweise aktiviert.

trace-local

Versucht auch Dateien aus /usr/local/ und /var/local/ zu analysieren. Dies ist auf einem Debian System meistens nicht sinnvoll, da dort keine Dateien abgelegt werden. Diese Funktion ist normalerweise deaktiviert.

catch-alternatives

Warnt bei Zugriffen auf Dateien, die über Debian-Alternativen verwaltet werden (siehe Alternativen). Diese Funktion ist normalerweise aktiviert.

discard-sgml-catalogs

Ignoriert SGML-Katalog-Dateien; diese werden von einigen SGML-Tools beim Starten geöffnet. Diese Funktion ist normalerweise aktiviert.

--no-conf, --noconf

Verhindert das Einlesen von Konfigurationsdateien und muss als erste Option auf der Kommandozeile angegeben werden.

-h, --help

Gibt einige Informationen zur Nutzung des Programms aus.

-v, --version

Gibt die Versionsnummer und Informationen zum Copyright des Programms aus.

dpkg-debcheck nutzt die beiden Konfigurationsdateien /etc/devscripts.conf und ~/.devscripts. Über die beschriebenen Kommandozeilenoptionen können die Werte in den Konfigurationsdateien überschrieben werden. Alternativ können hierzu auch Umgebungsvariablen genutzt werden.

dpkg-debcheck nutzt momentan nur die Umgebungsvariable DPKG_DEPCHECK_OPTIONS. Die hier angegebenen Optionen werden vor den Optionen auf der Kommandozeile ausgewertet. Hier ein Beispiel:

```
DPKG_DEPCHECK_OPTIONS="-b -f-catch-alternatives"
```

Mit dscverify können die PGP- oder GnuPG-Signaturen von .changes- oder .dsc-Dateien mit den Schlüsseln aus dem Debian Schlüsselbund (aus dem Paket debian-keyring) überprüft werden. Mit einer Option können auch andere Schlüsselbunde angegeben werden. Es werden alle angegebenen Dateien in den .changes- und .dsc-Dateien überprüft. Werden keine Abweichungen festgestellt, so ist der Rückgabewert 0.

--keyringkeyring

Schlüsselbunddatei, die dem Schlüsselbund hinzugefügt werden soll.

--no-conf, --noconf

Verhindert das Einlesen von Konfigurationsdateien. Muss als erste Option angegeben werden.

--help, -h

Zeigt eine kurze Hilfe zu dem Programm an.

--version

Zeigt Informationen zur Programmversion an.

Als einzige Konfigurationsvariable wird momentan DSCVERIFY_KEYRINGS unterstützt. Dies ist eine kommaseparierte Liste von Schlüsselbunden, die zur Überprüfung herangezogen werden sollen. Zusätzlich auf der Kommandozeile angegebene Schlüsselbunde werden ebenfalls berücksichtigt.

Es werden weiterhin die Konfigurationsdateien /etc/devscripts.conf und ~/.devscripts ausgewertet. Diese werden ignoriert, wenn die Option --no-conf oder --noconf angegeben wird.

Dieses Programm holt die Datei update_excuses.html aus dem Netz und durchsucht diese nach einem angegebenen Maintainer (Betreuer eines Debian Paketes). Diese Datei enthält, für die jeweils aktuelle „testing"-Distribution, „Ausreden" der Betreuer, warum ein Paket nicht aktuell oder in einem schlechten Zustand ist.

Die Syntax von grep-excuses lautet

```
grep-excuses [options] [maintainer|package]
```

Wird beim Aufruf des Programmes kein Name eines Paketbetreuers angegeben, so wird versucht, den Namen aus der Umgebungsvariablen DEBFULLNAME zu lesen. Schlägt auch dies fehl, so wird der Name aus den Konfigurationsdateien /etc/devscripts.conf und ~/.devscripts aus dem Wert GREP_EXCUSES_MAINTAINER ermittelt.

Alternativ kann zum Namen des Betreuers auch ein Paketname auf der Kommandozeile angegeben werden.

plotchangelog stellt Änderungen an einem Debian Paket anhand der Informationen aus der Datei changelog grafisch dar. Die Syntax von plotchangelog lautet

```
plotchangelog [options] changelog ...
```

plotchangelog nutzt das Programm gnuplot für die grafische Darstellung. Die X-Achse stellt dabei den Zeitpunkt des Release einer Version dar, die Y-Achse repräsentiert die Versionsnummer selbst. Jedes einzelne Release wird dabei durch einen Punkt auf der

Zeitachse dargestellt, betreuen unterschiedliche Maintainer ein Paket, so wird für jeden Maintainer dieses Paketes eine gesonderte Linie mit Punkten in unterschiedlichen Farben gezeichnet.

Alternativ kann auf der Y-Achse die Größe jedes Eintrages in der Datei changelog oder die Anzahl der beseitigten Fehler je Version des Paketes dargestellt werden.

-l, --linecount

Statt der Debian Versionsnummer des Paketes für die Daten der Y-Achse wird die Anzahl der Zeilen in der changelog-Datei für die Darstellung der Grafik verwendet. Diese Option kann nicht zusammen mit der Option --bugcount verwendet werden.

-b, --bugcount

Statt der Debian Versionsnummer des Paketes für die Daten der Y-Achse wird die Anzahl der behobenen Fehler (Bugs) von einer Version zur nächsten für die Darstellung der Grafik verwendet. Die Anzahl der Fehler wird ermittelt, indem nach Einträgen in der Form #nnnn in der Datei changelog gesucht wird. Werden die Werte nicht in dieser Form angegeben, so ist die erzeugte Grafik fehlerhaft. Diese Option kann nicht zusammen mit --linecount verwendet werden.

-c, --cumulative

Zusammen mit einer der Optionen --bugcount oder --linecount wird der durchschnittliche Wert der Fehler bzw. Zeilen angezeigt.

-v, --no-version

Zeigt nicht die Haupt-Versionsnummern (Upstream-Version) eines Paketes an. Dies ist bei vielen Änderungen an einem Paket sinnvoll, um die Grafik übersichtlicher erscheinen zu lassen.

-m, --no-maint

Differenziert nicht zwischen verschiedenen Betreuern (Maintainer) eines Paketes.

-s datei, --save=datei

Speichert die erzeugte Grafik im Postscript-Format in der Datei datei, statt diese direkt anzuzeigen.

-u, --urgency

Erzeugt größere Punkte, wenn das Paket mit einer höheren Priorität auf dem Server abgelegt wurde.

--verbose

Zeigt das erzeugte Skript an, welches an gnuplot weitergereicht wird. Diese Option dient zur Fehlersuche.

-gcommands, --gnuplot="commands"

Diese Option erlaubt es, weitere Kommandos an das Programm gnuplot durchzureichen. Diese werden im erzeugten Skript nach der Initialisierung von gnuplot aber noch vor dem eigentlichen „plot“-Kommando eingefügt. Auf diesem Weg kann das Layout der Grafiken beeinflusst werden. Weiterhin kann beispielsweise das Kommando set terminal png color verwendet werden, um das Dateiformat für die Ausgabe im Zusammenhang mit der Option -s zu verändern.

--help

Zeigt eine Hilfe zum Programm an.

--version

Zeigt die Versionsnummer, den Autor und die Copyright-Informationen zu diesem Programm an.

changelog ...

Pfad und Dateiname zur auszuwertenden changelog-Datei. Werden mehrere Dateien angegeben, so werden die Informationen aus allen Dateien in einer Grafik zusammengefasst. Die Dateien können mit gzip komprimiert sein, sie werden automatisch entpackt. Textabschnitte in den Dateien, die nicht dem Dateiformat einer Debian Changelog Datei entsprechen, werden ignoriert.

--no-conf

Diese Option verhindert, dass Konfigurationsdateien ausgewertet werden. Es werden die Konfigurationsdateien /etc/devscripts.conf und ~/.devscripts (in dieser Reihenfolge) in der Shell-Umgebung ausgewertet. Hierbei werden die folgenden Variablen genutzt:

PLOTCHANGELOG_OPTIONS

Diese Variable ist eine Zeile von durch Leerzeichen getrennten Optionen, die in jedem Fall zu benutzen sind, dies kann beispielsweise der String -l -b sein. Die Option -g oder --gnuplot ist hier nicht erlaubt und wird ggf. ignoriert.

PLOTCHANGELOG_GNUPLOT

Optionen für den Aufruf von gnuplot.

5.5.10.1 Überprüfung von Verzeichnisnamen

debclean be-„reinigt“ einen Quellcode-Verzeichnnisbaum. debclean durchsucht ab dem Verzeichnis, aus dem heraus es aufgerufen wurde, alle Unterverzeichnisse und führt das Kommando debian/rules clean für jedes gefundene Debian Quellcode-Verzeichnis aus. Die entsprechenden Verzeichnisse werden anhand einer vorhandenen Datei debian/changelog gefunden, in der der Paketname dem Verzeichnisnamen entspricht. Dieser Vergleich wird später noch genauer erläutert.

Der Aufruf von debclean erfolgt anhand der Syntax

```
debclean [options]
```

Wird die Option --cleandebs verwendet, so werden die Dateien mit der Endung *.deb, *.changes und *.build in allen Verzeichnissen entfernt. Die Dateien .dsc, .diff.gz und (.orig).tar.gz werden nicht verändert, so dass die aktuelle Version eines Paketes in jedem Fall wiederhergestellt werden kann. Weiterhin wird die Datei .upload nicht verändert, damit debchange korrekt funktioniert.

Die Option --nocleandebs verhindert das zuvor beschriebene Verhalten. Voreingestellt ist, dass keine Dateien gelöscht werden. debclean nutzt das Kommando debuild zum Bereinigen des Verzeichnisbaumes.

Wie viele andere Skripte aus dem Paket devscripts versucht auch debclean, rekursiv im Verzeichnisbaum die Datei debian/changelog zu finden. Wird diese Datei gefunden, so wird überprüft, ob das darüber liegende Verzeichnis dem Paketnamen entspricht. Wie dies genau durchgeführt wird, kann durch die beiden Konfigurationsvariablen DEVSCRIPTS_CHECK_DIRNAME_LEVEL und DEVSCRIPTS_CHECK_DIRNAME_REGEX sowie deren Kommandozeilen-

Gegenstücke --check-dirname-level bzw. --check-dirname-regex beeinflusst werden.

Die Variable DEVSCRIPTS_CHECK_DIRNAME_LEVEL kann dabei die folgenden Werte annehmen:

0

Der Verzeichnisname wird nicht überprüft.

1

Überprüft den Verzeichnisnamen nur, wenn bei der Suche nach der Datei debian/changelog ein Wechsel des Verzeichnisses notwendig ist.

2

Der Verzeichnisname wird in jedem Fall überprüft.

-adebian-architecture, -tGNU-system-type

Beeinflussen den Typ der Architektur, für die das Paket gebaut wird.

--check-dirname-level N

Wie unter „Überprüfen von Verzeichnisnamen“ beschrieben.

--check-dirname-regex N

Wie unter „Überprüfen von Verzeichnisnamen“ beschrieben.

--no-conf, --noconf

Es werden keine Konfigurationsdateien eingelesen. Dies muss als erste Option auf der Kommandozeile angegeben werden.

--help, --version

Zeigt eine kurze Hilfe und die Versionsinformationen zu diesem Programm.

Die Konfigurationsvariablen DEVSCRIPTS_CHECK_DIRNAME_LEVEL und DEVSCRIPTS_CHECK_DIRNAME_REGEX können die folgenden Werte annehmen:

0

Der Verzeichnisname wird nicht überprüft.

1

Überprüft den Verzeichnisnamen nur, wenn bei der Suche nach der Datei debian/changelog ein Wechsel des Verzeichnisses notwendig ist.

2

Der Verzeichnisname wird in jedem Fall überprüft.

--cleandebs

Entfernt die Dateien *.deb, *.changes und *.build in allen Verzeichnissen. Die Dateien .dsc, .diff.gz und (.orig).tar.gz werden nicht verändert, so dass die aktuelle Version eines Paketes in jedem Fall wiederhergestellt werden kann. Weiterhin wird die Datei .upload nicht verändert, damit debchange korrekt funktioniert.

--nocleandebs

Die Option --nocleandebs verhindert das zuvor beschriebene Verhalten. Voreingestellt ist, dass keine Dateien gelöscht werden. debclean nutzt das Kommando debuild zum Bereinigen des Verzeichnisbaumes.

--check-dirname-level N, --check-dirname-regex regex

Siehe Überprüfen von Verzeichnisnamen.

--no-conf, --noconf

Ignoriert alle Konfigurationsdateien. Wird diese Option verwendet, so muss sie als erste Option auf der Kommandozeile angegeben werden.

--help

Zeigt eine kurze Hilfe an.

--version

Zeigt die Versionsnummer des Programmes an.

debclean nutzt zwei Konfigurationsdateien /etc/devscripts.conf und ~/.devscripts. Diese werden in der genannten Reihenfolge ausgewertet, die folgenden Variablen werden dabei genutzt:

DEBCLEAN_CLEANDEBS

Das Setzen dieser Variablen auf „yes“ führt dazu, dass die Kommandozeilenoption --cleandebs gesetzt wird.

DEVSCRIPTS_CHECK_DIRNAME_LEVEL und DEVSCRIPTS_CHECK_DIRNAME_REGEX

Diese Variablen werden generell von den Programmen im Paket devscripts verwendet.

5.5.11.1 Überprüfung von Verzeichnisnamen

debi installiert die aktuelle Version eines Debian Pakets. Die Syntax für den Aufruf lautet:

```
debi [options] [changes file] [package ...]
```

debi ermittelt die aktuelle Version eines Debian Pakets und installiert dieses. Wird eine .changes-Datei auf der Kommandozeile angegeben, so wird versucht, den Paketnamen aus diesem Dateinamen zu ermitteln. Ist dies nicht der Fall, so muss debi innerhalb des Source-Code-Verzeichnisses aufgerufen werden. In diesem Fall versucht debi, eine .changes-Datei in dem Verzeichnis zu finden, das der aktuellen Paketversion entspricht (die notwendigen Angaben werden aus der .changes-Datei übernommen).

Danach wird debpkg -i für jedes in der .changes-Datei aufgeführte Paket ausgeführt. Dabei wird davon ausgegangen, dass alle Pakete im gleichen Verzeichnis liegen wie die Datei .changes.

Werden auf der Kommandozeile eine Anzahl von Paketen übergeben, so wird versucht, ausschließlich diese Pakete zu installieren.

Wie viele andere Skripte aus dem Paket devscripts versucht auch debchange, rekursiv im Verzeichnisbaum die Datei debian/changelog zu finden. Wird diese Datei

gefunden, so wird überprüft, ob das darüber liegende Verzeichnis dem Paketnamen entspricht. Wie dies genau durchgeführt wird, kann durch die beiden Konfigurationsvariablen DEVSCRIPTS_CHECK_DIRNAME_LEVEL und DEVSCRIPTS_CHECK_DIRNAME_REGEX sowie deren Kommandozeilen-Gegenstücke --check-dirname-level bzw. --check-dirname-regex beeinflusst werden.

Die Variable DEVSCRIPTS_CHECK_DIRNAME_LEVEL kann dabei die folgenden Werte annehmen:

0

Der Verzeichnisname wird nicht überprüft.

1

Überprüft den Verzeichnisnamen nur, wenn bei der Suche nach der Datei debian/changelog ein Wechsel des Verzeichnisses notwendig ist.

2

Der Verzeichnisname wird in jedem Fall überprüft.

-adebian-architecture, -tGNU-system-type

Beeinflussen den Typ der Architektur, für die das Paket gebaut wird.

--check-dirname-level N

Wie unter „Überprüfen von Verzeichnisnamen“ beschrieben.

--check-dirname-regex N

Wie unter „Überprüfen von Verzeichnisnamen“ beschrieben.

--no-conf, --noconf

Es werden keine Konfigurationsdateien eingelesen. Dies muss als erste Option auf der Kommandozeile angegeben werden.

--help, --version

Zeigt eine kurze Hilfe und die Versionsinformationen zu diesem Programm.

Die Konfigurationsvariablen DEVSCRIPTS_CHECK_DIRNAME_LEVEL und DEVSCRIPTS_CHECK_DIRNAME_REGEX können die folgenden Werte annehmen:

0

Der Verzeichnisname wird nicht überprüft.

1

Überprüft den Verzeichnisnamen nur, wenn bei der Suche nach der Datei debian/changelog ein Wechsel des Verzeichnisses notwendig ist.

2

Der Verzeichnisname wird in jedem Fall überprüft.

debsign erzeugt eine digitale Signatur mittels GPG/PGP für die Dateien .changes und .dsc eines Debian Paketes. Die Syntax dieses Kommandos lautet

```
debsign [options] [changes-file|dsc-file]
```

debsign verhält sich beim Signieren von Paketen wie dpkg-buildpackage, es wird eine unsignierte .dsc- oder .changes-Datei verwendet und digital signiert. Die Größe und die MD5-Checksumme der .dsc-Datei werden dabei neu berechnet und in der Datei .changes durch die neuen Werte ersetzt.

Wird eine .changes- oder .dsc-Datei angegeben, so wird diese signiert. Ist keine Datei angegeben, so wird die Datei debian/changelog ausgewertet, und es wird versucht, den Namen der .changes-Datei im darüber liegenden Verzeichnis zu ermitteln.

Dieses Programm wird eingesetzt, wenn ein Entwickler ein Paket auf einem System erzeugt, auf dem es nicht auf sicherem Wege signiert werden kann. Um das Paket auf einem sicheren System zu signieren, müssen lediglich die (kleinen) Dateien .dsc und .changes übertragen, signiert und wieder auf das ursprüngliche System transferiert werden. Dieser Prozess kann weitgehend automatisiert ablaufen, mit der Option -r werden die Dateien auf das lokale (sichere) System kopiert und nach dem Signieren wieder auf das ursprüngliche System transferiert. Liegen die Dateien alle auf dem lokalen

System, so können diese mittels debrsign auf ein anderes System kopiert und dort signiert werden.

Auch debsign nutzt die Einträge in den Konfigurationsdateien des Paketes devscripts.

-r [username@]remotehost

Die Dateien .changes und .dsc liegen auf dem angegebenen entfernten System, es wird SSH verwendet, um die Dateien zu kopieren.

-pprogname

progname kann den Wert pgp oder gpg annehmen, das entsprechnende Programm wird verwendet, um die Dateien zu signieren. Wird nichts angegeben, so wird gpg verwendet, falls die Datei ~/.gnupg/secring.gpg existiert. Ansonsten wird pgp eingesetzt.

-mmaintainer

Hiermit kann der Name des Paketbetreuers angegeben werden, mit dessen Namen das Paket signiert werden soll.

-emaintainer

Wie die Option -m, wird jedoch vorrangig behandelt.

-kkeyid

Die KeyID des zu verwendenden Schlüssels, diese Option überschreibt die Optionen -m und -e.

-spgp, -sgpg

Setzt die Kommandozeilenoptionen des zum Signieren verwendeten Programmes passend für GnuPG oder PGP.

-S

Sucht nach einer .changes-Datei für Quellcode-Pakete.

-adebian-architecture, -tGNU-system-type

Diese Optionen verändern das Verhalten bei der Suche nach passenden .changes-Dateien, so dass das Verhalten von dpkg-buildpackage nachgebildet wird. Genauere Informationen finden sich in der Manpage zu dpkg-architecture(1).

--no-conf, --noconf

Verhindert das Einlesen von Konfigurationsdateien. Dies muss als erste Option auf der Kommandozeile angegeben werden.

--help, -h

Zeigt eine kurze Hilfe an.

--version

Zeigt Informationen zur Programmversion und zum Copyright an.

Es werden die beiden Konfigurationsdateien /etc/devscripts.conf und ~/.devscripts in dieser Reihenfolge ausgewertet. Kommandozeilenoptionen überschreiben Werte in den Konfigurationsdateien. Eventuell gesetzte Umgebungsvariablen gleichen Namens werden ignoriert. Die folgenden Variablen können verwendet werden:

DEBSIGN_PROGRAM

Wie die Option -p.

DEBSIGN_SIGNLIKE

Kann den Wert gpg oder pgp annehmen und bewirkt ein Verhalten wie die Optionen -sgpg und -spgp.

DEBSIGN_MAINT

Wie die Option -m.

DEBSIGN_KEYID

Wie die Option -k.

dpatch dient zur Verwaltung von Patches für Quellcode-Pakete. Damit bietet sich die Möglichkeit, Patches aus einem anderen Release einzubinden oder auch die bestehenden Patches zu organisieren. Dies Syntax von dpatch lautet:

dpatch [options] command [command-arguments]

dpatch ermöglicht es, Patches oder Templates im Verzeichnis debian/patches/ ohne weitere Organisation im Quellcode-Baum abzulegen. Um diese Patches während der Übersetzung eines Paketes einzubinden, muss lediglich ein kurzer Eintrag im Makefile des Paketes aufgenommen werden und im Abschnitt patch/unpatch bzw. im Abschnitt clean der Datei debian/rules aufgenommen werden. Für den individuellen Einsatz kann auch /usr/bin/dpatch direkt aufgerufen werden.

dpatch kennt einige globale Optionen, welche das gesamte Verhalten von dpatch verändern. Diese sind:

--workdir (-d) DIRECTORY

Ohne Angabe dieser Option verwendet dpatch das aktuelle Verzeichnis, um die gewünschten Patches anzuwenden. Wird diese Option verwendet, so wird das angegebene Verzeichnis als Arbeitsverzeichnis genutzt und auch die Patches werden in diesem Verzeichnis gesucht.

--chdir, -c

Bewirkt, dass ein alternatives Verzeichnis für die anzuwendenden Patches verwendet wird. Nur sinnvoll im Zusammenhang mit --workdir.

--strict, -S

Aktiviert den „Strict"-Modus, eine etwas genauere Auswertung von Problemen. Dies bewirkt, dass dpatch auch bei Warnungen (warnings) abbricht, statt diese einfach zu übergehen. Dies ist beispielsweise der Fall, wenn ein Patch bereits auf den Quellcode angewendet wurde.

--force, -F

Forciert die Anwendung oder das Entfernen von Patches, auch wenn diese normalerweise von dpatch übersprungen werden, beispielsweise weil ein Patch bereits angewendet wurde.

debc zeigt den Inhalt und weitere Informationen zum aktuellsten Debian Paket. debc muss innerhalb des Quellcode-Verzeichnisses aufgerufen werden, aus dem die Debian Pakete für ein Software-Projekt erzeugt werden. debc ermittelt die neueste Version des Pakets und zeigt den Inhalt und Informationen über die Abhängigkeiten an. Hier ein Beispiel:

```
fr@wasabi:~/Daten/debian-pakete/modellbahn/pythontrain/pythontrain-1.0$ ls ../ backup pythontrain_1.0-1_i386.deb pythontrain_1.0-3_i386.deb pythonTrain.gladep messages.pot pythontrain_1.0-1.tar.gz pythontrain_1.0-3.tar.gz pythonTrain.gladep.bak pixmaps pythontrain_1.0-2.dsc pythontrain_1.1-1.dsc pythonTrain-lang.c project1 pythontrain_1.0-2_i386.changes pythontrain_1.1-1_i386.changes pythonTrain.png pythontrain-1.0 pythontrain_1.0-2_i386.deb pythontrain_1.1-1_i386.deb pythonTrain.py pythontrain_1.0-1.diff.gz pythontrain_1.0-2.tar.gz pythontrain_1.1-1.tar.gz pythonTrain.xml pythontrain_1.0-1.dsc pythontrain_1.0-3.dsc pythonTrain.glade pythontrain_1.0-1_i386.changes pythontrain_1.0-3_i386.changes pythonTrain.glade.bak fr@wasabi:~/Daten/debian-pakete/modellbahn/pythontrain/pythontrain-1.0$ debc pythontrain_1.1-1_i386.deb ---------- neues Debian-Paket, Version 2.0. Größe 174430 Byte: control-Archiv= 1276 Byte. 504 Bytes, 10 Zeilen control 1181 Bytes, 16 Zeilen md5sums 561 Bytes, 15 Zeilen * postinst #!/bin/sh 160 Bytes, 5 Zeilen * postrm #!/bin/sh 413 Bytes, 14 Zeilen * prerm #!/bin/sh Package: pythontrain Version: 1.1-1 Section: misc Priority: optional Architecture: i386 Depends: python2.2, python-glade, python-gtk, python2.2-gtk2, python2.2-glade2, rcsh, glade-common-2, glade-2, gettext, localeconf, localepurge, python2.2-xmlbase, sysutils, python2.2-optik, python2.2-optik Installed-Size: 300 Maintainer: Frank Ronneburg <fr@debiananwenderhandbuch.de> Description: Python/GTK Anwendung zur Steuerung von Modellbahnen Benötigt wird weiterhin ein SRCP kompatibler Server lokal oder im Netzwerk. drwxr-xr-x
```

root/root 0 2003-07-15 01:41:35 ./ drwxr-xr-x root/root 0 2003-07-15 01:41:29 ./usr/ drwxr-xr-x root/root 0 2003-07-15 01:41:24 ./usr/bin/ -rwxr-xr-x root/root 20738 2003-07-15 01:41:24 ./usr/bin/pythonTrain drwxr-xr-x root/root 0 2003-07-15 01:41:27 ./usr/share/ drwxr-xr-x root/root 0 2003-07-15 01:41:24 ./usr/share/doc/ drwxr-xr-x root/root 0 2003-07-15 01:41:33 ./usr/share/doc/pythontrain/ drwxr-xr-x root/root 0 2003-07-15 01:41:24 ./usr/share/doc/pythontrain/html/ -rw-r--r-- root/root 3181 2003-07-15 01:41:24 ./usr/share/doc/pythontrain/html/bensoftw.html -rw-r--r-- root/root 2300 2003-07-15 01:41:24 ./usr/share/doc/pythontrain/html/index.html -rw-r--r-- root/root 6094 2003-07-15 01:41:24 ./usr/share/doc/pythontrain/html/ptrain.html -rw-r--r-- root/root 3139 2003-07-15 01:41:24 ./usr/share/doc/pythontrain/html/pythontrain.html -rw-r--r-- root/root 19486 2003-07-15 01:41:24 ./usr/share/doc/pythontrain/html/pythonTrain-main.jpg -rw-r--r-- root/root 24026 2003-07-15 01:41:24 ./usr/share/doc/pythontrain/html/tamsprg.jpg -rw-r--r-- root/root 389 2003-06-27 11:47:02 ./usr/share/doc/pythontrain/copyright -rw-r--r-- root/root 2659 2003-07-15 01:41:24 ./usr/share/doc/pythontrain/pythontrain.txt.gz -rw-r--r-- root/root 54774 2003-07-15 01:41:24 ./usr/share/doc/pythontrain/pythontrain.pdf.gz -rw-r--r-- root/root 410 2003-07-15 01:36:36 ./usr/share/doc/pythontrain/changelog.Debian.gz drwxr-xr-x root/root 0 2003-07-15 01:41:24 ./usr/share/pythonTrain/ -rw-r--r-- root/root 4173 2003-07-15 01:41:24 ./usr/share/pythonTrain/pythonTrain.xml -rw-r--r-- root/root 56642 2003-07-15 01:41:24 ./usr/share/pythonTrain/pythonTrain.glade -rw-r--r-- root/root 61113 2003-07-15 01:41:24 ./usr/share/pythonTrain/pythonTrain.png drwxr-xr-x root/root 0 2003-07-15 01:41:27 ./usr/share/doc-base/ -rw-r--r-- root/root 555 2003-06-27 16:57:23 ./usr/share/doc-base/pythontrain drwxr-xr-x root/root 0 2003-07-15 01:41:29 ./usr/lib/ drwxr-xr-x root/root 0 2003-07-15 01:41:29 ./usr/lib/menu/ -rw-r--r-- root/root 150 2003-06-27 11:47:04 ./usr/lib/menu/pythontrain

Wie zu sehen ist, sind verschiedene (auch ältere) Versionen des Pakets vorhanden. Es wird das aktuellste Paket ermittelt, und es werden Informationen zu diesem Paket ausgegeben.

checkinstall (asic-linux.com.mx/~izto/checkinstall/) ist ein Werkzeug zum Übersetzen von Quellcode-Paketen, welche make einsetzen, und zum Erzeugen von Debian, aber auch RPM- oder SLP-Paketen aus diesen Quellcode-Archiven. Dies vereinfacht die Verwaltung von Quellcode-Paketen, die auf unterschiedlichen Distributionen eingesetzt werden sollen.

checkinstall übersetzt den Quellcode eines Paketes, in dem zunächst make install (oder ein anderes, auf der Kommandozeile von checkinstall übergebenes Argument) aufgerufen wird. Das Paket wird compiliert, es wird ein Binär-Paket erzeugt, und dieses wird installiert. Das erzeugte Paket kann auf weiteren Systemen installiert werden.

checkinstall ist als Debian Paket seit dem „Sarge"-Release verfügbar und kann mittels apt-get install checkinstall installiert werden. Auf älteren Systemen ist das Paket als „Backport" verfügbar. Hierzu ist die Zeile

```
deb http://www.backports.org/debian/ woody checkinstall
```

der Datei /etc/apt/sources.list hinzuzufügen.

```
fr@wasabi:~$ sudo apt-get install checkinstall Password: Paketlisten
werden gelesen... Fertig Abhängigkeitsbaum wird aufgebaut... Fertig
Die folgenden zusätzlichen Pakete werden installiert: installwatch Die
folgenden NEUEN Pakete werden installiert: checkinstall installwatch 0
aktualisiert, 2 neu installiert, 0 zu entfernen und 0 nicht aktualisiert. Es
müssen 46,9kB Archive geholt werden. Nach dem Auspacken werden
229kB Plattenplatz zusätzlich benutzt. Möchten Sie fortfahren? [J/n]
Hole:1 ftp://ftp.de.debian.org sarge/main installwatch 0.6.3-1 [12,0kB]
Hole:2 ftp://ftp.de.debian.org sarge/main checkinstall 1.5.3-3 [34,8kB]
```

```
Es wurden 46,9kB in 2s geholt (18,7kB/s) Wähle vormals abgewähltes
Paket installwatch. (Lese Datenbank ... 57608 Dateien und
Verzeichnisse sind derzeit installiert.) Entpacke installwatch (aus
.../installwatch_0.6.3-1_i386.deb) ... Wähle vormals abgewähltes Paket
checkinstall. Entpacke checkinstall (aus .../checkinstall_1.5.3-3_all.deb)
... Richte installwatch ein (0.6.3-1) ...  Richte checkinstall ein (1.5.3-3)
... fr@wasabi:~$
```

checkinstall nutzt das Programm installwatch, welches ebenfalls installiert wird.

Im nächsten Schritt sind, falls nicht schon geschehen, die Archive mit den Quellcode-Dateien in ein eigenes Verzeichnis zu entpacken. Wechseln Sie in dieses Verzeichnis, nun sollte zunächst geprüft werden, ob der Quellcode sich problemlos übersetzen lässt. Dies geschieht mittels ./configure --sysconfdir=/etc (falls ein configure-Skript vorhanden ist) und einem anschließenden Aufruf von make. Läuft alles problemlos durch, so kann checkinstall zum Einsatz kommen.

./configure

./configure kennt die Option --help, mit der sich alle weiteren Konfigurationsoptionen anzeigen lassen. Sinnvoll ist häufig auch die Option --bindir=/usr/bin, da viele Programme sonst im Verzeichnis /usr/local/ abgelegt werden.

Üblicherweise würde nun der Aufruf von make install ein Quellcode-Paket installieren. Dies hätte jedoch zur Folge, dass das Paketmanagement auf dem System keinerlei Kenntnis von den installierten Dateien erhalten würde. Deshalb wird nun das Kommando checkinstall -D make install aufgerufen.

Am Beispiel des Quellcodes zu wget soll der Einsatz von checkinstall exemplarisch gezeigt werden.

```
fr@wasabi:/tmp/wget-1.9.1$ ./configure  --sysconfdir=/etc --
bindir=/usr/bin configuring for GNU Wget 1.9.1  checking build system
type... i686-pc-linux-gnu checking host system type... i686-pc-linux-
gnu checking whether make sets $(MAKE)... yes checking for a BSD-
compatible install... /usr/bin/install -c checking for gcc... gcc ...
checking for perl5... no checking for perl... /usr/bin/perl checking for
pod2man... /usr/bin/pod2man configure: creating ./ config.status
config.status: creating Makefile config.status: creating src/Makefile
config.status: creating doc/Makefile config.status: creating
util/Makefile config.status: creating po/Makefile.in config.status:
creating windows/Makefile config.status: creating src/config.h
config.status: src/config.h is unchanged config.status: executing default
commands generating po/POTFILES from ./po/POTFILES.in creating
po/Makefile fr@wasabi:/tmp/wget-1.9.1$
```

Nun liegt ein Makefile vor, in dem bereits die Anpassungen der Verzeichnisse für die Debian Distribution vorgenommen wurden. Der Aufruf von make übersetzt den Quellcode in ein ausführbares Programm.

```
fr@wasabi:/tmp/wget-1.9.1$ make cd src && make CC='gcc'
CPPFLAGS="DEFS='-DHAVE_CONFIG_H -
DSYSTEM_WGETRC=\"/etc/wgetrc\"   -
DLOCALEDIR=\"/usr/local/share/locale\"' CFLAGS='-O2 -Wall -
Wno-implicit'   LDFLAGS='' LIBS='' prefix='/usr/local'
exec_prefix='/usr/local'   bindir='/usr/bin' infodir='/usr/local/info'
mandir='/usr/local/man' manext='1' make[1]: Entering directory
`/tmp/wget-1.9.1/src' gcc -I. -I. -I/opt/include   -DHAVE_CONFIG_H -
DSYSTEM_WGETRC=\"/etc/wgetrc\"   -
DLOCALEDIR=\"/usr/local/share/locale\" -O2 -Wall -Wno-implicit -c
cmpt.c ... make[1]: Leaving directory `/tmp/wget-1.9.1/util' cd windows
&& make CC='gcc' CPPFLAGS='' DEFS='-DHAVE_CONFIG_H -
DSYSTEM_WGETRC=\"/etc/rc\"   -
DLOCALEDIR=\"/usr/local/share/locale\"' CFLAGS='-O2 -Wall -
```

```
Wno-implicit'   LIBS='' prefix='/usr/local' exec_prefix='/usr/local'
bindir='/usr/bin'   infodir='/usr/local/info' mandir='/usr/local/man'
manext='1' make[1]: Entering directory `/tmp/wget-1.9.1/windows'
make[1]: Für das Ziel »all« ist nichts zu tun.make[1]: Leaving directory
`/tmp/wget-1.9.1/windows' fr@wasabi:/tmp/wget-1.9.1$
```

Der Aufruf von make muss ohne Fehlermeldungen durchlaufen, ansonsten ist eine Fehlersuche angesagt.

Abschließend kommt checkinstall zum Einsatz

```
wasabi:/tmp/wget-1.9.1$ checkinstall -D make install  checkinstall
1.5.3, Copyright 2001 Felipe Eduardo Sanchez Diaz Duran         This
software is released under the GNU GPL.   Installing with "make
install"...  ========================= Installation results
==========================  Copying documentation
directory... cd src && make CC='gcc' CPPFLAGS='' DEFS='-
DHAVE_CONFIG_H -DSYSTEM_WGETRC=\"/etc/wgetrc -
DLOCALEDIR=\"/usr/local/share/locale\"' CFLAGS='-O2 -Wall -
Wno-implicit' LDFLAGS='' S='' prefix='/usr/local'
exec_prefix='/usr/local' bindir='/usr/bin' infodir='/usr/local/nfo'
mandir='/usr/local/man' manext='1' install.bin make[1]: Entering
directory `/tmp/wget-1.9.1/src' ../mkinstalldirs /usr/bin /usr/bin/install -c
wget /usr/bin/wget ... /usr/bin/install -c -m 644 wget.1
/usr/local/man/man1/wget.1 make[1]: Leaving directory `/tmp/wget-
1.9.1/doc'  ========================= Installation succesful
==========================  Copying files to the temporary
directory...OK  Striping ELF binaries and libraries...OK  Compressing
man pages...OK  Building file list...OK   This package will be built
according to these values:  0 -  Maintainer: [ root@wasabi ] 1 -
Summary: [ ein erstes Paket ] 2 -  Name:    [ wget-1.9.1 ] 3 -  Version: [
1.9.1 ] 4 -  Release: [ 1 ] 5 -  License: [ GPL ] 6 -  Group:   [
checkinstall ] 7 -  Architecture: [ i386 ] 8 -  Source location: [ wget-
```

```
1.9.1 ] 9 -  Alternate source location: [  ]  Enter a number to change any
of them or press ENTER to continue:
```

Nach einiger Zeit können Informationen zu dem Paket angegeben werden. checkinstall versucht, Werte wie den Paketnamen und die Versionsnummer aus dem Verzeichnisnamen zu ermitteln. Die vorgeschlagenen Angaben können hier noch verändert werden.

```
****************************************** **** Debian
package creation selected ***
****************************************** Building Debian
package...OK  Installing Debian package...OK  Erasing temporary
files...OK  Writing backup package...OK  Deleting temp dir...OK
**************************************************************
************ Done. The new package has been installed and saved
to /tmp/wget-1.9.1/wget-1.9.1_1.9.1-1_i386.deb  You can remove it
from your system anytime using:       dpkg -r wget-1.9.1
**************************************************************
************
```

Abschließend wird das erzeugte Paket installiert.

Lintian (Homepage: lintian.debian.org/) untersucht Debian Pakete auf Fehler, die nicht der Debian Policy für Pakete (www.debian.org/doc/debian-policy/) entsprechen. Darüber hinaus werden die Pakete auch auf einige andere, häufig vorkommende Fehler hin geprüft.

Lintian kann Debian Binär- und Quellcode sowie „udeb“-Pakete (diese werden vom Debian Installer genutzt) untersuchen. Es wird eine Datenbank mit den Ergebnissen gepflegt; diese Datenbank wird „labratory“ genannt. Mit dieser Datenbank können mehrere Überprüfungen eines Pakets beschleunigt werden, da sehr aufwändige Operationen aus der Datenbank gelesen werden können.

lintian wird wie folgt aufgerufen:

```
lintian [action] [options] [packages] ...
```

Dabei kann das zu prüfende Paket auf drei verschiedene Arten angegeben werden: mit dem Dateinamen (paketname.deb), als Paketname, oder es kann die .changes-Datei eines Pakets angegeben werden. Wird der Paketname verwendet, so muss die Variable LINTIAN_DIST in der Konfigurationsdatei auf einen sinnvollen Wert gesetzt werden. lintian durchsucht dann den in dieser Variablen festgelegten Pfad nach Binär- oder Quellcode-Paketen, auf die der angegebene Name passt (die Optionen --binary, --udeb, --source können gesetzt werden, falls nur Pakete eines bestimmten Typs gesucht werden sollen).

Wird eine .changes-Datei angegeben, so werden alle in der Datei aufgeführten Dateien geprüft. Dies ist vor dem Upload von Dateien auf einen Server notwendig.

lintian kennt einige Optionen, von denen jeweils nur eine je Programmaufruf auf der Kommandozeile angegeben werden darf. Diese werden als Aktionen bezeichnet.

-S, --setup-lab

Initialisiert die Datenbank.

-R, --remove-lab

Löscht das Verzeichnis mit der Datenbank.

-c, --check

Führt alle bekannten Überprüfungen auf dem angegebenen Paket durch. Diese Aktion wird auch durchgeführt, wenn keine Aktion angegeben wird.

-C chk1,chk2,..., --check-part chk1,chk2,...

Es werden nur die angeführten Überprüfungen durchgeführt. Hierbei kann der Name des Skriptes oder die Bezeichnung angegeben werden.

-X chk1,chk2,..., --dont-check-part chk1,chk2,...

Führt alle Überprüfungen mit Ausnahme der aufgeführten durch; hierbei kann der Name des Skriptes oder die Bezeichnung angegeben werden.

-u, --unpack

Entpackt die gewünschten Pakete bis zum angegebenen Level. Der voreingestellte Level ist 1.

-r, --remove

Säubert das lintian-Verzeichnis für die angegebenen Verzeichnisse bis zum aktuellen Unpack-Level (siehe -u oder --unpack). Der voreingestellte Level ist hier 0.

-h, --help

Zeigt eine Hilfe zu lintian an.

-v, --verbose

Gibt mehr Informationen aus, die zur Fehlersuche dienen können.

-V, --version

Zeigt die Versionsnummer des Programms an.

-d, --debug

Zeigt Fehlermeldungen an, funktioniert nur zusammen mit -v.

-i, --info

Gibt detaillierte Informationen zu den Policy-Verletzungen aus.

-I, --display--info

Gibt auch die Informations-Tags (I:) mit aus, die normalerweise unterdrückt werden.

-l n, --unpack-level n

Setzt den Unpack-Level auf den Wert n.

-o, --no-override

Benutzt keine Override-Dateien.

--show-overrides

Zeigt Einträge an, die mit den Informationen aus einer Override-Datei überschrieben wurden.

-U info1,info2,..., --unpack-info info1,info2,...

Sammelt die angegebenen Informationen, auch wenn diese von den Tests nicht benötigt werden.

-m, --md5sums

Überprüft die MD5-Checksummen aller Dateien, wenn eine .changes-Datei angegeben wird. Normalerweise werden nur .dsc-Dateien hinsichtlich der Checksumme überprüft.

--allow-root

Zeigt keine Warnung an, falls lintian mit Administratorrechten gestartet wird.

--cfg configfile

Liest die angegebene Konfigurationsdatei statt der Standard-Konfigurationsdateien. Diese Option überschreibt die Umgebungsvariable LINTIAN_CFG.

--lab labdir

Setzt ein neues Verzeichnis für das Lintian-„Labratory“. In diesem Verzeichnis werden die Informationen über die überprüften Pakete gehalten. Dies überschreibt die Umgebungsvariable LINTIAN_LAB und den gleichnamigen Eintrag in der Konfigurationsdatei.

--archivedir archivedir

Pfad zum Verzeichnis mit den zu untersuchenden Dateien. Diese Option kann genutzt werden, um einen kompletten Verzeichnisbaum mit Paketen statt eines einzelnen Pakets von Lintian überprüfen zu lassen. Dies überschreibt die Umgebungsvariable LINTIAN_ARCHIVEDIR und den gleichnamigen Eintrag in der Konfigurationsdatei.

--section release

Wird die Option --distdir genutzt, so kann mit dieser Option die Suche auf Pakete aus einer bestimmten Sektion (beispielsweise main) begrenzt werden. Dies überschreibt die Umgebungsvariable LINTIAN_SECTION und den gleichnamigen Eintrag in der Konfigurationsdatei.

--arch arch

Wird die Option --distdir genutzt, so kann mit dieser Option die Suche auf Pakete für eine bestimmte Architektur (beispielsweise alpha) begrenzt werden. Dies überschreibt die Umgebungsvariable LINTIAN_ARCH und den gleichnamigen Eintrag in der Konfigurationsdatei.

--root rootdir

Sucht nach Lintian-Skripten (Test-Skripten und Analyse-Skripten) im angegebenen Verzeichnis statt in /usr/share/lintian/. Dies überschreibt die Umgebungsvariable LINTIAN_ROOT.

-a, --all

Prüft alle Pakete der Distribution. Hierzu muss die Variable LINTIAN_DIST in der Konfigurationsdatei gesetzt sein.

-b, --binary

Bei den im Folgenden auf der Kommandozeile gelisteten Paketen handelt es sich um Binär-Pakete.

-s, --source

Bei den im Folgenden auf der Kommandozeile gelisteten Paketen handelt es sich um Quellcode-Pakete.

--udeb

Bei den im Folgenden auf der Kommandozeile gelisteten Paketen handelt es sich um udeb-Pakete.

-p, --packages-file file

Analysiert alle in der Datei angegebenen Pakete. Jedes Paket muss hierzu in einer einzelnen Zeile in der Datei stehen, und die Zeile muss das folgende Format haben:

```
type package version file
```

Hierbei ist type das Zeichen b (Binär-Paket) oder s (Source/Quellcode-Paket). package ist der Paketname, version ist die Versionsnummer des Pakets, und file ist der absolute Pfad und der Dateiname des Pakets.

binaries (bin)

Sucht nach Fehlern in Binärdateien und Objekt-Dateien.

changelog-file (chg)

Prüft changelog-Dateien in einem Binär-Paket.

conffiles (cnf)

Prüft, ob die control-Datei eines Binär-Pakets fehlerfrei ist.

control-file (dctl)

Prüft die Datei debian/control in einem Quellcode-Paket.

copyright-file (cpy)

Prüft, ob eine Copyright-Datei in einem Binärpaket vorhanden ist.

cruft (deb)

Sucht nach unerwünschten Dateien in einem Paket. Solche Dateien sind beispielsweise Dateien von einer Versionskontrolle oder temporäre Dateien aus einem Übersetzungsvorgang.

deb-format (dfmt)

Prüft, ob das Paket mit einer fehlerhaften Version von tar erzeugt wurde. Dies kann zu Problemen bei der Installation mit dpkg führen.

debconf (dc)

Sucht in Paketen nach Fehlern, die oft bei der Verwendung von debconf gemacht werden.

debdiff (dif)

Sucht nach Dateien mit der Endung .orig oder .rej in den .diff-Dateien.

debian-readme (drm)

Prüft, ob die Datei README.Debian ein unverändertes Template ist, das von debmake erzeugt wurde.

debhelper (dh)

Sucht nach üblichen Fehlern in den Quellcode-Paketen mittels debhelper.

description (des)

Prüft, ob das Feld Description in der Beschreibung eines Binär-Pakets den Richtlinien des Policy-Manuals entspricht.

etcfiles (etc)

Prüft, ob alle Konfiurationsdateien im Verzeichnis /etc/ eines Pakets als Konfigurationsdateien ausgewiesen sind.

fields (fld)

Prüft die Kontrollfelder eines Binär- oder Quellcodepaketes.

files (fil)

Prüft, ob ein Binärpaket den Paketrichtlinien hinsichtlich Dateitypen, Zugriffsrechten und Eigentumsrechten entspricht.

huge-usr-share (hus)

Prüft, ob ein Architektur-abhängiges Paket viel Platz im Verzeichnis /usr/share/ beansprucht. Dies führt zu einer (meistens) unnötigen Belastung der Spiegel-Server.

infofiles (info)

Prüft, ob die Info-Dateien eines Pakets der Policy entsprechen.

init.d (ini)

Prüft, ob die Init-Dateien (in /etc/init.d/) eines Pakets der Policy entsprechen.

manpages (man)

Prüft, ob die Manpages eines Pakets der Policy entsprechen.

md5sums (md5)

Prüft, ob die Checksumme eines Pakets existiert und gültig ist.

menus (men)

Prüft, ob in einem Binär-Paket die Menüdateien der Policy entsprechen.

menu-format (mnf)

Prüft, ob die Syntax der Menüdateien korrekt ist.

perl (prl)

Prüft die enthaltenen Perl-Skripte auf verwendete Perl-Module und prüft, ob die notwendigen Abhängigkeiten definiert sind.

po-debconf (pd)

Prüft auf Fehler bei der Verwendung von po-debconf.

scripts (scr)

Prüft bei Shell-Skripten die Zeile #!.

shared-libs (shl)

Prüft in einem Binär-Paket die Shared-Libraries in Hinblick auf ihre Konformität zur Debian Policy.

spelling (spl)

Prüft die Datei copyright und das Feld Description auf häufig vorkommende Schreibfehler.

standards-version (std)

Prüft in einem Quellcode-Paket, ob die Datei debian/control ein gültiges Feld für die Debian Standards-Version enthält.

changelog-file

Kopiert die Datei changelog eines Paketes in das Lintian-Verzeichnis.

copyright-file

Kopiert die Datei copyright eines Paketes in das Lintian-Verzeichnis.

debfiles

Sammelt alle Dateien, die zu dem Quellcode-Paket eines Debian Paketes gehören.

debian-readme

Kopiert die Datei README.Debian eines Paketes in das Lintian-Verzeichnis.

diffstat

Speichert die Ausgabe des Programmes diffstat für die an einem Quellcode-Paket vorgenommen Änderungen in Bezug auf die Debian Distribution.

doc-base-files

Kopiert die Dateien aus dem Verzeichnis /usr/share/doc-base in das Lintian-Verzeichnis, unterhalb von doc-base.

file-info

Erzeugt und speichert die Informationen des Kommandos file für jede Datei in einem Binär-Paket.

init.d

Kopiert die Skripte aus dem Verzeichnis /etc/init.d in das Lintian-Verzeichnis, unterhalb von init.d.

md5sums

Erzeugt MD5-Checksummen für alle Dateien in einem Binär-Paket.

menu-files

Kopiert die Dateien aus dem Verzeichnis usr/share/doc/menu eines Binär-Paketes in das Menüverzeichnis des Lintian-Verzeichnisses.

objdump-info

Speichert Informationen des Kommandos objdump für jede Datei in einem Binär-Paket.

override-file

Kopiert die Override-Datei eines Paketes in das Lintian-Verzeichnis.

scripts

Speichert Informationen über die verwendeten Skripte in einem Binär-Paket.

source-control-file

Speichert Informationen über das Binär-Paket aus den Informationen in der Datei debian/control.

Der Einsatz von Lintian ist, trotz der vielen möglichen Optionen, recht einfach. Hier ein kleines Beispiel für ein sehr nachlässig gebautes Paket des Autors:

```
fr@wasabi:~/Daten/debian-pakete/modellbahn/pythontrain$ lintian
pythontrain_1.1-1_i386.deb E: pythontrain: binary-without-manpage
pythonTrain E: pythontrain: python-script-but-no-python-dep
./usr/bin/pythonTrain W: pythontrain: unquoted-string-in-menu-item
/usr/lib/menu/pythontrain needs:2 W: pythontrain: unquoted-string-in-
menu-item /usr/lib/menu/pythontrain section:2 E: pythontrain: menu-
icon-not-in-xpm-format /usr/share/pixmaps/gnome-diskfree.png E:
pythontrain: debian-changelog-file-uses-obsolete-national-encoding at
line 7 E: pythontrain: package-has-a-duplicate-relation depends:
python2.2-optik, python2.2-optik
```

Bei der Ausgabe unterscheidet Lintian zwischen Fehlern (E: Error) und Warnungen (W: Warning). Diese erscheinen in der ersten Spalte. Danach wird der Paketname angezeigt. Dies ist notwendig, wenn mehrere Pakete in einem Durchlauf geprüft werden. Danach folgt eine Infomation zu dem gefundenen Problem mit einer Angabe, welche Datei, Zeile oder welche Abhängigkeit betroffen ist.

Lintian sucht an den folgenden Orten nach Konfigurationsdateien:

Im über die Option --cfg angegebenen Verzeichnis.

Im in der Variablen $LINTIAN_CFG angegebenen Pfad.

In $LINTIAN_ROOT/lintianrc.

In $HOME/.lintianrc.

In /etc/lintianrc.

Lintian nutzt die folgenden Verzeichnisse:

/usr/share/lintian/checks, hier liegen verschiedene Skripte, die die Pakete überprüfen.

/usr/share/lintian/collection, enthält Skripte, die Informationen über die zu prüfenden Pakete sammeln und für die spätere Verwendung mit den vorgenannten Skripten zur Überprüfung zwischenspeichern.

/usr/share/lintian/info, enthält Informationen aus externen Quellen, die von Lintian verwendet werden.

/usr/share/lintian/lib, Hilfsskripte die von Lintian-Skripten verwendet werden.

/usr/share/lintian/unpack, Skripte zur Verwaltung der Prüfumgebung.

Das Verzeichnis /usr/share/lintian kann über die Umgebungsvariabe LINTIAN_ROOT überschrieben werden, alternativ kann auch die Option --root verwendet werden.

Je nachdem, ob die Ausführung von Lintian erfolgreich war, wird ein entsprechender Exit-Status zurückgegeben, mögliche Werte hierfür sind:

0 - Es wurden keine Verletzungen der Policy gefunden. Warnungen werden ausgegeben, aber nicht als Problem bewertet.

1 - Es wurden Verletzungen der Policy festgestellt.

2 - Lintian-Laufzeitfehler, es wird zusätzlich eine Nachricht auf der Standardausgabe angezeigt.

5.5 Erstellen, prüfen und verwalten von Debian Paketen

5.4 Debian Pakete - Aufbau der Sourcen 5.6 Package-Dateien

5.6 Package-Dateien

5.5 Erstellen, prüfen und verwalten von Debian Paketen 5.7 mini-dinstall

Die selbst erstellten Pakete können natürlich einfach mit dpkg -i paket.deb installiert werden; eleganter ist es jedoch, die Pakete komplett in die Debian Paketverwaltung einzubinden, so dass alle Informationen über das Paket mittels der Paketverwaltungstools verfügbar sind. Sie können so auch Pakete im Netz bereitstellen, die dann bei Veränderungen automatisch auf allen Systemen, die darauf zugreifen, aktualisiert werden.

Die notwendigen Informationen sind natürlich in den erzeugten Paketen enthalten und müssen aus diesen extrahiert werden. Auf dem Server, der die Pakete zur Verfügung stellt, stehen die gesammelten Informationen in der Datei Packages. Diese wird bei einem Update durch apt oder dselect eingelesen.

Zum Erzeugen einer Packages-Datei dient das Programm dpkg-scanpackages. Wenn Sie beispielsweise Pakete für die Gruppe „misc“ erzeugt haben, so lautet der Aufruf wie folgt:

```
dpkg-scanpackages misc override.woody.main.gz
dists/woody/main/binary-all/ > Packages
```

Die Datei override.woody.main.gz finden Sie ebenfalls auf dem Debian FTP-Server im Verzeichnis /indices/. Wenn kein kompletter Verzeichnisbaum für die Pakete existiert, so kann der Pfad - wie hier gezeigt - auf der Kommandozeile angegeben werden.

Um die Pakete nun auf einem Server bereitzustellen, muss noch ein entsprechender Eintrag in die Datei /etc/apt/sources.list eingefügt werden. Eine passende Kombination stellt folgendes Beispiel dar:

```
% cd /wo/die/pakete/liegen % dpkg-scanpackages . /dev/null >
Packages && gzip -9c Packages > Packages.gz
```

Dieses erzeugt zunächst die Datei Packages und eine mit gzip komprimierte Version dieser Datei. Der passende Eintrag für die Datei sources.list müsste dann wie folgt aussehen:

```
deb http://www.mein.server.name.domain/wo/die/pakete/liegen ./
```

5.6 Package-Dateien

5.5 Erstellen, prüfen und verwalten von Debian Paketen 5.7 mini-dinstall

5.7 mini-dinstall

5.6 Package-Dateien 5.8 Debian Repositories

mini-dinstall ist ein Programm, mit dem sich sehr einfach Debian Repositories aus lokal vorhandenen Paketen erzeugen lassen.

Zunächst sind die Pakete mini-dinstall und dput zu installieren. mini-dinstall nutzt die Konfigurationsdatei .mini-dinstall.conf im Heimatverzeichnis des aktuellen Benutzers.

Hier ein Beispiel für eine bereits angepasste Konfiguration:

```
[DEFAULT] architectures = all, i386 archivedir =
/home/fr/privatarchiv/ use_dnotify = 0 verify_sigs = 1 extra_keyrings =
~/.gnupg/pubring.gpg mail_on_success = 0 archive_style = flat
poll_time = 40 mail_log_level = NONE generate_release = 1
release_description = Frank Ronneburg's Unofficial Packages
[unstable]
```

Zusätzlich ist noch das Programm dput zu konfigurieren, die Konfigurationsdatei .dput.cf liegt ebenfalls im Heimat-Verzeichnis des Benutzers.

```
[local] fqdn = localhost method = local incoming =
/home/fr/privatarchiv/mini-dinstall/incoming allow_unsigned_uploads
= 0
```

Nun werden die eigenen Pakete in das Archiv kopiert:

```
dput local meinpaket_1.0-0.1.changes
```

Abschließend sind der Datei /etc/apt/sources.list Einträge für das lokale Archiv hinzuzufügen. Sinnvoll sind hier Zeilen für Binär- und Quellcode-Pakete.

```
deb file:/home/fr/meinarchiv unstable/ deb-src file:/home/fr/meinarchiv unstable/
```

5.7 mini-dinstall

5.6 Package-Dateien 5.8 Debian Repositories

5.8 Debian Repositories

5.7 mini-dinstall 5.9 Upload von Paketen

Ein Debian Repository ist eine Sammlung von Debian Paketen, die durch einige zusätzliche Dateien so organisiert sind, dass aus diesem Verzeichnisbaum heraus Pakete installiert werden können. Dies wird durch einige (wenige) zusätzliche Dateien ermöglicht. Dies sind im Wesentlichen Index-Dateien und Dateien mit Checksummen zu den Paketen. Fügt ein Anwender ein Repository seiner Datei /etc/apt/sources.list hinzu, so können alle darin enthaltenen Dateien mit den verfügbaren Werkzeugen leicht eingesehen oder installiert werden.

Repositories können sowohl online, auf einem FTP- oder HTTP-Server, als auch offline, beispielsweise auf CD-ROM, abgelegt werden und bestehen mindestens aus einem Verzeichnis mit Debian Paketen sowie einer Datei Packages.gz (bei einem Binär-Repository) oder Sources.gz (bei einem Quellcode-Repository).

In der Datei Packages.gz sind zu jedem Paket Informationen über den Paketnamen, die Version, die Größe sowie eine Kurz- und Langbeschreibung, die Abhängigkeiten zu anderen Paketen und einige andere Informationen abgelegt. Diese Informationen können mit den Programmen zur Installation (wie beispielsweise dselect, oder auch aptitude) angezeigt werden.

In der Datei Sources.gz finden Sie Informationen über den Namen, die Version und die Abhängigkeiten zum Übersetzen des Quellcodes („build dependencies“) zu jedem Paket. Diese Informationen werden von apt-source und anderen Programmen genutzt.

Die Datei Release, die nicht zwingend vorhanden sein muss, enthält einige Informationen über das Repository, die für das „Pinning“ (siehe „APT Pinning“) von Paketen benötigt werden. Das Erzeugen einer Release-Datei ist in „Verwaltung des Spiegels“ und „Erzeugen von Release-Dateien“ beschrieben.

Neben dieser vereinfachten Form von Repositories besteht auch die Möglichkeit, Repositories für unterschiedliche Hardware-Architekturen anzulegen. APT wird dann automatisch die für die verwendete Architektur passenden Pakete finden und installieren. Weiterhin können Pakete zu Gruppen zusammengefasst werden, beispielsweise main, contrib und non-free.

Aus Benutzer- und Administratorsicht gibt es zwei Arten von Repositories: Einfach zu benutzende Repositories sind für den Administrator recht aufwändig einzurichten. Repositories mit einer einfachen Struktur sind dagegen für den Benutzer schwieriger zu handhaben. Für Repositories, die Pakete für verschiedene Architekturen bereitstellen sollen, sind die komplexen Repositories vorzuziehen. Diese sind für den Benutzer einfacher zu handhaben, erfordern aber vom Administrator einen erhöhten Aufwand.

Die Verzeichnisstruktur eines komplexen (oder auch „automatischen“) Repository stellt sich für die verschiedenen Debian Architekturen und Bereiche wie folgt dar:

```
(Wurzelverzeichnis des Repositories) | +-dists | |-stable | |-main | |
|-binary-alpha | | |-binary-arm | | |-binary-... | | +-source | |-contrib
| | |-binary-alpha | | |-binary-arm | | |-binary-... | | +-source | +-non-
free | |-binary-alpha | |-binary-arm | |-binary-... | +-source | |-
testing | |-main | | |-binary-alpha | | |-binary-arm | | |-binary-... | |
+-source | |-contrib | | |-binary-alpha | | |-binary-arm | | |-binary-...
| | +-source | +-non-free | |-binary-alpha | |-binary-arm | |-
binary-... | +-source | +-unstable |-main | |-binary-alpha | |-
binary-arm | |-binary-... | +-source |-contrib | |-binary-alpha
| |-binary-arm | |-binary-... | +-source +-non-free |-binary-
alpha |-binary-arm |-binary-... +-source
```

Pakete, die einer freien Lizenz unterliegen (siehe Debian Free Software Guidelines), finden sich im Bereich „main“. Nicht-freie Software befindet sich in „non-free“, und in „contrib“ finden sich freie Pakete, die aber auf nicht-freien Paketen basieren. Debian unterstützt eine Anzahl weiterer Hardware-Architekturen, die in diesem Beispiel nicht alle aufgeführt sind.

In jedem binary-...-Verzeichnis befindet sich eine Index-Datei Packages.gz sowie eine optionale Datei Release. Die Pakete müssen nicht im Verzeichnis der Index-Datei liegen, da die Pfade zu den Paketen in der Index-Datei stehen. Dies ermöglicht es, so genannte „Package-Pools“ zu erzeugen.

Es können beliebig viele Architekturen und Bereiche mit beliebigen Namen angelegt werden.

Einfache Repositories bestehen aus einem Wurzelverzeichnis und beliebig vielen Unterverzeichnissen. Da die Benutzer sowohl das Wurzelverzeichnis als auch den relativen Pfad zu den Unterverzeichnissen angeben müssen, kann hier eine beliebige Struktur angelegt werden. So können auch alle Dateien (Index-Dateien und Pakete) im Wurzelverzeichnis liegen. Hier sehen Sie ein einfaches Repository mit zwei Unterverzeichnissen:

```
(Wurzelverzeichnis des Repositorys)  | |-binary +-source
```

Das Erzeugen der Index-Dateien für Binärpakete übernimmt das Programm dpkg-scanpackages (siehe dpkg-scanpackages) bzw. dpkg-scansources (siehe dpkg-scansources) für die Quellcode-Pakete. Beide Programme erzeugen die Ausgabe auf der Standardausgabe, so dass dieses Ergebnis in eine Datei umgeleitet werden muss. Um die entstehenden Dateien zu komprimieren, kann im gleichen Arbeitsschritt noch das Programm gzip verwendet werden. Für ein einfaches Repository können die Index-Dateien für die Binär- und Quellcode-Pakete wie folgt erstellt werden:

```
$ cd Pfad/zum/Repository $ dpkg-scanpackages binary /dev/null | gzip -9c > binary/Packages.gz $ dpkg-scansources source | gzip -9c > source/Sources.gz
```

Eine detaillierte Anleitung zu den beiden Programmen findet sich in den oben beschriebenen Abschnitten, dort wird auch auf die notwendigen Argumente eingegangen.

Debian Pakete auf Webseiten verfügbar machen

Mit einem kleinen PHP-Skript (zu finden unter svn.debian.org/wsvn/php-apt-parser/trunk/parse-apt-files.inc?op=file) ist es sehr einfach möglich, die Dateien

Packages.gz, bzw. Sources.gz zu durchsuchen und die Informationen zu den Paketen automatisch aufzubereiten. Alle gefundenen Pakete werden auf einer Webseite mit einem Link zum Download angeboten. Darüber hinaus wird zu jedem Paket die Beschreibung angezeigt.

Die Einbindung in die eigene Webseite erfolgt über die beiden Zeilen

```
include_once("parse-apt-files.inc"); parse_and_list( "Packages.gz", "Sources.gz" );
```

Werden auf der Seite Pakete für unterschiedliche Hardware-Architekturen zur Verfügung gestellt, so erfolgt die Einbindung in die Webseite wie folgt:

```
include_once("parse-apt-files.inc"); parse_and_list( Array("i386/Packages.gz",
"powerpc/Packages.gz"), "Sources.gz" );
```

Um das „Pinning“ von Paketen zu ermöglichen (siehe Pinning), muss eine Release-Datei in jedem Verzeichnis, in dem sich eine Index-Datei befindet, zur Verfügung gestellt werden. Release-Dateien sind einfache Text-Dateien mit folgendem Aufbau:

```
Archive: archive Component: component Origin: YourCompany Label:
YourCompany Debian repository Architecture: architecture
```

Zum Erzeugen dieser Dateien kann auch das in „Verwaltung des Spiegels“ beschriebene Skript verwendet werden.

Archive

Der Name der Distribution, für die die Pakete vorgesehen sind, beispielsweise „stable“, „testing“ oder auch „unstable“.

Component

Der Bereich, dem die Pakete zugeordnet sind. Dies kann „main“, „non-free“ oder „contrib“ sein.

Origin

Der Name der Person, die dieses Repository erzeugt hat.

Label

Eine beliebige Beschreibung für dieses Repository.

Architecture

Hardware-Architektur, für die die Pakete erzeugt wurden, beispielsweise „i386“, „powerpc“, „alpha“ oder auch „sparc“.

Insbesondere müssen Sie darauf achten, dass die Angaben zu Archive und Architecture korrekt sind, da diese am häufigsten für das Pinning von Paketen eingesetzt werden. Alle anderen Angaben sind eher unwichtig.

Bei komplexen Repositories führt die umfangreiche Verzeichnisstruktur schnell zur Unübersichtlichkeit und auch zur Verschwendung von Plattenplatz, da viele Pakete (beispielsweise Dokumentation) für unterschiedliche Architekturen geeignet sind. Diese Pakete müssen nicht mehrfach im Repository vorgehalten werden.

Die Lösung dieses Problems sind so genannte „Package Pools“. Ein Pool ist ein gesondertes Verzeichnis unterhalb des Wurzelverzeichnisses des Repository, in dem alle Pakete (Binaries für alle Architekturen, für alle Distributionen und Bereiche sowie die Quellcode-Pakete) abgelegt werden. Durch die Verwendung von „override-files“ und einigen Skripten können die oben genannten Probleme umgangen werden. Ein gutes Beispiel für die Verwendung eines solchen Pools ist sicherlich das Debian Repository selbst.

Um die notwendigen Index-Dateien für ein solches Repository zu erzeugen, sollten Sie den Abschnitt zu apt-ftparchive aus dem Paket apt-utils zu Rate ziehen. Für einfache Repositories ist apt-move eine gute Hilfe.

5.7 mini-dinstall 5.9 Upload von Paketen

5.9 Upload von Paketen

5.8 Debian Repositories 5.10 Debian Package Tags (debtags)

5.9.1 Debian Policies

5.9.2 Sponsored Uploads

Um zu erreichen, dass ein Paket in die offizielle Debian Distribution aufgenommen wird, müssen vier Bedingungen erfüllt sein:

Die Software muss im Debian Paketformat vorliegen.

Ein Debian Entwickler muss das Paket auf einem Debian Server der Distribution hinzufügen.

Die Software muss den Debian Richtlinien für freie Software (siehe DFSG) entsprechen.

Die Verantwortlichen für die Debian FTP-Server müssen das Paket akzeptieren.

Nicht-Entwickler Upload

Die Webseite mentors.debian.net/ bietet auch Nicht-Debian Entwicklern die Möglichkeit, Pakete auf einem Server verfügbar zu machen.

Bevor mit den Arbeiten zu einem neuen Paket begonnen wird, sollte im Debian Bug Tracking System (BTS, bugs.debian.org/) ein neuer ITP-Bug eröffnet werden. Dies bedeutet, dass ein Paket mit dem Status „Intend To Package“ (www.debian.org/devel/wnpp/being_packaged) in der Datenbank aufgenommen wird. Mit diesem Schritt wird das Risiko minimiert, dass bereits ein anderer Entwickler am gleichen Paket arbeitet. So werden doppelte Arbeiten vermieden.

Natürlich sollte auch die Liste der Pakete für die bereits ein ITP-Bug eröffnet wurde, durchgesehen werden. Eventuell arbeitet ja bereits jemand an diesem Paket.

Pakete für den privaten Gebrauch zu erzeugen, ist ein sehr einfacher Prozess. Um jedoch das Paket in die Debian Distribution aufnehmen zu lassen, sind eine Reihe von Punkten und Verfahren zu beachten. Konformität zu den Debian Standardts und Übereinstimmung mit dem Debian Policy Manual (www.us.debian.org/doc/debian-policy/) gewährleisten, dass das Paket den hohen Qualitätsansprüchen der Debian Community entspricht.

Im Debian Policy Manual ist festgelegt, wie der Name eines Paketes bestimmt wird, in welchem Abschnitt der Distribution das Paket abgelegt wird und wie verschiedene Teile der Installation behandelt werden. Es gibt weitere Dokumentation wie beispielsweise die „perl-policy“ (www.debian.org/doc/packaging-manuals/perl-policy/) oder die „menu-policy“ (www.debian.org/doc/packaging-manuals/menu-policy/) in den Perl-Paketen bzw. die Handhabung von Menüeinträgen behandelt werden.

Der komplette Neueinstieg in alle diese Vorschriften ist sicher sehr kompliziert, nach ein oder zwei einfachen Paketen wird aber Vieles klarer. Eine sinnvolle Referenz für den ersten Einstieg ist der „Debian New Maintainer Guide“ (www.us.debian.org/doc/maint-guide/), in welchem das Erstellen von Debian Paketen anhand einiger Beispiele erklärt wird.

Nachdem das gewünschte Paket erzeugt wurde, ist der wichtigste Teil der Arbeiten abgeschlossen. Wie das Paket nun auf einen offiziellen Debian Server gelangt, ist sehr davon abhängig, ob das Paket von einem Debian Entwickler erzeugt wurde oder nicht. Ein Debian Entwickler kann das Paket direkt auf einem Debian Server ablegen. Ist man kein Debian Entwickler, plant aber, in Zukunft diesen Status zu erreichen, so besteht die Möglichkeit, mit einem anderen Debian Entwickler einen so genannten Sponsor zu finden. Dieser kann das Paket prüfen und auf dem Debian Server ablegen.

Normalerweise werden neue Entwickler vom Debian-Projektteam aufgenommen und laden anschließend die von ihnen entwickelten Pakete auf einen Debian Server. Dies können sowohl komplett neue Pakete sein, welche noch nicht in der Debian Distribution enthalten sind, es kann sich aber auch um verwaiste (orphaned) Pakete handeln, welche aktuell keinen Betreuer (Maintainer) haben. Da es ausschließlich Debian Entwicklern erlaubt ist, Pakete auf den Servern abzulegen, müssen Pakete von Nicht-Entwicklern von einem Debian Entwickler „gesponsert“ werden.

Ein geeigneter Debian Entwickler kann auf der Mailingliste „debian-mentors“ gefunden werden. Die Leser dieser Liste sind spezialisiert darauf, neue Debian Entwickler beim Durchlaufen des Prozesses zum Entwicklerstatus zu begleiten. Die Mail an diese Liste sollte eine Beschreibung des Paketes enthalten, um das Interesse der Debian Entwickler zu wecken. Weiterhin ist eine URL anzugeben, unter der die Binär- und Quellcodepakete zu finden sind. Besonders wichtig sind die Quellcode-Pakete, da ein potenzieller Sponsor natürlich einen Einblick in das Paket bekommen möchte.

5.9 Upload von Paketen

5.8 Debian Repositories 5.10 Debian Package Tags (debtags)

5.10 Debian Package Tags (debtags)

5.9 Upload von Paketen 5.11 Debian Spiegel

5.10.1 debtags

5.10.2 debtags-edit

Mit dem Start der Entwicklung für die Debian Version „Etch“ wurden schon sehr frühzeitig die sogenannten „Package Tags“ eingeführt. Die Package-Tags erlauben es, Meta-Daten jedem Debian Paket hinzuzufügen. Diese Daten sollen die Suche nach Paketen oder Paketgruppen erleichtern.

Ursprünglich waren Debian Pakete in verschiedenen Sektionen (Abschnitten) sortiert. Jedes Paket konnte dabei genau einer Sektion zugeordnet werden. Dies erwies sich leider als wenig zweckmäßig, da beispielsweise ein Paket wie ein Mailprogramm mit GNOME- oder KDE-Oberfläche in die Sektion Mail oder GNOME bzw. KDE passen würde. Auch erlaubt diese Einteilung keine feinere Abstuffung, welche oft wünschenswert ist. So ist es nicht nur sinnvoll, Informationen darüber zu bekommen, für welchen Einsatzzweck ein Paket brauchbar ist (Mail), sondern auch in welcher Umgebung es lauffähig ist (X11).

Das Ergebnis dieser Einteilung nach Sektionen war, dass viele Leute einfach neue Sektionen eingeführt haben, um Pakete differenzierter zu organisieren. Verschiedentlich wurde bereits vorgeschlagen, mit „Package Tags“ einen universellen und flexiblen Weg einzuschlagen, um Pakete zu klassifizieren. Dies wurde mit der Entwicklung von „Etch“ umgesetzt.

Die Erweiterung für „Package Tags“ wurde direkt in die Paketbeschreibungen (Package-Files) der Distribution eingefügt. Tags sind nach der ausführlichen Beschreibung eines Paketes zu finden, hier am Beispiel des Paketes tidy:

```
fr@surimi:~$ apt-cache show tidy Package: tidy Priority: optional
Section: web Installed-Size: 40 Maintainer: Jason Thomas Architecture:
i386 Version: 20050415-1 Depends: libc6 (>= 2.3.2.ds1-21), libtidy0
Suggests: tidy-doc Filename: pool/main/t/tidy/tidy_20050415-
1_i386.deb Size: 17020 MD5sum:
983571c271b64f93b01903f56479a70d Description: HTML syntax
checker and reformatter  Corrects markup in a way compliant with the
latest standards, and  optimal for the popular browsers.  It has a
comprehensive knowledge  of the attributes defined in the HTML 4.0
recommendation from W3C,  and understands the US ASCII, ISO
Latin-1, UTF-8 and the ISO 2022  family of 7-bit encodings.  In the
output:  .  * HTML entity names for characters are used when
```

appropriate. * Missing attribute quotes are added, and mismatched quotes found. * Tags lacking a terminating '>' are spotted. * Proprietary elements are recognized and reported as such. * The page is reformatted, from a choice of indentation styles. . Tidy is a product of the World Wide Web Consortium. Tag: interface::commandline, use::checking, role::sw-utility, format::html, devel

Wie an diesem Beispiel zu sehen ist, enthält die „Tag“-Zeile Informationen über das Benutzerinterface (Kommandozeile), den Zweck (Überprüfung), die Role (Hilfsprogramm) sowie das Dateiformat (HTML). Mit Hilfe dieser Informationen ist es möglich, sehr detailiert nach Paketen für verschiedene Einsatzzwecke oder Benutzerschnittstellen zu suchen.

Da die Package-Tags in den Package-Dateien (und nicht in den eigentlichen Debian Paketen) enthalten sind, stehen die Informationen bereits vor dem Download und der Installation eines Paketes zur Verfügung.

Um in den Package-Tags Informationen zu suchen, ist es notwendig, das Programm debtags zu installieren. Eine direkte Suche über apt-get oder aptitude ist zurzeit noch nicht implementiert. Als weiteres Programm sollte debtags-edit installiert werden. Mit diesem Programm kann ebenfalls in den Package-Tags gesucht werden, zusätzlich ist es möglich, Packages-Tags zu verändern.

Die Suche mittels debtags kann nun beispielsweise dazu genutzt werden, Pakete zu suchen, die eine Relation zu bash haben.

fr@wasabi:~$ debtags related bash bash3 - The GNU Bourne Again SHell (Version 3)

Ein interessanteres Ergebnis bringt die Suche nach Relationen zu IMAP Mail hervor:

fr@wasabi:~$ debtags grep mail::imap mutt: application, interface::text-mode, made-of::lang-c, mail::imap, mail::pop ,

protocol::imap, protocol::ipv6, protocol::pop, role::sw-client, uitoolkit::nc urses, works-with::mail nail: interface::commandline, interface::shell, mail::imap, mail::list, mail::p op, mail::smtp, protocol::imap, protocol::pop, protocol::smtp, role::sw-client, special::completely-tagged, use::transmission, works-with::mail cyrus21-imapd: interface::daemon, mail::filters, mail::imap, network::service, protocol::imap, protocol::ipv6, role::sw-server, works-with::mail imapproxy: interface::daemon, mail::imap, protocol::imap, use::proxying squirrelmail: interface::web, made-of::lang-php, mail::imap, protocol::imap, wo rks-with::mail getmail4: mail::imap, mail::pop, protocol::imap, protocol::pop, protocol::ssl

Webbasierte Tag-Suche

Unter debian.vitavonni.de/packagebrowser/index.cgi steht ein Frontend für die Suche nach Paketen und deren Tags zur Verfügung.

Eine Übersicht aller Package-Tags, die sich mit dem Thema Mail befassen, liefert folgendes Kommando:

fr@wasabi:~$ debtags tagsearch mail mail::TODO - Need an extra tag mail::filters - Filters mail::imap - Mail access via IMAP mail::list - Mailing Lists mail::notification - Notification mail::pop - Mail access via POP3 mail::smtp - Mail transfer via SMTP media::mail - Email protocol::pop - Mail access via POP3 protocol::smtp - SMTP Simple Mail Transport Protocol works-with::mail - Email

Das Program debtags verwaltet auf der Kommandozeile die Informationen zu Package-Tags und kann einfache Suchoperationen auf Basis dieser Daten durchführen. Die Syntax von debtags lautet:

```
debtags [options] [command] [args...]
```

Package-Tags-Informationen werden aus verschiedenen Quellen in das System eingebunden. Ähnlich wie bei Informationen zu den Debian Paketen verwaltet die Datei /etc/debtags/sources.list die Quellen mit den Informationen zu den Tags eines Paketes. Grundlegende Informationen sind aber auch bereits in den Paketdateien (Packages.gz) der Distribution (ab der Version „Etch“) enthalten. Damit sind diese Informationen zu Package-Tags auch in der APT Datenbank verfügbar. Die Informationen zu Tags in den offiziellen Package-Dateien können nicht verändert werden, es ist auch nicht vorgesehen, lokale Anpassungen vorzunehmen. Die Aktualisierung wird mit dem Kommando debtags update durchgeführt, dieses muss mit Administratorrechten ausgeführt werden. Das Programm debtags kann auch für einfache Abfragen der APT Datenbank genutzt werden, die notwendigen Optionen hierfür lauten tagshow und tagsearch.

Eine Package-Tag-Datenbank kann mit der Option check überprüft werden, vergleichbare Pakete oder Gruppen lassen sich mit der Option related ermitteln, und die kompletten Informationen können mit der Option tagcoll auf der Standardausgabe ausgegeben werden.

debtags benötigt bei jedem Aufruf eine Option, die die durchzuführende Aktion beschreibt, möglich sind dabei die im Folgenden beschriebenen Aktionen:

update

Lädt die aktuelle Tag-Datenbank aus dem Netz und aktualisiert die lokale System-Datenbank. Diese Aktion benötigt Administrator-Rechte.

check [filename]

Prüft, ob zu allen Tags in der Datenbank eine passende Beschreibung vorhanden ist. Wird keine optionale Datei angegeben, so wird die zentrale Datenbank überprüft.

tagshow tag

Zeigt die gespeicherten Informationen zu einem Tag.

tagsearch pattern [pattern [pattern [...]]]

Zeigt eine Zusammenfassung aller angegebenen Tags.

show package

Nutzt die Ausgabe von apt-cache show package und ergänzt diese um die Tag-Informationen.

related pkg1[,pkg2[,pkg3,[...]]]

Zeigt Pakete, die mit den angegebenen in Verbindung stehen.

cat

Gibt die kompletten Informationen der Tag-Datenbank aus.

search [-v] [-q] tag expression

Gibt die Paketnamen und Beschreibungen aus, auf die der angegebene reguläre Ausdruck zutrifft.

grep [-v] [-q] tag expression

Gibt alle Zeilen der Tag-Datenbank aus, auf die der Suchbegriff zutrifft.

install [-v] [-q] tag expression

Führt apt-get install mit den Namen der Pakete aus, auf die der reguläre Ausdruck zutrifft.

mkpatch [filename]

Gibt die Unterschiede zwischen der aktuellen Tag-Datenbank und der angegebenen Datenbank auf der Standardausgabe aus.

maintainers

Erstellt eine Liste aller Maintainer und der Pakete, zu denen diese Personen Tags beigesteuert haben.

tag add package tags

Fügt einen oder mehrere Tags dem angegebenen Paket hinzu.

tag rm package tags

Löscht einen oder mehrere Tags zu dem angegebenen Paket.

tag ls package

Zeigt die Tags zum angegebenen Paket an.

submit [patch]

Verschickt den angegebenen Patch an die zentrale Tag-Datenbank. Wird keine Patch-Datei angegeben, so werden die Änderungen zur lokalen Datenbank ermittelt und verschickt.

todo

Zeigt alle lokal installierten Pakete an, zu denen noch keine Package-Tags definiert wurden.

score

Bewertet alle nicht installierten Pakete daraufhin, wie oft deren Tags in Paketen erwähnt werden, die bereits installiert sind.

facetcoll

Zeigt die Sammlung von Tags, bei der jedes Paket nur mit den „Facetten“ ausgegeben wird. Dies ist sinnvoll, um die Ausgabe mittels tagcoll weiter zu bearbeiten.

stats

Zeigt einige statistische Informationen zu Debtags an.

todoreport

Zeigt eine Zusammenfassung zu den Paketen an, die noch einer Überarbeitung bedürfen.

-d, --distance

Bestimmt die maximale Abweichung zu einer „related“ Option.

-v, --invert-match

Invertiert die Suche, im Zusammenhang mit grep, so dass ausschließlich nicht auf das Suchkriterium passende Begriffe gefunden werden.

-q, --quiet

Gibt keinerlei Informationen auf der Standardausgabe aus, der Rückgabewert 0 zeigt jedoch an, dass ein passender Eintrag gefunden wurde.

-g, --groups-items

Gruppiert, im Zusammenhang mit dem „cat“ Kommando, die Tags, die einen identischen Inhalt haben.

--verbose

Eine Option für Entwickler zur Fehlersuche, gibt mehr Informationen über den Programmablauf aus.

-V, --version

Zeigt die aktuelle Version des Programmes debtags an und beendet dann das Programm.

-?, --help

Zeigt Informationen zu den Kommandozeilenoptionen von debtags an.

TIPP

Um selbst erstellte Pakete mit eigenen, noch nicht definierten Tags zu versehen, können der Datei /etc/debtags/sources.list weitere Einträge hinzugefügt werden. Diese Einträge verweisen dann auf die selbst erstellten Tags. Hierzu sind die folgenden Schritte notwendig:

Es muss die Datei /etc/debtags/personaltags/vocabulary erzeugt werden. Eventuell ist auch das entsprechende Verzeichnis noch anzulegen. Die Datei enthält die neuen Tags mit Beschreibung, beispielsweise

Facet: personal Description: Personal preference Tag: personal::essential Description: I cannot live without it Tag: personal::useful Description: Tried it and found it useful Tag: personal::bad Description: Tried it and did not like it Tag: personal::interesting Description: It looks interesting, but I have not tried it yet

Die Datei /etc/debtags/personaltags/tags-current enthält die Namen der Tags und die Zuordnung zu den Paketen, beispielsweise

```
mmv: personal::essential mc: personal::essential xdiskusage: personal::essential buffy:
personal::useful debtags: personal::interesting
```

Beide Dateien müssen mit gzip komprimiert werden (gzip -9 dateiname).

Die neue Quelle muss der Datei /etc/debtags/sources.list hinzugefügt werden. Der Eintrag sieht wie folgt aus:

```
tags file:/etc/debtags/personaltags/
```

Abschließend wird das Kommando debtags update aufgerufen.

Hier einige Beispiele für den Einsatz von debtags:

debtags update

Aktualisiert die debtags -Datenbank.

debtags show mutt

Zeigt Paket- und Tag-Informationen zum Paket mutt.

debtags -d 7 related mutt

Zeigt die mit mutt in einer Beziehung stehenden Pakete mit einer maximalen Entfernung von 7 an.

debtags -d 5 related galeon, mozilla-browser

Zeigt Pakete mit einer Beziehung zu galeon und mozilla-browser mit einer Tiefe von maximal 5 an.

debtags tagsearch mail

Durchsucht die Tag-Datenbank nach Paketen, die mit dem Schlagwort „mail“ zu tun haben.

debtags search "use::editing && media::rasterimage && \! (role::aux-shlib || role::aux-dummy)"

Listet alle Pakete auf, die Bilddateien („rasterimage“) bearbeiten können, ausgenommen werden Dummy-Pakete und Shared-Libraries.

debtags search 'works-with::mail && role::sw:client'

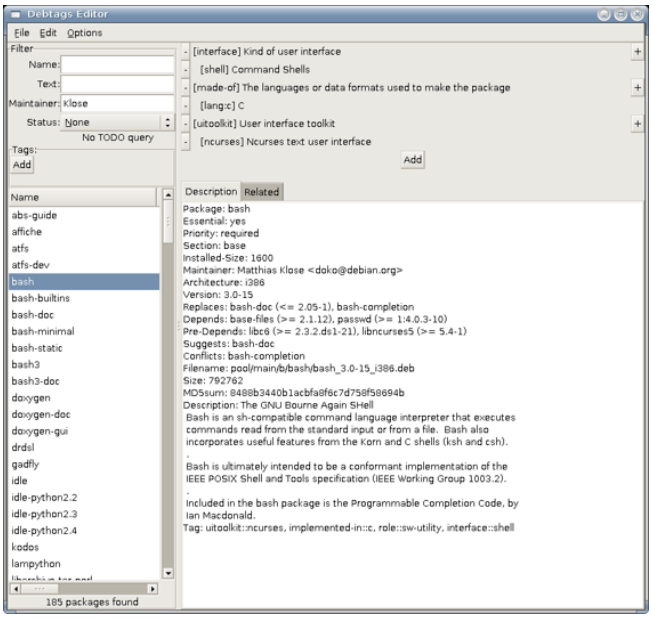
Zeigt alle Mail-Clients an.

debtags install 'protocol::irc && role::sw:client'

Installiert alle Pakete, die IRC-Clients zur Verfügung stellen.

debtags-edit ist ein Programm mit grafischer Benutzeroberfläche, welches sich zum Suchen und Verändern von Informationen in der APT- und Debtags-Datenbank eignet. Ziel dieses Programmes ist es, möglichst vielen Leuten zu ermöglichen, an Debian Tags mitzuarbeiten, ohne komplizierte Kommandozeilen Programme einsetzen zu müssen. Die Syntax von debtags-edit lautet:

debtags-edit [options]



Wird debtags-edit aufgerufen, so werden zunächst die APT- und Debtags-Datenbanken gelesen. Ist die Umgebungsvariable DEBEMAIL gesetzt, so geht das Programm davon aus, dass nach Paketen passend zu dieser Mail-Adresse gesucht werden soll, dies ist für Debian Entwickler sinnvoll. Ist diese Variable nicht gesetzt, so wird eine Liste aller installierten Pakete angezeigt. Das Anklicken eines Paketes erlaubt es, zu diesem Paket Tag-Informationen hinzuzufügen oder auch zu löschen.

Das Programm kann die Änderungen in einer Patch-Datei im aktuellen Verzeichnis ablegen. Alle Änderungen werden auch beim Beenden des Programmes automatisch gespeichert.

debtags-edit kennt die drei Kommandozeilenoptionen --help, --version und --verbose.

5.10 Debian Package Tags (debtags)

5.9 Upload von Paketen 5.11 Debian Spiegel

5.11 Debian Spiegel

5.10 Debian Package Tags (debtags) 5.12 CD- und DVD-Images herunterladen

Ein eigener Debian Spiegel (englisch „Mirror“), also eine lokale Kopie aller benötigten Dateien vom Debian FTP-Server, kann für verschiedene Zwecke sinnvoll sein. Zunächst einmal hat man sofort Zugriff auf alle Dateien und kann diese installieren, ohne auf den Download der Dateien warten zu müssen. Natürlich ist auch die Aktualität der Software um einiges besser als bei der Installation von CD-ROMs. Für die Installation der Software auf mehreren Rechnern ist eine solche lokale Kopie ebenfalls die erste Wahl. Wenn, wie im nächsten Abschnitt beschrieben, eigene Debian CDs erstellt werden sollen, ist ein Spiegel unumgänglich.

Aufgrund des bereits großen Umfangs des Debian Archivs ist es nicht sinnvoll, eine komplette Kopie aller Dateien lokal vorzuhalten. Meist ist es ausreichend, lediglich die Binaries für die verwendete Architektur (beispielsweise i386) vorzuhalten. Die Sourcen der Pakete werden nur in seltenen Fällen benötigt und können bei Bedarf gezielt besorgt werden. Informationen zum Spiegeln eines Debian Servers finden Sie auch unter http://www.debian.org/mirror/ftpmirror.

Das Programm debmirror bietet eine sehr einfache Möglichkeit, einen Debian FTP-Server - bei Bedarf auch nur in Teilen - zu spiegeln. Das Programm steht als offizielles Debian Paket zur Verfügung.

debmirror benutzt keine Konfigurationsdatei; alle benötigten Parameter können auf der Kommandozeile angegeben werden. Sehr vorteilhaft ist bei diesem Skript, dass es speziell auf die Bedürfnisse von Debian angepasst wurde; so kann es beispielsweise mit Package-Pools problemlos umgehen. Das Package-Pool-Konzept wurde mit der Entwicklung von „woody“ eingeführt und erlaubt es, verschiedene Releases (testing, stable, unstable usw.) auf Basis der Daten innerhalb des Verzeichnisses pool/ zu verwalten. Weiterhin werden die benötigten Dateien auf Basis der Package-Dateien ermittelt; somit ist auch hier immer ein einheitlicher Stand sichergestellt.

Die Verwendung von debmirror ist ziemlich simpel. Wenn das Programm ohne weitere Parameter aufgerufen wird, bekommen Sie eine Übersicht der möglichen Parameter:

```
sushi:~# ./debmirror  mirrordir not specified Usage: ./debmirror
mirrordir [--debug] [--progress] [--source|--nosource]          [--
md5sums|--nomd5sums] [--passive] [--host=remotehost]          [--
user=remoteusername]          [--root=directory] [--dist=foo[,bar,..] ...]
[--section=foo[,bar,..] ...] [--arch=foo[,bar,..] ...]          [--cleanup|--
nocleanup] [--skippackages] [--adddir=directory]    [--ignore=regex] [--
getcontents] [--exclude=regex] [--help]
```

Als einziger Parameter ist „mirrordir“ zwingend erforderlich. Dieser beschreibt das lokale Verzeichnis, in dem der Debian Spiegel erzeugt werden soll. Wenn dieses Verzeichnis nicht vorhanden ist, so wird es erzeugt.

Folgendes Kommando erzeugt einen Debian Spiegel für die Architektur„i386“ im Verzeichnis /home/ftp/debian/; es werden die Versionen „potato“, „woody“ und „sid“ mit den Bereichen „main“, „contrib“ und „non-free“ kopiert. Als Server dient „ftp.debian.org“ (hier sollte auf alle Fälle ein gut erreichbarer Server benutzt werden). Es werden keine Sourcen übertragen, und das Verzeichnis „debian-non-US“ auf der lokalen Festplatte wird nicht angetastet (dazu gleich mehr). Weiterhin wird der Fortschritt laufend angezeigt.

```
debmirror -a i386 -s main -s contrib -s non-free \  -h ftp.debian.org \  -d
woody -d potato -d sid \  /home/ftp/debian \  --nosource \  --
ignore=debian-non-US/ --progress
```

debmirror auf Sarge-Systemen

ⓘ

Mit dem Release von „Sarge“ können die Optionen für„non-US“ entfallen, die Pakete sind in den „main“-Zweig integriert worden.

Da auf dem offiziellen Debian FTP-Server die Pakete aus dem Non-US-Bereich fehlen, können diese Pakete mit einer zweiten Zeile auf den heimischen Rechner geholt werden:

```
debmirror -a i386 -h ftp.de.debian.org \  -r /debian-non-US -s non-
US/main,non-US/contrib,non-US/non-free \  -d woody -d potato -d sid \
/home/ftp/debian/debian-non-US --progress
```

Je nach Bandbreite Ihrer Internetverbindung steht Ihnen nach einiger Zeit ein lokaler Debian Spiegel zur Verfügung.

Im Folgenden wird das Erstellen eines partiellen Spiegels beschrieben. Bei einem partiellen Spiegel wird nur ein ausgewählter Teil von Paketen im lokalen Spiegel abgelegt. Wolfgang Borgert stellt auf seinen Webseiten einige Skripte vor, mit denen ein solcher Spiegel erzeugt werden kann. Dabei wird auf die Programme apt, apt-move und debootstrap (siehe apt-move und debootstrap) zurückgegriffen. Die Abhängigkeiten aller in der Liste angegebenen Pakete werden dabei aufgelöst, so dass ein vollständig funktionsfähiger Spiegel entsteht. Durch die Verwendung von debootstrap wird das zugrunde liegende Basissystem nicht verändert, alle Dateien werden in einer chroot-Umgebung abgelegt.

In den folgenden Beispielen wird die chroot-Umgebung unter /var/local/chroot/ erzeugt, innerhalb dieses Verzeichnisses wird der Spiegel mit den Debian Paketen im Verzeichnis/var/local/mirrors/ angelegt. Der Spiegel wird über einen symbolischen Link aus der chroot-Umgebung heraus leichter zugänglich gemacht:

```
ln -s /var/local/chroot/var/local/mirrors/ /var/local/mirrors/
```

Das folgende Skript dient zum Erzeugen des Spiegels. Es sollte als Datei create-mirror abgelegt werden.

```
#!/bin/sh # generate a partial Debian mirror with debootstrap and apt-move # 2004-02, W. Borgert <debacle@debian.org> # Released under the terms of the GNU General Public License  # the root directory of the mirror system CHROOT=/var/local/chroot # the mirror directory inside of the chroot MIRRORDIR=/var/local/mirrors/debian # the Debian archive next to you DEBIAN=http://ftp.de.debian.org/debian ####  OUT=adduser,aptitude,apt-utils,at,base-config,bsdmainutils,\ console-common,console-data,console-tools,cpio,cron,debconf-i18n,\ dhcp-client,ed,exim4,exim4-base,exim4-config,exim4-daemon-light,\ fdutils,gettext-base,groff-base,ifupdown,info,ipchains,iptables,\ iputils-ping,klogd,libconsole,libgcrypt1,libgcrypt7,libgdbm3,\ libgnutls7,libgpg-error0,libident,liblocale-gettext-perl,\ liblockfile1,liblzo1,libnewt0.51,libopencdk8,libpcap0.7,libpcre3,\ libpopt0,libsigc++-1.2-5c102,libssl0.9.7,libtasn1-0,\ libtext-charwidth-perl,libtext-iconv-perl,libtext-wrapi18n-perl,\ libwrap0,lilo,logrotate,mailx,makedev,man-db,manpages,mbr,modconf,\ modutils,nano,netbase,netkit-inetd,net-tools,nvi,pciutils,ppp,\ pppconfig,pppoe,pppoeconf,procps,psmisc,setserial,slang1,sysklogd,\ syslinux,sysvinit,tasksel,tcpd,telnet,wget,whiptail  IN=apt-move,bc,bzip2,dash,debconf-english,libbz2-1.0,libpam0g,\ libreadline4 ####  apt-get install debootstrap mkdir $CHROOT debootstrap --exclude="$OUT" --include="$IN" sarge $CHROOT $DEBIAN
```

Im Wesentlichen wird hier debootstrap eingesetzt, um ein minimales Debian System zu erzeugen. Dabei werden einige wenige Pakete hinzugefügt (über die Variable IN) und eine ganze Reihe Pakete nicht installiert, die normalerweise von debootstrap verwendet werden (Variable OUT). Dies führt zu einem Basissystem von lediglich circa 80 MByte Größe (bei Verwendung des Debian Release „Sarge“).

Dieses Skript wird nur einmalig - bei der ersten Initialisierung des Spiegels - benötigt.

Zunächst ist die Datei /var/local/chroot/etc/apt/sources.list anzupassen. Als erster Eintrag muss das lokale Verzeichnis erscheinen, weitere Einträge verweisen auf externe Debian Spiegel.

```
deb file:/var/local/mirrors/debian testing main deb
http://ftp.de.debian.org/pub/debian/ testing main
```

Nun sind noch einige Variablen in der Konfigurationsdatei von apt-move (/etc/apt-move.conf) anzupassen, insbesondere APTSITES, LOCALDIR, DIST, PKGCOMP und CONTENTS. Hier sehen Sie ein Beispiel:

```
APTSITES="ftp.de.debian.org" LOCALDIR=/var/local/mirrors/debian
DIST=testing PKGCOMP="none gzip" CONTENTS=yes
```

Nun können zunächst die Paketinformationen für den Spiegel aktualisiert werden; danach werden die (in der chroot-Umgebung) installierten Pakete geholt.

```
chroot /var/local/chroot apt-get update chroot /var/local/chroot apt-
move get
```

Leider reicht eine Verzeichnisstruktur mit Debian Paketen nicht aus. Wir benötigen noch Packages- und Release-Dateien. Um Letztere zu erzeugen, dient folgendes Skript, das als make-release im Verzeichnis /var/local/chroot/root/ gespeichert wird.

```
#!/bin/sh # generate Release files for a Debian mirror # 2003-10, W. Borgert <debacle@debian.org> # Released under the terms of the GNU General Public License  ORIGIN="Debian" LABEL="Debian" SUITE="testing" CODENAME="sarge" DESC="Debian Testing distribution - Not Released" UNPACK=0 ARCHIVE=.  # long options not yet implemented args=`getopt 'a:c:d:l:o:s:u' $*` for o     do case "$o" in         -a | --arch*) ARCHIVE="$2"; shift; shift;;         -c | --code*) CODENAME="$2"; shift; shift;;         -d | --desc*) DESC="$2"; shift; shift;;         -l | --label) LABEL="$2"; shift; shift;;         -o | --orig*) ORIGIN="$2"; shift; shift;;         -s | --suite) SUITE="$2"; shift; shift;;         -u | --unpa*) UNPACK=1; shift;;         --) shift; break;;     esac done ####  cd $ARCHIVE DATE=`date -u` if [ $UNPACK -eq 1 ]; then PACKAGGZ=`find . -name Packages.gz -o -name Sources.gz`     for p in $PACKAGGZ; do         gunzip -c $p > `dirname $p`/`basename $p .gz`     done fi  PACKAGES=`find . -name Packages -o -name Packages.gz -o \     -name Sources -o -name Sources.gz` DIRS=`for p in $PACKAGES; do dirname $p; done|sort -u` COMPS=`for d in $DIRS; do echo $d | \     sed 's,[^A-Za-z0-9-]*\([A-Za-z0-9-]*\)/.*,\1,'; done|sort -u` ARCHS=`for d in $DIRS; do echo $d | \     sed 's,.*binary-\([A-Za-z0-9-]*\),\1,'; done|sort -u`  for d in $DIRS; do     cd $d     ARCH=`echo $d | sed 's,.*binary-\([A-Za-z0-9-]*\),\1,'`     COMP=`echo $d | \             sed 's,[^A-Za-z0-9-]*\([A-Za-z0-9-]*\)/.*,\1,'`         echo "Archive: $SUITE Component: $COMP Origin: $ORIGIN Label: $LABEL Architecture: $ARCH" > Release     cd - done  rm -f Release MD5FILES=`find . -name Packages -o -name Packages.gz -o \     -name Sources -o -name Sources.gz -o -name Release \     | sed 's,^[^/]*/,,'`  echo "Origin: $ORIGIN Label: $LABEL Suite: $SUITE Codename: $CODENAME Date: $DATE Architectures:" `echo $ARCHS`" Components:" `echo $COMPS`" Description: $DESC MD5Sum:" > Release  for m in $MD5FILES; do echo -n " " >> Release     SIZE=`wc -c $m | sed 's/ *\([0-9]*\).*/\1/'`     SUM=`md5sum $m | sed 's/^ *\([A-Fa-f0-9]*\) .*/\1/'`     printf "%s %16d %s\n" $SUM $SIZE $m >> Release done  echo "SHA1:" >> Release for m in $MD5FILES; do     echo -n " " >> Release     SIZE=`wc -c $m | sed 's/ *\([0-9]*\).*/\1/'`     SUM=`sha1sum $m | sed
```

```
's/^ *\([A-Fa-f0-9]*\) .*/\1/'`      printf "%s %16d %s\n" $SUM $SIZE
$m >> Release done
```

Die Information, welche Pakete in dem Spiegel vorgehalten werden sollen, findet sich in der Datei /var/local/chroot/root/packages. Ergänzungen können hier jederzeit vorgenommen werden. Die eigentliche Verwaltung des Spiegels übernimmt folgendes Skript:

```
#!/bin/sh # update a partial Debian mirror with apt-move # 2004-02, W.
Borgert <debacle@debian.org> # Released under the terms of the GNU
General Public License  # the root directory of the mirror system
CHROOT=/var/local/chroot # the mirror directory inside of the chroot
MIRRORDIR=/var/local/mirrors/debian # master packages file inside
of the chroot MASTER=/root/packages  ####  MODE=$1; shift  case
"$MODE" in     clean)        chroot $CHROOT apt-move --force delete
       exit 0        ;;     install)        PACKAGES="$*"      if [ -f
$CHROOT/$MASTER ]; then        OLDPACKAGES=`cat
$CHROOT/$MASTER`     else        OLDPACKAGES=          fi        for
p in $OLDPACKAGES $PACKAGES; do echo $p; done | \       sort -u
> $CHROOT/$MASTER    ;;     upgrade)      if [ -f
$CHROOT/$MASTER ]; then        PACKAGES=`cat
$CHROOT/$MASTER`     else        exit 0     fi     ;;     *)        echo
"Usage:"   echo "$0 clean"  echo "$0 install <packages>"        echo "$0
upgrade"   exit -1        ;; esac  chroot $CHROOT apt-get update chroot
$CHROOT apt-get install \     --option 'DPkg::Options::=--dry-run' --
yes $PACKAGES chroot $CHROOT apt-move get chroot $CHROOT
apt-move move chroot $CHROOT apt-move packages chroot
$CHROOT /root/make-release -a $MIRRORDIR chroot $CHROOT
apt-get autoclean
```

Das Hinzufügen weiterer Pakete ist recht einfach:

```
./update-mirror install dia ethereal glade-2
```

Dabei werden die Abhängigkeiten aller Pakete aufgelöst, und alle benötigten Pakete werden dem Spiegel hinzugefügt.

Wenn der Spiegel aktualisiert werden soll, so geschieht dies mit dem Kommando:

```
./update-mirror upgrade
```

Um einen Spiegel auch für andere Hardware-Architekturen bereitzustellen, ist in der Datei /var/local/chroot/etc/apt/apt.conf die Architektur hinzuzufügen:

```
APT::Architecture "arm";
```

Weiterhin ist dies auch in /var/local/chroot/etc/apt-move.conf bekannt zu machen:

```
ARCHS="arm i386"
```

debian-multimirror ist ein in Perl geschriebenes Programm, das in der Lage ist, auch einen partiellen Spiegel eines oder mehrerer Debian Server anzulegen. Die Basis für die Paketauswahl bilden dabei eine oder mehrere Listen von Paketen. Auch ist es möglich, ganze Bereiche von Paketen (beispielsweise *games* oder *kde*) auszublenden. debian-multimirror ist noch nicht als Paket in der Debian Distribution enthalten, die Software ist unter http://pedro.larroy.com/debian/debian-multimirror/ zu finden.

mirror steht als Debian Paket zur Verfügung und wird seit vielen Jahren auch auf großen FTP-Servern (beispielsweise an Universitäten) benutzt, um Inhalte anderer FTP-Server lokal zu speichern. Für eine lokale Kopie der Daten eines Debian FTP-Servers wird neben dem eigentlichen Programm mirror vor allem viel Festplattenplatz benötigt. Da mirror als universelles Programm zum Spiegeln von FTP-Servern entwickelt wurde, kennt es natürlich keine Debian Besonderheiten wie Package-Pools. Um nicht unnötig Pakete zu kopieren, ist leider eine etwas aufwändige Konfiguration nötig.

Die Konfigurationsdateien für das Programm mirror finden sich nach der Installation im Verzeichnis /etc/mirror/packages/. Hier kann für jeden gewünschten Spiegel eine Datei angelegt werden. Das Programm selbst wird dann mit der Konfigurationsdatei als Parameter aufgerufen. Zusätzlich kann die Option -d (auch mehrfach) angegeben werden, um mehr Informationen über den Vorgang zu bekommen.

Hier sehen Sie eine Konfigurationsdatei, die die Architekturen i386 und powerpc vom Debian FTP-Server kopiert.

```
package=Debian # comment=Mirror of parts of
ftp.debian.org/pub/debian # # specify remote host, directory and
ls-lR file site=ftp.de.debian.org remote_dir=/debian
store_remote_listing=/tmp/DEBIANREMOTE ls_lR_file=ls-lR.gz #
# specify the local directory local_dir=/home/ftp/debian # #
inform this user about results mail_to=fr # # Compress these files.
#compress_patt=\.*(C|c)ontents(-i386)? #compress_patt+|Packages(-
Master)?|ls-lR|md5sums #compress_patt+|Maintainers|msdos-names
# # Exclude these files or directories
exclude_patt=private/|project/|ls-lR|ls-lR.gz|ls-lR.patch.gz|dsync.list
exclude_patt+|debian-(bugs|lists)/ exclude_patt+|\.*(hurd-
```

i386|hppa|mipsel|mips|ia64|s390|arm|m68k|alpha|hurd|mips|sparc).deb?
exclude_patt+|\.*(hurd-
i386|hppa|mipsel|mips|ia64|s390|arm|m68k|alpha|hurd|mips|sparc).udeb
? exclude_patt+|\.*(hurd-
i386|hppa|mipsel|mips|ia64|s390|arm|m68k|alpha|hurd|mips|sparc).gz?
exclude_patt+|\.*(hurd-
i386|hppa|mipsel|mips|ia64|s390|arm|m68k|alpha|hurd|mips|sparc).diff.g
z? exclude_patt+|\.*(hurd-
i386|hppa|mipsel|mips|ia64|s390|arm|m68k|alpha|hurd|mips|sparc).dsc?
exclude_patt+|\.*(hurd-
i386|hppa|mipsel|mips|ia64|s390|arm|m68k|alpha|hurd|mips|sparc).tar.gz
? exclude_patt+|\.tar.gz? exclude_patt+|\.dsc?
exclude_patt+|\.changes? exclude_patt+|\.diff.gz? exclude_patt+|bo/
exclude_patt+|rex/ exclude_patt+|slink/ exclude_patt+|binary-arm/
exclude_patt+|binary-m68k/ exclude_patt+|binary-ia64/
exclude_patt+|binary-hppa/ exclude_patt+|binary-sparc/
exclude_patt+|binary-mipsel/ exclude_patt+|binary-mips/
exclude_patt+|binary-alpha/ exclude_patt+|disks-alpha/
exclude_patt+|disks-m68k/ exclude_patt+|disks-sparc/
exclude_patt+|upgrade-alpha/ exclude_patt+|upgrade-m68k/
exclude_patt+|upgrade-sparc/ exclude_patt+|binary-s390/
exclude_patt+|binary-hurd-i386/ exclude_patt+|md5sums
exclude_patt+|md5sums.gz exclude_patt+|/potato/main/disks-arm/
exclude_patt+|/potato/main/binary-arm/
exclude_patt+|/potato/contrib/binary-arm/ exclude_patt+|/potato/non-
free/binary-arm/ exclude_patt+|/potato/main/binary-sparc/
exclude_patt+|/potato/contrib/binary-sparc/
exclude_patt+|/potato/non-free/binary-sparc/
exclude_patt+|woody/non-free/binary-sparc/
exclude_patt+|woody/non-free/binary-alpha/
exclude_patt+|woody/main/binary-sparc/
exclude_patt+|woody/main/binary-alpha/
exclude_patt+|woody/contrib/binary-sparc/
exclude_patt+|woody/contrib/binary-alpha/
exclude_patt+|woody/main/binary-hurd/
exclude_patt+|woody/main/binary-hurd-i386/
exclude_patt+|woody/main/binary-arm/

```
exclude_patt+|woody/main/binary-mips/
exclude_patt+|woody/main/binary-mipsel/  exclude_patt+|binary-
hppa/  exclude_patt+|binary-sh/  exclude_patt+|woody/main/binary-
sparc/  exclude_patt+|woody/main/disks-sparc/
exclude_patt+|woody/main/binary-alpha/
exclude_patt+|woody/Contents-arm.gz
exclude_patt+|woody/Contents-hurd-i386.gz
exclude_patt+|woody/Contents-alpha.gz
exclude_patt+|woody/Contents-sparc.gz
exclude_patt+|woody/Contents-hppa.gz
exclude_patt+|woody/Contents-sh.gz  exclude_patt+|woody/Contents-
mips.gz  exclude_patt+|woody/Contents-mipsel.gz  #
exclude_patt+|\.notar  #  # Don't delete what is mirrored by the other
mirror package below  #  # delete_excl=(local|debian-non-US|non-
US|project)(/|$)  # Do not delete if more than 20% of all files would
vanish  max_delete_files=20%
```

5.11 Debian Spiegel

5.10 Debian Package Tags (debtags) 5.12 CD- und DVD-Images herunterladen

5.12 CD- und DVD-Images herunterladen

5.11 Debian Spiegel 5.13 Erstellen von eigenen Debian GNU/Linux-CD-ROMs

Das Debian-Projekt stellt ISO-Dateien von CD- und DVD-Images zum Download bereit. Wer schon einmal mehrere hundert Megabyte große Dateien aus dem Internet heruntergeladen hat, kennt sicher die Probleme mit defekten Dateien, beispielsweise durch abgebrochene Übertragungen.

Da die CD- und DVD-Medien zum größten Teil aus ganz normalen Debian Paketen bestehen, wurde ein Weg geschaffen, die ISO-Dateien aus einem kleinen Template und den einzelnen Paketen lokal auf dem System zu generieren. Dabei können die Debian Pakete selbst von jedem beliebigen Server (mit entsprechend schneller Anbindung) oder aus einem lokalen Verzeichnisbaum stammen. Solche Dateien lassen sich mittels Jigdo herunterladen und zu dem ursprünglichen Image zusammensetzen.

Eine andere Methode stellt BitTorrent zur Verfügung. Hier werden die Dateien von verschiedenen Rechnern aus einem Peer-to-Peer-Netzwerk heruntergeladen.

Jigdo (die Abkürzung steht für „Jigsaw Download“ zu Deutsch: „Kettensägen herunterladen“) ermöglicht diesen sehr effizienten Weg des Herunterladens von Image-Dateien. Grundsätzlich lassen sich alle möglichen großen Dateien mittels Jigdo übertragen; wir beschränken uns hier aber auf ISO-Images der Debian Distribution. Jigdo selbst installiert zwei Hilfsprogramme: jigdo-file, das ein Image auf dem Server zum Herunterladen vorbereitet, und jigdo-lite, mit dem der Benutzer die Dateien herunterladen kann.

Jigdo selbst erzeugt keine Images, diese müssen mittels debian-cd oder über mkisofs erstellt werden.

Zunächst muss das Paket jigdo-file installiert werden. Dies kann - wie üblich - mittels apt-get install jigdo-file erfolgen.

Die notwendigen .jigdo-Dateien und die Templates sind momentan für die „stable“- und „testing“-Releases von Debian verfügbar.

Offizielle Jigdo-Dateien für das „stable“-Release: http://cdimage.debian.org/debian-cd/current/

Für andere Architekturen finden sich die Images in den entsprechenden Verzeichnissen, der Architekturname ist entsprechend anzupassen.

Im nächsten Schritt sind nun mindestens die notwendigen Dateien für ein Image herunterzuladen. Die Dateinamen folgen dabei dieser Schreibweise:

```
distro-version-arch-n.jigdo       distro-version-arch-n.template
```

Mögliche Dateinamen wären also beispielsweise debian-31r0a-i386-binary-1.jigdo bzw. debian-31r0a-i386-binary-1.template. Die Debian Version Sarge hat momentan den Umfang von 14 CD- oder 2 DVD-Images. Für den kompletten Satz an Images müssen natürlich auch alle .jigdo- und .template-Dateien heruntergeladen werden.

Danach kann jigdo-lite benutzt werden, um das Image herunterzuladen.

```
fr@sushi:/home/fr/sarge-cd$ jigdo-lite sarge-i386-1.jigdo   Jigsaw Download "lite" Copyright 2001-2003 by Richard Atterer <jigdo@atterer.net> Loading settings from `/home/fr/.jigdo-lite'   ---------- Images offered by `sarge-i386-1.jigdo':   1: 'Debian GNU/Linux testing "Sarge" - Official Snapshot i386 Binary-1' (sarge-i386-1.iso)   Further information about `sarge-i386-1.iso': Generated on Sat, 21 Feb 2004 18:04:03 -0700   ---------- If you already have a previous version of the CD you are downloading, jigdo can re-use files on the old CD that are also present in the new image, and you do not need to download them again. Mount the old CD ROM and enter the
```

path it is mounted under (e.g. `/mnt/cdrom'). Alternatively, just press enter if you want to start downloading the remaining files. You can also enter a single digit from the list below to select the respective entry for scanning: 1: /var/cache/apt/archives/ 2: /mnt/cdrom Files to scan:

Debian CD- und DVD-Images werden von Zeit zu Zeit aktualisiert. Für die jeweils aktuelle „stable“-Version geschieht das seltener, für noch in der Entwicklung befindliche Versionen („testing“) jedoch unter Umständen mehrmals pro Woche. Jigdo bietet die Möglichkeit, bereits vorhandene Images zu aktualisieren oder Dateien von vorhandenen CDs zu übernehmen.

Ist noch ein altes Image auf der Platte vorhanden, so müssen Sie dieses via „Loopback-Device“ mounten (mount -o loop <isodatei> /cdrom). Sind die Daten noch auf einer herkömlichen CD vorhanden, so kann diese wie gewohnt in das Dateisystem eingehängt werden.

Zunächst kann ein lokaler Pfad angegeben werden, in dem nach Paketen gesucht wird. Dies kann eine bereits ins Dateisystem eingehängte CD-ROM oder DVD sein, aber auch im Cache-Verzeichnis von APT lohnt sich unter Umständen eine Suche.

Wenn an dieser Stelle lediglich die **RETURN**-Taste gedrückt wird, so werden alle benötigten Dateien von einem Server aus dem Netz geholt.

---------- The jigdo file refers to files stored on Debian mirrors. Please choose a Debian mirror as follows: Either enter a complete URL pointing to a mirror (in the form `ftp://ftp.debian.org/debian/'), or enter any regular expression for searching through the list of mirrors: Try a two-letter country code such as `de', or a country name like `United States', or a server name like `sunsite'. Debian mirror [ftp://wasabi/debian/]: ftp://ftp.de.debian.org/debian/

Sie können hier einen der offiziellen Debian Server angeben (eine Liste finden Sie unter http://www.debian.org/mirror/list) oder aber einen internen Server verwenden.

Jigdo speichert die Auswahl in der Datei ~/.jigdo-lite.

Jigdo versucht nun, alle notwendigen Pakete von dem angegebenen Server zu holen und daraus eine Image-Datei zu erstellen.

```
---------------------------------------------------------- Merging parts from `file:' URIs, if any... Found 0 of the 1175 files required by the template Copied input files to temporary file `sarge-i386-1.iso.tmp' - repeat command and supply more files to continue --11:36:53-- ftp://ftp.de.debian.org/debian/pool/main/k/kdegraphics/kpovmodeler_3.1.4-1_i386.deb => `sarge-i386-1.iso.tmpdir/ftp.de.debian.org/debian/pool/main/k/kdegraphics/kpovmodeler_3.1.4-1_i386.deb' Resolving ftp.de.debian.org... 141.76.2.4 Connecting to ftp.de.debian.org[141.76.2.4]:21... connected. Logging in as anonymous ... Logged in! ==> SYST ... done.    ==> PWD ... done. ==> TYPE I ... done.  ==> CWD /debian/pool/main/k/kdegraphics ... done. ==> PASV ... done.    ==> RETR kpovmodeler_3.1.4-1_i386.deb ... done. Length: 1,793,206 (unauthoritative)
100%[==========================================================>] 1,793,206 225.86K/s    ETA 00:00    11:37:03 (223.98 KB/s) - `sarge-i386-1.iso.tmpdir/ftp.de.debian.org/debian/pool/main/k/kdegraphics/kpovmodeler_3.1.4-1_i386.deb' saved [1793206]
```

Dieser Vorgang kann, je nach Bandbreite der Internetanbindung, einige Zeit in Anspruch nehmen. Sollten nicht alle Pakete auf dem angegebenen Server verfügbar sein, so kann später ein weiterer Server angegeben werden. Bereits übertragene Pakete werden dabei nicht erneut übertragen. Abschließend sollte eine Meldung angezeigt werden, dass das Image erfolgreich erzeugt werden konnte.

```
FINISHED --13:32:58-- Downloaded: 7,469,872 bytes in 9 files Found 9 of the 9 files required by the template                    Successfully created `sarge-i386-1.raw'    ----------------------------------------------
```

```
---------- Finished! The fact that you got this far is a strong
indication that `sarge-i386-1.raw' was generated correctly. I will
perform an additional, final check, which you can interrupt safely with
Ctrl-C if you do not want to wait.        OK: Checksums match, image is
good!
```

BitTorrent (en.wikipedia.org/wiki/Category:Linux_BitTorrent_clients) ist ein Peer-to-Peer-Programm, um Debian Images von verschiedenen Rechnern aus dem Netz herunterzuladen. Die notwendigen Dateien finden sich in dem gleichnamigen Debian Paket bittorrent.

Neben dem Programm wird noch eine so genannte Torrent-Datei benötigt. Für die Debian Images finden Sie solche Dateien unter www.debian.org/CD/torrent-cd/. Der Download erfolgt mit dem Kommandozeilen Programm btdownloadheadless und der ermittelten URL wie folgt:

```
fr@wasabi:/home/fr/$ btdownloadheadless --url \
http://cdimage.debian.org/pub/weekly/torrents/i386/sarge-i386-
1.iso.torrent   saving:         sarge-i386-1.iso (647.0 MB) percent done:
0.0 time left: download to:    /home/fr/download/sarge-i386-1.iso
download rate:  0.00 kB/s upload rate:    0.00 kB/s download total: 0.0
MiB upload total:   0.0 MiB
```

Der Fortschritt des Downloads lässt sich am Bildschirm verfolgen. Neben dem Download wird die aktuell bearbeitete Datei auch anderen Teilnehmern am BitTorrent-Netzwerk zur Verfügung gestellt, so dass auch in der Zeile „upload rate“ eine Datenübertragung festzustellen ist.

5.12 CD- und DVD-Images herunterladen

5.11 Debian Spiegel 5.13 Erstellen von eigenen Debian GNU/Linux-CD-ROMs

5.13 Erstellen von eigenen Debian GNU/Linux-CD-ROMs

5.12 CD- und DVD-Images herunterladen | Kapitel 6. Systemadministration

Debian GNU/Linux-CD-ROMs sind im Handel, als Beilage zu Zeitschriften oder zum Download als Image erhältlich. Natürlich bietet Debian Ihnen aber auch die Möglichkeit, eigene bootfähige CDs herzustellen und diese an Ihre eigenen Bedürfnisse anzupassen. Hierzu steht das Paket debian-cd zur Verfügung.

Zunächst sollten Sie jedoch dafür sorgen, dass die benötigten Dateien für die CDs auf Ihrem System vorhanden sind. Hierzu ist es üblich, eine lokale Kopie (Spiegel) der gewünschten Architektur(en) (z.B. i386) anzulegen. Das Programm mirror ist hierzu hervorragend geeignet.

Neben einigen gebräuchlichen Tools (wie zum Beispiel apt-get, perl, bash, make...), die auf jedem System vorhanden sein sollten, benötigt debian-cd die Programme mkisofs/ mkhybrid, dpkg-multicd, das Perl-MD5-Modul, dpkg-dev und lynx. Natürlich benötigen Sie auch noch einigen freien Festplattenplatz.

Nach der Installation dieses Pakets finden sich die notwendigen Dateien im Verzeichnis /usr/share/debian-cd/. In der Datei CONF.sh müssen einige Zeilen angepasst werden.

export CODENAME=woody - Hier wird die Variable für die gewünschte Version gesetzt. Es sollte bei einem kompletten Spiegel einer stabilen Version jederzeit möglich sein, die gewünschten CDs zu erstellen. Bei Entwicklerversionen können dringend benötigte Pakete fehlen, oder es sind noch keine angepassten Bootdisketten verfügbar. In einem solchen Fall kann aber auch das Paket debian-cd in einem Zustand sein, der es nicht erlaubt, CDs zu erstellen.

export DEBVERSION="2.3 r0" - Versionsnummer und Release-Stand der Distribution.

export OFFICIAL="Unofficial" - Dieser Wert sollte nicht verändert werden. Die Bezeichnung „Official“ ist nur für CD-Images erlaubt, die vom Debian-Projekt erstellt wurden. Für Vorabversionen kann hier auch die Bezeichnung „Unofficial Beta“ benutzt werden.

export ARCH=`dpkg --print-installation-architecture` - Diese Variable beschreibt, für welche Architektur die CD-Images erstellt werden sollen. Häufig wird diese

Architektur identisch mit der bereits verwendeten Architektur sein, so dass man beruhigt dem Programm dpkg die Aufgabe übertragen kann, den richtigen Wert zu ermitteln. Natürlich ist es auch möglich, CD-Images für eine andere Architektur zu erstellen. In diesem Fall ist der Wert entsprechend anzupassen und natürlich müssen die benötigten Pakete für die Zielarchitektur vorliegen. Je nach Zielarchitektur müssen ggf. noch weitere Pakete, beispielsweise aboot für die Alpha-Architektur, installiert werden.

Die nächsten vier Variablen sollten auf Verzeichnisse auf der gleichen Partition und auf dem gleichen Device zeigen. Wenn dies nicht möglich ist, kann die Variable COPYLINK auf 1 gesetzt werden. Dies benötigt einiges an zusätzlichem Festplattenplatz, da keine symbolischen Links verwendet werden können. Diese Option muss auch gesetzt sein, falls die Daten auf einem per NFS gemounteten Laufwerk liegen.

export MIRROR=/home/ftp/debian - Pfad zum Debian Spiegel. Das Verzeichnis, unterhalb dessen sich die Kopie des Debian FTP-Servers befindet.

export NONUS=/home/ftp/debian/debian-non-US - Das Verzeichnis, in dem sich die Dateien aus dem Non-US-Bereich befinden. Wenn dieser Bereich nicht auf den CDs enthalten sein soll, kann diese Zeile auskommentiert werden.

export FORCENONUSONCD1=1 - Wenn diese Zeile durch Entfernen des Zeichens # aktiviert wird, werden zwei Versionen der ersten CD erzeugt. Eine enthält alle Pakete aus non-US, die andere nicht. Diese Option ist nur sinnvoll, wenn tatsächlich beide CD-Images benötigt werden.

export TDIR=/home/ftp/debian/.tmp - Pfad zu einem Verzeichnis, in dem Dateien, die zur Erzeugung der CDs benötigt werden, abgelegt werden können.

export OUT=/home/ftp/CD-Images - Pfad, in dem die CD-Images abgelegt werden. Dieser kann sich auf einer anderen Partition befinden.

Die weiteren Optionen müssen nicht zwingend angepasst werden; trotzdem lohnt es sich, einmal einen Blick darauf zu werfen.

export APTTMP=/home/ftp/debian/.tmp/apt - Das Verzeichnis für temporäre Dateien, die von apt benötigt werden.

export NONFREE=1 - Beschreibt, ob auch die Non-Free-Teile der Distribution auf den CDs enthalten sein sollen.

export EXTRANONFREE=1 - Hiermit werden die Non-Free-Teile auf eine Extra-CD geschrieben. Bitte beachten Sie, dass nur eine der Optionen NONFREE oder EXTRANONFREE benutzt werden kann.

export LOCAL=1 - Wenn ein Verzeichnis $MIRROR/dists/$CODENAME/local/binary-$ARCH existiert und die

Dateien aus diesem Verzeichnis auf die zu erstellende CD kopiert werden sollen, so ist bei dieser Zeile das Kommentarzeichen (#) zu entfernen.

export LOCALDEBS=/home/fr/pakate/debian - Wenn die lokalen Pakete nicht in dem oben beschriebenen Verzeichnis liegen, so kann mit dieser Variablen das gewünschte Verzeichnis angegeben werden.

export BOOTDIR=/boot - Diese Variable ist zu benutzen, wenn die Dateien cd.b und second.b nicht an der üblichen Stelle zu finden sind und CD-Images für die Sparc-Architektur erzeugt werden.

export SYMLINK=1 - Mit dieser Option werden symbolische Links zu den benötigten Paketen erzeugt.

export COPYLINK=1 - Hiermit werden alle Dateien kopiert und keine symbolischen Links angelegt.

export MKISOFS=/usr/bin/mkhybrid - Hier kann der Pfad zum Programm mkhybrid angegeben werden, mit dem die ISO-Dateien erzeugt werden. Es kann auch ein anderes Programm mit ähnlicher Funktionalität, beispielsweise mkisofs, benutzt werden.

export MKISOFS_OPTS=" -a -r -T" #For normal users oder export MKISOFS_OPTS=" -a -r -F . -T" #For symlink farmers - Optionen für das Programm mkhybrid.

export VERBOSE_MAKE=1 - Gibt mehr Informationen während der Zusammenstellung der CD-Images aus.

ATTEMPT_FALLBACK=yes - Mit dieser Option kann versucht werden, eine einfachere CD zu erstellen, falls es ohne diese Option fehlschlägt.

export EXCLUDE="$BASEDIR"/tasks/exclude-potato - Liste einiger Dateien, die nicht enthalten sein sollen, in diesem Fall, um auf der ersten CD etwas Platz zu sparen.

export UNEXCLUDE2="$BASEDIR"/tasks/unexclude-CD2-potato - Hiermit werden die eben ausgeschlossenen Dateien auf die CD Nummer 2 geschrieben.

export SRCEXCLUDE="$BASEDIR"/tasks/exclude-src-potato - Liste einiger Quellpakete, die ebenfalls nicht auf den CDs erscheinen sollen.

Hier sehen Sie ein Beispiel für eine bereits angepasste Konfiguration:

This file will have to be sourced where needed # # The debian-cd dir # Where I am (hoping I'm in the debian-cd dir) export BASEDIR=`pwd` # Building potato cd set ... export CODENAME=woody # Version number, "2.2 r0", "2.2 r1" etc. export DEBVERSION="4.0 r0" # Official or non-official set. # NOTE: THE "OFFICIAL" DESIGNATION IS ONLY ALLOWED FOR IMAGES AVAILABLE # ON THE OFFICIAL DEBIAN CD WEBSITE http://cdimage.debian.org export OFFICIAL="Unofficial" #export OFFICIAL="Official" #export OFFICIAL="Official Beta" # ... for arch export ARCH=`dpkg --print-installation-architecture` # IMPORTANT : The 4 following paths must be on the same partition/device. # If they aren't then you must set COPYLINK below to 1. This # takes a lot of extra room to create the sandbox for the ISO # images, however. Also, if you are using an NFS partition for # some part of this, you must use this option. # Paths to the mirrors export MIRROR=/home/ftp/debian # Comment the following line if you don't have/want non-US export NONUS=/home/ftp/debian/debian-non-US # And this option will make you 2 copies of CD1 - one with all the # non-US packages on it, one with none. Useful if you're likely to # need both. #export FORCENONUSONCD1=1 # Path of the temporary directory export TDIR=/home/ftp/debian/.tmp # Path where the images will be written export OUT=/home/fr/mp3/IMAGES # Where we keep the temporary apt stuff. # This cannot reside on an NFS mount. export APTTMP=/home/ftp/debian/.tmp/apt # Do I want to have NONFREE merged in the CD set export NONFREE=1 # Do I want to have NONFREE on a separate CD (the last CD of the CD set) # WARNING: Don't use NONFREE and EXTRANONFREE at the same time ! # export EXTRANONFREE=1 # If you have a $MIRROR/dists/$CODENAME/local/binary-$ARCH dir with # local packages that you want to put on the CD set then # uncomment the following line # export LOCAL=1 # If your local packages are not under $MIRROR, but somewhere else, # you can uncomment this line and edit to to point to a directory # containing dists/$CODENAME/local/binary-$ARCH # export LOCALDEBS=/home/joey/debian/va/debian # Sparc only : bootdir

```
(location of cd.b and second.b) # export BOOTDIR=/boot  # Symlink farmers should uncomment this line : # export SYMLINK=1  # Use this to force copying the files instead of symlinking or hardlinking # them. This is useful if your destination directories are on a different # partition than your source files. # export COPYLINK=1  # Options # export MKISOFS=/usr/bin/mkhybrid # export MKISOFS_OPTS="-a -r -T"
        #For normal users # export MKISOFS_OPTS="-a -r -F . -T"
        #For symlink farmers  # uncomment this to if you want to see more of what the Makefile is doing export VERBOSE_MAKE=1  # uncoment this to make build_all.sh try to build a simple CD image if # the proper official CD run does not work #ATTEMPT_FALLBACK=yes  # We don't want certain packages to take up space on CD1... export EXCLUDE="$BASEDIR"/tasks/exclude-potato # ...but they're okay on other CDs  (UNEXCLUDEx == may be included on CD >= x) export UNEXCLUDE2="$BASEDIR"/tasks/unexclude-CD2-potato # Any packages listed in EXCLUDE but not in any UNEXCLUDE will be # excluded completely.  # We also exclude some source packages export SRCEXCLUDE="$BASEDIR"/tasks/exclude-src-potato
```

Für das eigentliche Erstellen der CD-Image-Dateien stehen zwei Shell-Skripts zur Verfügung. build.sh erstellt Image-Dateien für eine einzelne Architektur. Die gewünschte Architektur wird über die Konfigurationsdatei ermittelt oder kann auf der Kommandozeile übergeben werden.

build_all.sh erstellt CD-Image-Dateien für alle Architekturen.

Wenn CD-Images für eine Entwicklungsversion von Debian erstellt werden sollen, so kann es passieren, dass das Paket nicht auf dem allerneuesten Stand ist. In einem solchen Fall bietet sich ein Versuch mit einer Entwicklungsversion von debian-cd aus dem CVS-Baum an.

Eine komplette Kopie der CVS-Version kann mit folgenden Befehlen erstellt werden:

```
cvs -d :pserver:anonymous@cvs.debian.org:/cvs/debian-boot login cvs -d :pserver:anonymous@cvs.debian.org:/cvs/debian-boot -z3   checkout debian-cd cvs -d :pserver:anonymous@cvs.debian.org:/cvs/debian-boot logout
```

Im aktuellen Verzeichnis wird ein Verzeichnis debian-cd erstellt, in dem sich die neuesten Dateien befinden. Wenn diese Version aktualisiert werden soll, sind beim nächsten Aufruf folgende Kommandos notwendig:

```
cvs -d :pserver:anonymous@cvs.debian.org:/cvs/debian-boot login cvs -d :pserver:anonymous@cvs.debian.org:/cvs/debian-boot -z3 update -d -P cvs -d :pserver:anonymous@cvs.debian.org:/cvs/debian-boot logout
```

5.13 Erstellen von eigenen Debian GNU/Linux-CD-ROMs

◁ 5.12 CD- und DVD-Images herunterladen △ Kapitel 6. Systemadministration ▷

Kapitel 6. Systemadministration

5.13 Erstellen von eigenen Debian GNU/Linux-CD-ROMs 6.2 Init-Skripte

Inhaltsverzeichnis

6.1.1 LILO

6.1.2 GRUB

Vor dem Laden des Linux-Kernels wird mit dem so genannten Bootloader ein kleines Programm geladen, das Ihnen eine Auswahl zwischen verschiedenen installierten Betriebssystemen erlaubt. Die unter GNU/Linux gebräuchlichsten Bootloader sind LILO und grub auf der i386-Hardware-Architektur. Andere Architekturen verwenden Bootloader wie SILO, MILO oder auch yaboot.

Mit Debian Sarge (3.1) wurde vom bisher verwendeten Bootloader lilo zu grub gewechselt.

Auf anderen Hardware-Architekturen wie i386 kommen andere Bootloader wie beispielsweise elilo (ia64) oder yaboot (powerpc) zum Einsatz.

6.1.1.1 LILO einrichten

6.1.1.2 LILO und fremde Betriebssysteme

Bei der Installation von Debian GNU/Linux auf Ihrem System wurde bis zur Version 3.1 von Debian der Bootloader LILO auf Ihrer Festplatte installiert und so eingerichtet, dass das neue Debian GNU/Linux-System gestartet wird.

LILO (http://lilo.alioth.debian.org/) ist ein vollständiger Boot-Manager, mit dem nicht nur Linux gebootet werden kann, sondern auch jedes andere System, das sich an die im PC vorherrschenden Konventionen hält. Konfiguriert wird dieser Boot-Manager über die Datei /etc/lilo.conf.

Wann immer Sie diese Datei ändern, müssen Sie das Programm lilo auf der Kommandozeile als Superuser aufrufen, um die Änderungen tatsächlich zu übernehmen. Es reicht nicht, nur die Konfigurationsdatei zu verändern!

Wichtig in diesem Zusammenhang sind die Zeilen, die mit image und other anfangen, sowie die jeweils nachfolgende Zeile. Diese Schlüsselwörter dürfen mehrfach verwendet werden. Jede so beginnende Zeile bezeichnet ein System, das von LILO gebootet werden kann. Eine solche Partition kann einen Kernel, eine Root-Partition oder ein anderes, Nicht-Linux-System (other) beinhalten. Die Reihenfolge dieser Systeme ist entscheidend, denn das erste wird automatisch gebootet, wenn die Wartezeit (Konfigurationsoption delay in der LILO-Konfiguration) abgelaufen ist und LILO nicht durch Drücken der **SHIFT**-

Taste angehalten wurde. Nach einer Erstinstallation existiert lediglich ein einziger Eintrag, der das aktuelle Debian GNU/Linux-System bootet. Hat der Debian Installer weitere Betriebssysteme auf anderen Partitionen gefunden, so werden diese in die Bootloader-Konfiguration aufgenommen.

Um ein zweites Linux-System zu booten (z.B. unter Verwendung eines anderen Kernels), müssen Sie die Datei /etc/lilo.conf um folgende Zeilen ergänzen:

```
image=/boot/vmlinuz.neu label=neu append='mcd=0x320,11' read-only
```

Lediglich die ersten beiden Zeilen sind erforderlich. Um die Bedeutung der nachfolgenden Zeilen zu erfahren, lesen Sie bitte die Dokumentation zu LILO.

Um ein anderes System als Linux zu booten, verwenden Sie das Schlüsselwort `other` wie folgt:

```
other=/dev/hda1 label=win
```

Rufen Sie danach `lilo` als Superuser auf, um LILO neu zu installieren. Bei einem Neustart des Systems können Sie nun am Bootprompt durch Eingeben von „win“ das Betriebssystem auf der Partition `/dev/hda1` starten.

GRUB (GRand Unified Bootloader, bedeutet in deutscher Sprache „Larve“) http://www.gnu.org/software/grub/grub.html ist ein Bootloader, dessen Aufgabe es ist,

nach dem Einschalten des Rechners die weitere Kontrolle an das Betriebssystem (also zunächst den Kernel) zu übergeben. Das Betriebssystem initialisiert dann über die geeigneten Treiber alle weitere Hardware im System. GRUB ist in der Lage, viele verschiedene Betriebssysteme zu laden. Wird ein Betriebssystem nicht direkt von GRUB unterstützt, so kann dieses dennoch von GRUB über einen so genannten „Chain-Bootloader“ geladen werden. Für die wichtigsten Betriebssysteme sind diese bereits im GRUB-Paket enthalten.

Debian verwendet ab der Version 3.1 „Sarge“ GRUB als Standard-Bootloader.

Einer der größten Vorteile von GRUB ist die Möglichkeit des direkten Zugriffs auf das Dateisystem. Das bedeutet, dass GRUB nicht wissen muss, wo genau der zu ladende Kernel auf der Festplatte liegt. Sie können also jederzeit einen neuen Kernel erzeugen und müssen hinterher nicht noch einmal GRUB im MBR der Festplatte installieren. Es ist ebenfalls nicht notwendig, alle Kernel-Versionen, die sich auf der Festplatte befinden, in die Konfigurationsdatei einzutragen. Sie können mit GRUB jederzeit auf jeden Kernel auf Ihrer Festplatte zugreifen, wenn auch mit ein wenig Tipparbeit. Unterstützte Dateisysteme sind dabei Fat32, Fat16, ext2 und ReiserFS (unter Linux),UFS (BSD) und Minix.

GRUB kann durch die Unterstützung von BIOS-Erweiterungen Partitionen oberhalb der 8 GByte booten, und es ist möglich, Betriebssysteme von der 2. Festplatte zu starten. Auch kann ein Betriebssystem über das Netzwerk geladen werden.

Die Installation von GRUB unter Debian ist wie üblich mit einem einfachen apt-get install grub so gut wie abgeschlossen. Sie sollten danach zunächst das Paket lilo, falls installiert, entfernen; dieses wird nicht mehr benötigt.

Um die notwendigen Daten (die Stage-1- und -2-Bootloader) in den Masterboot-Record der Festplatte zu schreiben, benutzen Sie das Programm „grub-install“. Wenn Sie „grub-install“ zunächst ohne Optionen aufrufen, bekommen Sie eine kurze Übersicht der möglichen Optionen angezeigt:

```
debian:~# grub-install  install_device not specified. Usage: grub-install
[OPTION] install_device Install GRUB on your drive.  -h, --help
print this message and exit -v, --version          print the version
information and exit --root-directory=DIR    install GRUB images
under the directory DIR                    instead of the root directory. --
grub-shell=FILE       use FILE as the grub shell. --force-lba
force GRUB to use LBA mode even for a buggy BIOS. --recheck
probe a device map even if it already exists.
```

```
INSTALL_DEVICE can be a GRUB device name or a system device
filename.                    Reports bugs to <bug-grub@gnu.org>.
```

In den meisten Fällen ist es ausreichend, lediglich das gewünschte Device anzugeben, auf dem die Daten installiert werden sollen. Dies wird bei einer IDE-Festplatte meist /dev/hda sein.

```
surimi:/home/fr# grub-install /dev/hda Installation finished. No error
reported. This is the contents of the device map /boot/grub/device.map.
Check if this is correct or not. If any of the lines is incorrect, fix it and
re-run the script `grub-install'.   (fd0)   /dev/fd0 (hd0)    /dev/hda (hd1)
      /dev/hdb
```

GRUB ist nun funktionsfähig installiert. Sie sollten aber noch ein Menü einrichten, um nicht bei jedem Systemstart die Parameter für das Root-Device und den Kernel von Hand eingeben zu müssen. Kopieren Sie hierzu am besten das mitgelieferte Beispiel an die entsprechende Stelle.

Die Konfiguration von GRUB kann herkömmlich, mittels eines Editors, erfolgen oder aber ganz elegant mit update-grub.

ist Bestandteil des GRUB Debian Pakets. Existiert noch keine Konfigurationsdatei, so wird diese automatisch erstellt. Als Basis dienen dazu die Kernel- und RAM-Disk-Images im Verzeichnis /boot/. Zu jedem vorhandenen Kernel wird jeweils ein Eintrag für den normalen Start sowie für den Start im Single-User-Modus erstellt.

```
wasabi:~# update-grub  Searching for GRUB installation directory ...
found: /boot/grub . Testing for an existing GRUB menu.list file...
Could not find /boot/grub/menu.lst file. Would you like one  generated
```

```
for you? (y/N) y Updating /boot/grub/menu.lst ... done
Please note that configuration parameters for GRUB are stored in
/boot/grub/menu.lst . You must edit this file in order to set the options
which GRUB passes to the kernel, as well as the drive which GRUB
looks in to for the kernel.                    Everything on the line after
"kopt=" is passed to the kernel as parameters, and "groot=" must be set
to the partition(in GRUB terms, such as "(hd0,0)") which GRUB will
load the kernel from.                    After you have edited
/boot/grub/menu.lst, please re-run 'update-grub'.
```

Natürlich sind Veränderungen an der automatisch erstellten Konfiguration möglich. In der Konfigurationsdatei gibt es einen Abschnitt, der mit

```
### BEGIN AUTOMAGIC KERNELS LIST
```

beginnt. Diese Einträge können verändert werden; es dürfen aber auf keinen Fall die Kommentarzeichen (#) am Anfang jeder Zeile verändert werden! Ein Beispiel: Aus

```
# kopt=root=/dev/hda1 ro
```

wird:

```
# kopt=root=/dev/hda1 ro apm=on ide2=0x180,0x386
```

Die hinzugefügten Parameter werden beim Starten des Kernels übergeben. Nach der Änderung wird nochmals update-grub aufgerufen, und die Änderungen werden in die eigentliche Konfiguration am Ende der Datei übernommen.

Das Kommando grub-reboot erlaubt es, einen bestimmten Eintrag aus der GRUB-Konfiguration auszuwählen und einmalig zu starten. Dies entspricht der der von LILO bekannten Option -R.

Wird dieses Programm auf einem entfernten Rechner ausgeführt, so ist zu beachten, dass Einträge, die in der GRUB-Konfiguration mit einem Passwort versehen sind, einen Administrator vor Ort verlangen, der dieses Passwort eingibt. Das Passwort kann nicht als Option auf der Kommandozeile von grub-reboot angegeben werden.

Wenn Sie im Verzeichnis /boot/grub/ eine Datei menu.lst erzeugen, so wird GRUB beim nächsten Start aus dieser Datei ein Menü erstellen, und Sie können dann aus den verschiedenen Einträgen dieses Menüs den gewünschten auswählen. Ein Beispiel für ein solches Menü finden Sie unter /usr/share/doc/grub/examples/menu.lst.

```
debian:~# cp /usr/share/doc/grub/examples/menu.lst /boot/grub/
```

In dem Beispiel finden Sie einige Einträge für die verschiedensten Betriebssysteme sowie einige andere Goodies, die Sie mit GRUB anstellen können. Eine funktionsfähige Minimalkonfiguration könnte wie folgt aussehen:

```
timeout 10 default 0  # For booting Linux title  Debian GNU/Linux root (hd0,0) kernel /vmlinuz root=/dev/hda1
```

Wenn ein Kernel-Paket verwendet wird, so ist darauf zu achten, dass diese normalerweise eine Init-Ramdisk (initrd) verwenden. Ein entsprechender Eintrag, um eben diese Ramdisk ergänzt, sieht wie folgt aus:

```
title   GNU/Linux 2.6 SMP root    (hd0,6) kernel  /vmlinuz-2.6-686-smp
root=/dev/hda1 read-only initrd  /initrd-2.6-686-smp
```

Dieses Beispiel verwendet einen SMP-Kernel und eine etwas andere Festplattenaufteilung, aber das Prinzip sollte erkennbar sein.

Die Benutzung von Grafiken innerhalb des Bootloaders GRUB ist eine nicht-offizielle Erweiterung, die noch nicht in den Quellcode von GRUB eingeflossen ist. Trotzdem ist ein optisch ansprechender Hintergrund eine sehr attraktive Funktion, die bereits in viele Linux-Distributionen eingeflossen ist. Zuerst tauchte diese Erweiterung im Jahr 2001 auf einer GRUB-Mailingliste auf. Sie wird endgültig ab der Version 1.0 von GRUB auch im offiziellen Quellcode von GRUB auftauchen.

Unabhängig davon ist im Debian Paket von GRUB dieser Patch bereits integriert. Es sind aber einige Änderungen an der Konfigurationsdatei notwendig.

Ältere Versionen des Debian Pakets von GRUB waren noch nicht mit diesem Patch ausgestattet, so dass bei einem Upgrade von GRUB auch unbedingt das Kommando grub-install nochmals auszuführen ist.

Weiterhin benötigt man eine Grafikdatei im komprimierten XPM-Format mit maximal 14 Farben und einer Auflösung von 640x480 Punkten. Diese kann mit folgenden Kommandozeilen aus einer JPEG-Datei erzeugt werden:

```
$ convert -colors 14 -geometry 640x480 grubsplash.jpg grubsplash.xpm
$ gzip grubsplash.xpm
```

Abschließend ist noch eine Anpassung an der Konfigurationsdatei von GRUB vorzunehmen. Um die Grafikdatei zu laden, muss in der GRUB-Konfigurationsdatei (/boot/grub/menu.lst) angegeben werden, wo diese Datei zu finden ist. Da GRUB das Dateisystem direkt lesen kann, muss diese Datei nicht zwingend auf der Partition liegen,

auf der auch die GRUB-Konfigurationsdatei zu finden ist. Ein entsprechender Eintrag könnte wie folgt aussehen:

```
splashimage=(hd0,5)/boot/grub/grubsplash.xpm.gz
```

Nach einem Neustart müssen eventuell noch die Farben angepasst werden, so dass der Text lesbar ist.

Vielleicht haben Sie sich schon über die etwas ungewohnten Bezeichnungen der Laufwerke in der Konfigurationsdatei gewundert. GRUB benutzt die vom HURD-Kernel verwendeten Bezeichnungen für Gerätedateien; die Umstellung ist aber für den geübten Linux-Admin nicht sehr groß.

Zunächst ist zu bemerken, dass alle Bezeichnungen für Geräte in Klammern () geschrieben werden. Statt der unter Linux üblichen Buchstaben (z.B. hda) werden die einzelnen Festplatten mit fortlaufenden Zahlen ab 0 bezeichnet. hda entspricht also hd0. Die einzelnen Partitionen einer Festplatte werden wie unter Linux mit Zahlen bezeichnet, allerdings beginnend mit 0 und getrennt mit einem Komma. Somit entspricht /dev/hda1 also (hd0,0).

Weitere Geräte werden als (fd0) (Diskette) und (sd0) (SCSI-Festplatte) bezeichnet. Sie können GRUB auch zum Booten über das Netz, über PXE, einsetzen oder das Root-Filesystem per NFS mounten. Die Bezeichnung des Netzwerk-Device lautet (nd).

GRUB verfügt auch über eine sehr leistungsfähige Kommandozeile. Bedenken Sie dabei, dass zu dieser Zeit noch kein kompletter Kernel oder gar ein komplettes Betriebssystem geladen ist! Sollte es Ihnen nicht gelungen sein, ein funktionsfähiges Menü zu erstellen, so landen Sie automatisch auf der GRUB-Kommandozeile. Sie können hier alle Kommandos, Laufwerksbezeichnungen und Dateinamen mit der **TAB**-Taste automatisch vervollständigen lassen; dieses Feature kennen Sie sicher schon aus der Shell.

Wenn jedoch ein Menü angezeigt wird, können Sie dieses mit der Taste **c** verlassen und auf der Kommandozeile zum Beispiel von Hand einen anderen Kernel starten. Fehlerhafte Einträge im Menü lassen sich aus dem Menü mit der Taste **e** temporär verändern. Vergessen Sie nicht, diese Änderungen später in die Datei /boot/grub/menu.lst einzutragen.

Da zu diesem Zeitpunkt (GRUB wurde von der Festplatte gestartet, es ist kein Betriebssystem geladen) noch keine Tastaturbelegung geladen wurde, ist zu beachten, dass eine US-Tastenbelegung gilt. Dies kann durch Anpassungen in der GRUB-Konfigurationsdatei (/boot/grub/menu.lst)) verändert werden. Das GRUB-Kommando setkey alt neu erlaubt es, Tasten umzubelegen. Um die wichtigsten Tasten einer in Deutschland gebräuchlichen Tastatur richtig zu belegen, sind die folgenden Einträge in der GRUB-Konfigurationsdatei notwendig:

```
setkey y z setkey z y setkey Y Z setkey Z Y
setkey underscore question setkey question underscore setkey
ampersand percent setkey percent caret setkey equal parenright
setkey semicolon less setkey parenright parenleft setkey less
numbersign setkey parenleft asterisk setkey numbersign
backslash setkey doublequote at setkey colon greater setkey
plus bracketright setkey greater bar setkey minus
slash setkey asterisk braceright setkey slash ampersand
```

Wird GRUB von einem laufenden Betriebssystem aus aufgerufen, so gilt die aktuell geladene Tastaturbelegung.

Testen eines neuen Kernels mit GRUB

GRUB unterstützt eine Option, die es erlaubt, einen Kernel nur ein einziges Mal zu starten, um dann beim folgenden Start wieder einen Default-Kernel zu starten. Dies ist insbesondere nützlich, wenn das System in einem Rechenzentrum untergebracht ist oder am Standort des Systems keine Administratoren Zugriff auf das System haben, die in der Lage sind, einen Fehler beim Starten des Kernels zu beheben. Auch der Bootloader LILO kennt hierzu die Option -R.

Wird zur Konfiguration von GRUB das Skript update-grub eingesetzt, so sind keine Anpassungen an der Konfigurationsdatei notwendig. Sie müssen lediglich auf der Kommandozeile die GRUB-Shell starten (durch den Aufruf von grub). Dort ist die Kommandozeile

```
savedefault --default=N --once
```

einzugeben. N steht dabei für die Nummer des Eintrags in der Konfigurationsdatei; die Zählung beginnt bei 0. Beim nächsten Neustart wird nun der gewünschte (neue) Kernel gestartet, beim übernächsten Start der in der GRUB-Konfiguration eingestellte Kernel. Das folgende Skript erspart Ihnen das Eintippen des Kommandos in der GRUB-Shell, Sie müssen lediglich den gewünschten Eintrag angeben.

```
#!/bin/sh  if [ `id -u` != 0 ] ; then     echo "you must be root!" else          if [ ! $1 ] ; then
                echo "Usage: "$0" kernelnumber"     else               echo "setting kernel "$1"
once"                    grub --batch <<EOT             savedefault --default=$1 --once
        quit EOT        fi fi
```

Weitere Informationen zu GRUB finden Sie auf Ihrem System unter file:///usr/share/doc/grub-doc/ (hierzu muss auch das Paket grub-doc installiert sein) und auf den Projekt-Webseiten zu GRUB.

Kapitel 6. Systemadministration

5.13 Erstellen von eigenen Debian GNU/Linux-CD-ROMs 6.2 Init-Skripte

6.2 Init-Skripte

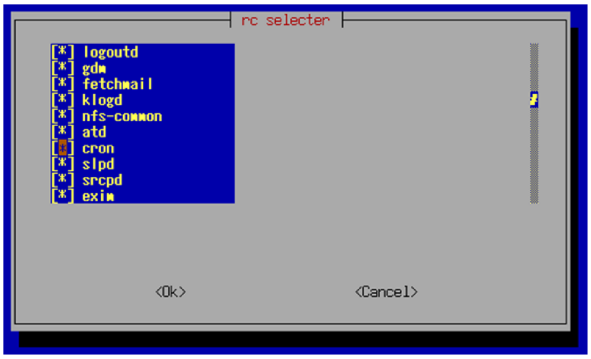
6.2.1 rcconf

6.2.2 update-rc.d

6.2.3 file-rc

Nach dem Laden des Linux-Kernels wird als erstes Programm überhaupt das Programm init gestartet. Dieses Programm startet alle weiteren Programme; welche das sind, wird über die Skripte im Verzeichnis /etc/init.d/ aus dem Paket sysv-rc bestimmt. Je nach gewünschtem Runlevel des Systems zeigen Links aus den Verzeichnissen /etc/rcN.d/ (N ist die Nummer des aktuellen Runlevels, normalerweise 2) auf die Skripte im Verzeichnis /etc/init.d/, so ist beispielsweise /etc/rc2.d/S20exim4 ein Link auf das Startskript für Exim (/etc/init.d/exim4). Den aktuellen Runlevel des Systems zeigt der Befehl runlevel an. Das Anlegen und Löschen dieser Links wird bei der Installation der jeweiligen Dienste automatisch vorgenommen. Ein installierter Dienst wird auf einem Debian System immer auch gestartet. Natürlich können Links von Hand hinzugefügt oder entfernt werden, aber auch für diese Arbeiten am System stellt Debian einige Werkzeuge zur Verfügung.

rcconf ist ein einfaches Werkzeug, mit dem einzelne Dienste aus den Init-Skripten aktiviert oder deaktiviert werden können. Es ist mit rcconf nicht möglich, den Runlevel eines Dienstes zu bestimmen.



Leider werden bei einer Aktualisierung eines Paketes alle Links auf die Skripte wieder hergestellt. Dies bedeutet auch, dass mittels rcconf deaktivierte Dienste nach einem Update der Software wieder aktiviert werden.

Detailliertere Einstellmöglichkeiten bietet update-rc.d. Zunächst sollte update-rc.d in jedem Fall mit der Option -n aufgerufen werden; diese zeigt lediglich an, welche Aktionen durchgeführt werden sollen, lässt die entsprechenden Links aber unangetastet.

```
usage: update-rc.d [-n] [-f] <basename> remove       update-rc.d [-n]
<basename> defaults [NN | sNN kNN]       update-rc.d [-n] <basename>
start|stop NN runlvl [runlvl] [...] .                -n: not really
-f: force
```

Grundsätzlich ist natürlich immer der Name des Pakets anzugeben, dessen Init-Skripte angepasst werden sollen. Bei Verwendung der Option remove werden alle Links auf das entsprechende Skript entfernt. Hierbei prüft update-rc.d, ob das Skript bereits entfernt wurde. Ist dies nicht der Fall, so bricht update-rc.d ab. Mit der Option -f kann erzwungen werden, dass die Links auch bei Vorhandensein eines Skriptes gelöscht werden.

Mit den Optionen start, stop bzw. defaults werden die notwendigen Links angelegt, gelöscht bzw. wiederhergestellt. Werden zusätzlich einer oder mehrere Runlevel angegeben, so können die Links entsprechend den Wünschen des Systemadministrators angepasst werden.

Leider werden bei einer Aktualisierung eines Paketes alle Links auf die Skripte wieder hergestellt. Dies bedeutet auch, dass mittels update-rc.d deaktivierte Dienste nach einem Update der Software wieder aktiviert werden.

Als Alternative zu den System-V-Init-Skripten steht das Paket file-rc zur Verfügung. Bei diesem Paket wird die Konfiguration des Systemstarts zentral in einer einzelnen Datei verwaltet. Natürlich kann nur ein Paket den Systemstart steuern, so dass bei der

Installation dieser für das System sehr essenziellen Funktionen eine deutliche Warnung ausgegeben wird.

```
wasabi:~# apt-get install file-rc
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut... Fertig                    Die
folgenden Pakete werden ENTFERNT:
sysv-rc                                                  Die folgenden
NEUEN Pakete werden installiert:                          file-rc
WARNUNG: Die folgenden essenziellen Pakete werden entfernt.
Die sollte NICHT geschehen, wenn Sie nicht genau wissen, was Sie
tun!            sysv-rc (wegen sysvinit)
0 aktualisiert, 1 neu installiert, 1 zu entfernen und 0 nicht aktualisiert.
Es müssen 36,2kB Archive geholt werden.
Nach dem Auspacken werden 4096B Plattenplatz freigegeben sein.
Sie sind im Begriff, etwas potenziell Schädliches zu tun.
Zum Fortfahren geben Sie bitte "Ja, tu was ich sage!" ein.
?] Ja, tu was ich sage!                                   Hole:1
http://ftp.de.debian.org sarge/main file-rc 0.8.5 [36,2kB]          Es
wurden 36,2kB in 0s geholt (49,9kB/s)                     dpkg:
sysv-rc: Abhängigkeitsproblem, aber lösche es auf Anfrage dennoch:
sysvinit hängt ab von sysv-rc (>= 2.85-2) | file-rc (>> 0.7.0); aber:
Paket sysv-rc soll gelöscht werden.                       Paket
file-rc ist nicht installiert.                    (Lese Datenbank ...
43720 Dateien und Verzeichnisse sind derzeit installiert.)   Entferne
sysv-rc ...                                       Wähle vormals
abgewähltes Paket file-rc.                        (Lese Datenbank ...
43703 Dateien und Verzeichnisse sind derzeit installiert.)   Entpacke
file-rc (aus .../archives/file-rc_0.8.5_all.deb) ...          Richte file-
rc ein (0.8.5) ...                                wasabi:~#
```

file-rc vereinheitlicht den Zugriff auf die Init-Skripte der einzelnen Dienste, indem alle (bisher von sysv-rc verwendeten) symbolischen Links entfernt werden und die komplette Verwaltung über die Datei /etc/runlevel.conf erfolgt.

This file was automatically generated by /usr/share/file-rc/rclink2file.sh. # You can use your favourite editor or update-rc.d(8) to modify it. # Read runlevel.conf(5) man page for more information about this file.
Format: # <sort> <off-> <on-levels> <command>
01 0,1,6 - /etc/init.d/gdm
02 - S /etc/init.d/mountvirtfs
05 - 1 /etc/init.d/single
05 - S /etc/init.d/bootlogd
05 - S /etc/init.d/initrd-tools.sh
05 - S /etc/init.d/keymap.sh
10 - 2,3,4,5 /etc/init.d/sysklogd
10 - S /etc/init.d/checkroot.sh
11 0,1,6 - /etc/init.d/cron
11 - 2,3,4,5 /etc/init.d/klogd
14 - 2,3,4,5 /etc/init.d/ppp
18 - 2,3,4,5 /etc/init.d/portmap
18 - S /etc/init.d/hwclockfirst.sh
19 0,6 - /etc/init.d/setserial
20 0,1,6 2,3,4,5 /etc/init.d/exim4
20 0,1,6 2,3,4,5 /etc/init.d/fam
20 0,1,6 2,3,4,5 /etc/init.d/inetd
20 0,1,6 2,3,4,5 /etc/init.d/lpd
20 0,1,6 2,3,4,5 /etc/init.d/makedev
20 0,1,6 2,3,4,5 /etc/init.d/ssh
20 - 0,6 /etc/init.d/sendsigs
20 - S /etc/init.d/modutils
21 - 2,3,4,5 /etc/init.d/nfs-common
25 0,6 - /etc/init.d/hwclock.sh
30 - 0,6 /etc/init.d/urandom
30 0,6 S /etc/init.d/etc-setserial
30 - S /etc/init.d/checkfs.sh
30 - S /etc/init.d/procps.sh
31 - 0,6 /etc/init.d/umountnfs.sh
32 - 0,6 /etc/init.d/portmap
35 - 0,6 /etc/init.d/networking
35 - S

```
/etc/init.d/mountall.sh                 36      -       S
/etc/init.d/discover                    36      -       S
/etc/init.d/mountvirtfs                 38      -       S
/etc/init.d/pppd-dns                    39      -       S
/etc/init.d/dns-clean                   39      -       S
/etc/init.d/ifupdown                    40      -       0,6
/etc/init.d/umountfs                    40      -       S
/etc/init.d/hostname.sh                 40      -       S
/etc/init.d/hotplug                     40      -       S
/etc/init.d/networking                  43      -       S
/etc/init.d/portmap                     45      -       S
/etc/init.d/mountnfs.sh                 46      -       S
/etc/init.d/setserial                   48      -       S
/etc/init.d/console-screen.sh           50      -       S
/etc/init.d/hwclock.sh                  55      -       S
/etc/init.d/bootmisc.sh                 55      -       S
/etc/init.d/urandom                     70      -       S
/etc/init.d/nviboot                     70      -       S
/etc/init.d/screen-cleanup              70      -       S
/etc/init.d/xfree86-common              79      0,1,6   -
/etc/init.d/nfs-common                  81      1       -
/etc/init.d/portmap                     86      0,1,6   -
/etc/init.d/ppp                         89      0,1,6   2,3,4,5
/etc/init.d/atd                         89      0,1,6   -
/etc/init.d/klogd                       89      0,6     -
/etc/init.d/hotplug                     89      -       2,3,4,5
/etc/init.d/cron                        90      0,1,6   -
/etc/init.d/sysklogd                    90      -       0
/etc/init.d/halt                        90      -       6
/etc/init.d/reboot                      99      -       2,3,4,5
/etc/init.d/gdm                         99      -       2,3,4,5
/etc/init.d/rmnologin                   99      -       2,3,4,5
/etc/init.d/stop-bootlogd               # THE LAST LINE IS NEVER
READ!
```

6.2 Init-Skripte

6.3 Alternativen (update-alternatives)

6.2 Init-Skripte 6.4 Systemzeit

„Alternativen“ sind eine weitere dieser vielen netten Kleinigkeiten, die das Leben in einem Debian System leichter und schöner machen. Beliebte Programme (beispielsweise vi) können durch „Alternativen“ (beispielsweise vim) ersetzt werden. Hierbei sind keine Einträge in irgendwelchen Konfigurationsdateien notwendig.

Die Debian Distribution liefert einige tausend Programme mit, für jeden Einsatzzweck kann der Anwender meist zwischen verschiedenen Programmen wählen. Dies geschieht im Normalfall auf der Kommandozeile durch Eintippen des entsprechenden Kommandos oder via Maus über ein Menü. Mit den „Alternativen“ ist der Administrator in der Lage, eine sinnvolle Voreinstellung für das Gesamtsystem zu geben.

Sehen wir uns das einmal am Beispiel vi an. Zunächst versuchen wir zu ermitteln, wo das Programm vi eigentlich zu finden ist:

```
wasabi:/home/fr# ls -l `which vi` lrwxrwxrwx 1 root root 20 24. Feb 13:47 /usr/bin/vi -> /etc/alternatives/vi
```

Und der Link

```
wasabi:/home/fr# ls -l /etc/alternatives/vi lrwxrwxrwx 1 root root 12 20. Mär 08:15 /etc/alternatives/vi -> /usr/bin/vim
```

zeigt wiederum auf das Programm vim. Nun lassen sich auch diese Links natürlich mit Unix-Bordmitteln verändern, doch halt! Auch hier gibt es wieder ein Werkzeug aus der Debian Werkzeugkiste.

update-alternatives ist das Werkzeug des Debian Systemadministrators, um die Konfiguration dieser Links vorzunehmen. Hierbei ist es möglich, aus sinnvollen Alternativen auszuwählen; es werden sogar Einschätzungen zum am besten geeigneten Programm für den jeweiligen Zweck abgegeben.

```
update-alternatives: need --display, --config, --install,   --remove or --
auto                Debian update-alternatives 1.9.20. Copyright (C) 1995
Ian Jackson. Copyright (C) 2000,2001,2002 Wichert Akkerman This is
free software; see the GNU General Public Licence version 2 or later
for copying conditions.  There is NO warranty.              Usage:
update-alternatives --install <link> <name>        <path> <priority>
[--slave <link> <name> <path>] ... update-alternatives --remove
<name> <path> update-alternatives --auto <name> update-alternatives -
-display <name> update-alternatives --config <name> <name> is the
name in /etc/alternatives. <path> is the name referred to. <link> is the
link pointing to /etc/alternatives/<name>. <priority> is an integer;
options with higher numbers are chosen.              Options:  --
verbose|--quiet  --test  --help  --version         --altdir <directory>  --
admindir <directory>
```

Zunächst sollte man sich einen Überblick über den aktuellen Zustand des Programms, hier vi, verschaffen:

```
wasabi:/home/fr# update-alternatives --display vi vi - status is manual.
link currently points to /usr/bin/vim /usr/bin/nvi - priority 30  slave
vi.1.gz: /usr/share/man/man1/nvi.1.gz /usr/bin/vim - priority 120  slave
vi.1.gz: /usr/share/man/man1/vim.1.gz Current `best' version is
/usr/bin/vim.
```

Hier ist sehr schön zu sehen, dass zwei Programme zur Verfügung stehen, die dem System als Editor vi dienen können: nvi und vim. Beide Versionen sind mit unterschiedlich hohen Prioritäten versehen; die höhere Version wird bevorzugt und in der letzten Zeile als Empfehlung ausgegeben.

Mit der Option --config können Veränderungen an der aktuellen Einstellung vorgenommen werden. Die Auswahl des neuen Wertes erfolgt mit der jeweils vorangestellten Zahl.

6.3 Alternativen (update-alternatives)

6.2 Init-Skripte 6.4 Systemzeit

6.4 Systemzeit

6.3 Alternativen (update-alternatives) 6.5 Verwalten von Konfigurationsdateien

6.4.1 date

6.4.2 rdate

(Fast) jeder Rechner verfügt heute über eine eingebaute Uhr, die über eine eingebaute Batterie die Uhrzeit auch im ausgeschalteten Zustand weiterlaufen lässt, so dass immer eine korrekte Systemzeit eingestellt ist. Das Betriebssystem kümmert sich während der Laufzeit um die richtige Zeit und liest sie einmalig beim Systemstart aus der Hardware-Uhr aus. Die Zeit des Betriebssystems lässt sich mit dem Kommando date anzeigen.

```
bash-2.03$ date Sat Jan 22 17:54:00 CET 2000
```

Die Uhrzeit, die vom Uhrenchip auf dem Motherboard geliefert wird, ist normalerweise ausreichend genau. Wenn diese Zeit nach dem Einschalten des Rechners um mehrere Monate oder Jahre abweicht, sollten Sie die Batterie auf dem Motherboard auswechseln (lassen).

Das Kommando date, so wie oben gezeigt, zeigt den Wochentag, den Monat, den Tag, die Uhrzeit, die Zeitzone sowie das Jahr an.

Ein weiteres Kommando, welches auf die Systemzeit zugreift, ist das Kommando time. Dieses zeigt allerdings nicht die Zeit an, wie man vielleicht vermuten könnte, sondern dient dazu, die Zeit festzulegen, die bestimmte Kommandos zur Ausführung benötigen.

```
time sleep 60                real    1m0.057s user    0m0.000s sys
0m0.000s
```

Wundern Sie sich nicht: Dieses Kommando benötigt natürlich eine Minute, bis es beendet wird. Das Kommando sleep „schläft“ in diesem Beispiel 60 Sekunden, und time ermittelt diese Zeit ;-). Wenn Sie sich einen neuen Rechner zugelegt haben und Ihre Freunde wissen, dass Sie diesen unter GNU/Linux betreiben, werden sie vielleicht fragen, wie lange es dauert, einen neuen Kernel zu übersetzen. Sie können diese Zeit mit dem Kommando time ermitteln. Natürlich kommt es auch darauf an, welche Treiber Sie in den Kernel einbinden wollen und welche Kernel-Version Sie übersetzen. Doch das ist ein anderes Thema.

Um die Systemzeit auf Ihrem Rechner zu korrigieren, können Sie dieses Kommando als Superuser benutzen. date hat eine Vielzahl von Optionen, um die Zeit zu stellen, Sie können die gewünschte Zeit in allen nur denkbaren Formaten angeben. Es gibt folgende Optionen:

bash-2.03$ date --help Usage: date [OPTION]... [+FORMAT] or: date [OPTION] [MMDDhhmm[[CC]YY][.ss]] Display the current time in the given FORMAT, or set the system date. -d, --date=STRING display time described by STRING, not 'now' -f, --file=DATEFILE like --date once for each line of DATEFILE -I, --iso-8601[=TIMESPEC] output an ISO-8601 compliant date/time string. TIMESPEC=`date' (or missing) for date only, 'hours', 'minutes', or 'seconds' for date and time to the indicated precision. -r, --reference=FILE display the last modification time of FILE -R, --rfc-822 output RFC-822 compliant date string -s, --set=STRING set time described by STRING -u, --utc, --universal print or set Coordinated Universal Time --help display this help and exit --version output version information and exit FORMAT controls the output. The only valid option for the second form specifies Coordinated Universal Time. Interpreted sequences are: %% a literal % %a locale's abbreviated weekday name (Sun..Sat) %A locale's full weekday name, variable length (Sunday..Saturday) %b locale's abbreviated month name (Jan..Dec) %B locale's full month name, variable length (January..December) %c locale's date and time (Sat Nov 04 12:02:33 EST 1989) %d day of month (01..31) %D date (mm/dd/yy) %e day of month, blank padded (1..31) %h same as %b %H hour (00..23) %I hour (01..12) %j day of year (001..366) %k hour (0..23) %l hour (1..12) %m month (01..12) %M minute (00..59) %n a newline %p locale's AM or PM %r time, 12-hour (hh:mm:ss [AP]M) %s seconds since 00:00:00, Jan 1, 1970 (a GNU extension) %S second (00..60) %t a horizontal tab %T time, 24-hour (hh:mm:ss) %U week number of year with Sunday as first day of week (00..53) %V week number of year with Monday as first day of week (01..52) %w day of week (0..6); 0 represents Sunday %W week number of year with Monday as first day of week (00..53) %x

```
locale's date representation (mm/dd/yy) %X   locale's time representation (%H:%M:%S) %y   last two digits of year (00..99) %Y year (1970...) %z   RFC-822 style numeric timezone (-0500) (a nonstandard extension) %Z   time zone (e.g., EDT), or nothing if no time zone is determinable  By default, date pads numeric fields with zeroes.  GNU date recognizes the following modifiers between `%' and a numeric directive.                `-' (hyphen) do not pad the field `_' (underscore) pad the field with spaces  Report bugs to <bug-sh-utils@gnu.org>.
```

In der Praxis werden Sie die Zeit aber nur selten korrigieren müssen, so dass an dieser Stelle ein Beispiel reichen sollte. Sehen Sie sich noch einmal die Hilfe zu date an. In der zweiten Zeile finden Sie Folgendes: or: date [OPTION] [MMDDhhmm[[CC]YY][.ss]]. Die gewünschte Zeit kann im Format MM (Month - Monat), DD (Day - Tag), hh (Hour - Stunde) und mm (Minute) angegeben werden. Die anderen Werte können Sie ignorieren, diese werden Sie nur selten gebrauchen. Um die Zeit drei Tage und eine halbe Stunde zurückzustellen, benutzen Sie folgendes Kommando:

```
linux:/home# date Sun Jan 23 10:47:59 CET 2000 linux:/home# date 01201017 Thu Jan 20 10:17:00 CET 2000
```

Wenn Sie über einen Internetanschluss verfügen, können Sie das Setzen der korrekten Uhrzeit auch Programmen wie zum Beispiel rdate überlassen. rdate übernimmt die Zeit von einem Server aus dem Netz, der über eine Funkuhr (DCF77-Signal) verfügt.

Sie können dieses Kommando in die Skripte einbinden, die Sie zum Aufbau der Internetverbindung benutzen. Wenn Ihr Rechner ständig oder immer zu bestimmten Zeiten mit dem Internet verbunden ist, können Sie rdate auch über das Programm cron aufrufen, indem Sie es in die crontab des Superusers eintragen. Rufen Sie dazu das Kommando crontab mit der Option -e (edit - verändern) auf:

```
bash-2.03$ su Password: linux:/home# crontab -e
```

Ergänzen Sie nun die Einträge in der Crontab um folgende Zeile:

```
10 01   * * *   /usr/sbin/rdate time.fu-berlin.de > /dev/null
```

Ein solcher Eintrag stellt jede Nacht um zehn Minuten nach eins die Systemzeit nach den Angaben des Servers time.fu-berlin.de neu. Benutzen Sie bitte einen Server in Ihrer Nähe. Viele Provider bieten einen solchen Service an - fragen Sie einfach nach.

Wenn Sie keinen geeigneten Server finden können, sollten Sie die Server der Physikalisch-Technischen Bundesanstalt (PTB) in Braunschweig benutzen. Wie Sie vielleicht wissen, wird von der PTB das amtliche Zeitsignal ausgesendet, das von DCF-77-Empfängern (Funkuhren) empfangen werden kann. Seit Anfang 2000 stellt die PTB diesen Dienst auch über das Internet zur Verfügung. Sie können die beiden öffentlichen Zeitserver der PTB als ptbtime1.ptb.de und ptbtime2.ptb.de erreichen.

Um die vorhin in die Vergangenheit beförderte Systemzeit zu korrigieren, können Sie rdate natürlich auch auf der Kommandozeile ausführen:

```
linux:/home# /usr/sbin/rdate time.fu-berlin.de Sun Jan 23 10:49:22 2000
```

6.4 Systemzeit

◁ 6.3 Alternativen (update-alternatives) △ 6.5 Verwalten von Konfigurationsdateien ▷

6.5 Verwalten von Konfigurationsdateien

6.4 Systemzeit 6.6 Installations- und Rettungsdiskette

6.5.1 dpsyco

Mit fast jedem auf einem Linux-System installierten Paket werden auch die notwendigen Konfigurationsdateien für die jeweilige Software-Komponente mitgeliefert. Das Debian Team leistet an dieser Stelle sehr gute Arbeit und erstellt zu allen Paketen angepasste Konfigurationsdateien, die nicht nur auf das Debian System angepasst sind, sondern auch in fast 100 Prozent aller Fälle sofort lauffähig sind.

Dies bedeutet für den Administrator eine große Arbeitserleichterung, da die gewünschte Software sofort ohne weitere Konfigurationsarbeiten lauffähig ist. Natürlich können diese allgemeinen Vorgaben nicht alle Bedürfnisse abdecken, und so sind viele Pakete bereits mit Konfigurationsprogrammen ausgestattet, die den Administrator durch die Konfiguration führen. Mit „debconf“ wurde sogar eine zentrale Datenbank bereitgestellt, die auch im Netz via LDAP verfügbar gemacht werden kann. In ihr werden alle Einstellungen gespeichert.

In einem größeren Netzwerk kann es aber wünschenswert sein, ein einheitliches System für die Verwaltung von Konfigurationsdateien, vielleicht sogar mit einer Versionskontrolle, zu etablieren. Hier bietet sich prinzipiell das Erstellen von eigenen Debian Paketen mit den gewünschten Konfigurationsdateien an. Das Debian Paketmanagement verhindert jedoch das Überschreiben von Dateien, so dass das Erstellen von Paketen mit angepassten Konfigurationsdateien ein Problem darstellt. Abhilfe schafft an dieser Stelle dpsyco.

dpsyco (Debian Packages of System Configurations) kann Debian „Konfigurationspakete“ erzeugen und verwalten. Ein „Konfigurationspaket“ kann „über“ ein normales Debian Paket installiert werden und so die in dem Debian Paket enthaltenen Konfigurationsdateien ersetzen oder auch ergänzen.

dpsyco-Pakete können Konfigurationsdateien aus einem Debian Paket überschreiben, patchen oder auch Benutzer und Gruppen hinzufügen. Es sind verschiedene Pakete verfügbar, die Sie mit dpsyco verwenden können:

```
sushi:/home/fr# apt-cache search dpsy
dpsyco - Debian packages of system configurations
dpsyco-base - Base package for the debian packages of system configurations
dpsyco-cfengine - Automate applying of cfengine configs
dpsyco-devel - Tools to create
```

```
configuration packages dpsyco-doc - Documentation for the debian packages of system configurations dpsyco-lib - Libraries for the debian packages of system configurations dpsyco-mysql - Automate administration of access to mysql dpsyco-patch - Automatically patch the debian file-system dpsyco-samba - Automate administration of access to samba dpsyco-skel - Automatically install a add-on skeleton dpsyco-ssh - Automate administration of access via ssh dpsyco-sudo - Automate administration of sudo privileges
```

Zunächst müssen Sie das Paket dpsyco selbst installieren. Dies hat zur Folge, dass auch die Pakete dpsyco-base und dpsyco-lib installiert werden:

```
sushi:/home/fr# apt-get install dpsyco Reading Package Lists... Done Building Dependency Tree... Done The following extra packages will be installed: dpsyco-base dpsyco-lib Suggested packages: dpsyco-doc dpsyco-mysql Recommended packages: dpsyco-skel dpsyco-sudo The following NEW packages will be installed: dpsyco dpsyco-base dpsyco-lib 0 upgraded, 3 newly installed, 0 to remove and 522 not upgraded. Need to get 54,2kB of archives. After unpacking 218kB of additional disk space will be used.  continue? [Y/n]  Get:1 ftp://ftp.de.debian.org sid/main dpsyco-lib 1.0.23 [19,5kB] Get:2 ftp://ftp.de.debian.org sid/main dpsyco-base 1.0.23 [24,3kB] Get:3 ftp://ftp.de.debian.org sid/main dpsyco 1.0.23 [10,4kB] Fetched 54,2kB in 1s (53,8kB/s)
```

Für die ersten Schritte können die Kommandos dpsyco-delhome und dpsyco-restorehome dienen. Hiermit kann ein Heimat-Verzeichnis eines Benutzers gelöscht und wieder hergestellt werden. Wichtig sind dabei die Einstellungen in der Datei /etc/dpsyco/adduser.conf.

6.5 Verwalten von Konfigurationsdateien

6.4 Systemzeit 6.6 Installations- und Rettungsdiskette

6.6 Installations- und Rettungsdiskette

6.5 Verwalten von Konfigurationsdateien 6.7 Technische Informationen zu den Bootdisketten

Sie können die Installationsdiskette oder die CD - falls Sie von dieser das System bei der Installation gestartet haben - auch zur Reparatur Ihres Systems benutzen. Dies kann in seltenen Fällen erforderlich sein, z. B. falls Sie den Rechner nicht ordnungsgemäß heruntergefahren haben, was ebenso bei einem Stromausfall der Fall ist. Manchmal kommt es auch vor, dass Sie etwas an den Dateien in /etc geändert haben oder dass ein neu erstellter Kernel nicht startet. Aber auch, wenn Sie das Passwort für den Superuser vergessen haben, kann die Bootdiskette Sie vor der neu installation bewahren.

In einem Notfall legen Sie die Rettungsdiskette ein und starten das System neu. Statt wie bei der Installation einfach die Eingabetaste zu drücken, können Sie nun am Boot-Prompt zusätzliche Parameter eingeben. Eine umfangreiche Übersicht der Befehle befindet sich auf der Diskette. Diese können Sie sich mit den Funktionstasten ansehen. Wir werden hier nur zwei gebräuchliche Fälle vorstellen.

Sollte zum Beispiel die LILO-Konfiguration fehlgeschlagen sein oder lässt sich der neu übersetzte Kernel nicht booten, so sind immer noch alle Daten auf der Root-Partition in einwandfreiem Zustand. Sie haben jetzt die Möglichkeit, den auf der Rettungsdiskette befindlichen Kernel zu booten und das Dateisystem auf der Root-Partition zu mounten. Geben Sie hierzu am Prompt Folgendes ein: rescue root=/dev/hda1

Hierbei ist zu beachten, dass Sie - je nachdem, auf welcher Partition sich Ihre Root-Partition befindet - das passende Device angeben. Nachdem der Kernel von der Diskette geladen worden ist, wird das System ganz normal gestartet, und Sie können sich danach am System anmelden und den Fehler beheben. Im Fall einer nicht funktionierenden Konfiguration von LILO loggen Sie sich als Superuser ein, beheben die Fehler in der Konfigurationsdatei und rufen anschließend das Kommando lilo auf.

Ein anderer Fall wäre ein Fehler im Dateisystem der Root-Partition, der ein normales Booten des Systems verhindert. In den meisten Fällen kommt man hierbei sogar ohne eine Rettungsdiskette aus. Meist hält das System beim Start an, und nach Eingabe des Root-Passworts können Sie das Dateisystem dann reparieren.

Dies lässt sich auch mit einer Rettungsdiskette ausführen. Booten Sie hierzu ganz normal mit der Diskette, als wollten Sie eine Installation durchführen. Wählen Sie noch die gewünschte Tastatur aus und wechseln Sie dann mit der Tastenkombination **ALT+F3** auf die dritte Konsole. Nach Drücken der Eingabetaste steht dort eine Shell zur Verfügung. Sie können von dort aus das defekte Dateisystem manuell reparieren. Benutzen Sie dazu das Kommando: e2fsck /dev/hda1. Auch hier ist wieder das entsprechende Device anzugeben. Nach einem Neustart des Systems sollte alles wieder wie gewohnt funktionieren.

6.5 Verwalten von Konfigurationsdateien 6.7 Technische Informationen zu den Bootdisketten

6.7 Technische Informationen zu den Bootdisketten

6.6 Installations- und Rettungsdiskette Kapitel 7. Systemsicherheit

Das Paket boot-floppies enthält den gesamten Quellcode, um die Installationsdisketten herstellen zu können.

Auf der Notfalldiskette ist ein MS-DOS-Dateisystem enthalten, auf das von jedem System zugegriffen werden kann, das DOS-Disketten mounten kann. Der Linux-Kernel liegt in der Datei linux. Die Datei root.bin ist ein mit dem Programm gzip komprimiertes Image eines 1.44 MByte großen Minix-Dateisystems. Dieses wird in die RAM-Disk im Hauptspeicher geladen und dort als Festplattenersatz für das Root-Dateisystem verwendet.

Wenn der Kernel auf der Notfalldiskette ersetzt werden soll, muss der neue Linux-Kernel mit den folgenden Eigenschaften (als fester Bestandteil, nicht als ladbare Module) konfiguriert werden:

„Initial RAM disk“-Unterstützung

MS-DOS-, Minix- und EXT2-Dateisystem

Ausführbare Dateien im ELF-Format

Danach wird der neu erzeugte Kernel (zImage oder bzImage) mit dem Dateinamen „linux“ auf die Notfalldiskette kopiert. Anschließend rufen Sie das Shell-Skript rdev.sh auf, das ebenfalls auf der Diskette zu finden ist. Dieses Programm nimmt noch ein paar kleine Einstellungen an dem neuen Linux-Kernel vor.

Die Disketten des Basis-Systems enthalten einen 512 Byte großen Kopf sowie jeweils einen Teil eines mit gzip komprimierten tar-Archivs. Werden jeweils die Köpfe entfernt und danach die Inhalte der Disketten zusammengefügt, so ergibt sich das vollständige komprimierte tar-Archiv.

Dieses Archiv enthält das Grundsystem, das während der Installation auf Ihr System kopiert wird. Es bietet die Basis-Funktionalität eines Debian GNU/Linux-Systems, dem jedoch noch zahlreiche Programme fehlen. Wenn das Archiv installiert ist, muss das System zunächst mit dem Menüpunkt Configure the Base System konfiguriert werden. Mit Hilfe von weiteren Menüeinträgen im Installationssystem werden die Netzwerkanbindung, der Betriebssystemkern und seine Module installiert und konfiguriert. Erst danach kann das System eingesetzt werden.

6.7 Technische Informationen zu den Bootdisketten

6.6 Installations- und Rettungsdiskette | Kapitel 7. Systemsicherheit

Kapitel 7. Systemsicherheit

6.7 Technische Informationen zu den Bootdisketten | 7.2 Securing Debian HOWTO

Inhaltsverzeichnis

In diesem Abschnitt sollen einige Hinweise gegeben werden, wie ein Debian GNU/Linux-System besser gegen unbefugte Zugriffe auf das System abgesichert werden kann. Debian bietet hierzu einige vorbereitete Pakete, die mit wenig Aufwand zu installieren oder zu konfigurieren sind.

Zunächst sollten Sie bereits bei der Installation des Systems darauf achten, dass Sie sowohl die Frage nach der Verwendung von Shadow-Passwörtern als auch die Frage nach MD5-verschlüsselten Passwörtern mit „Ja“ beantworten.

Generell ist zu sagen, dass jeder auf einem Unix-System laufende Dienst ein Sicherheitsrisiko darstellt. Installieren Sie also ausschließlich Dienste, die auch auf dem System eingesetzt werden sollen.

Das Debian Team nimmt die Sicherheit des Systems sehr ernst. Sicherheitslücken werden umgehend beseitigt, und aktualisierte Versionen der Software-Pakete sind auf den Debian FTP-Servern verfügbar. Um Zugriff auf die jeweils aktuellen Sicherheitsupdates zu haben, sollten Sie folgende Zeile der Datei /etc/apt/sources.list hinzufügen:

```
deb http://security.debian.org/ stable/updates main contrib non-free
```

Um sicherzustellen, dass sich keine Pakete mit Sicherheitslücken auf dem System befinden, kann das Paket harden installiert werden. In diesem Paket sind verschiedene Task-Pakete aufgeführt, die wiederum mit entsprechenden Anhängigkeiten belegt sind. Die „harden-*“-Pakete werden laufend aktualisiert, und sicherheitskritische Software wird durch entsprechende Abhängigkeiten, die in diesen Paketen festgelegt sind, von der Verwendung ausgeschlossen. Es kann also durchaus passieren, dass die Installation dieses Pakets dazu führt, dass ein benötigter Dienst gelöscht wird!

Das Debian Team ist dabei der Meinung, dass es besser ist, einen Dienst eine Zeit lang nicht zur Verfügung zu stellen, statt einem Angreifer einen Angriffspunkt zu liefern.

Das Paket harden besitzt definierte Abhängigkeiten zu diversen anderen Paketen, die auch gesondert installiert werden können. Diese decken kleinere Bereiche ab, so dass es möglich ist, einzelne Dienste oder bestimmte Systeme gezielt zu schützen. Diese Pakete sind:

harden - Makes your system hardened. harden-3rdflaws - Avoid packages with security problems. harden-clients - Avoid clients that are known to be insecure. harden-development - Development tools for creating more secure programs. harden-doc - Useful documentation to secure a Debian system. harden-environment - Hardened system environment. harden-localflaws - Avoid packages with security holes. harden-nids - Harden a system by using a network intrusion detection system. harden-remoteaudit - Audit your system from this host. harden-remoteflaws - Avoid packages with security holes. harden-servers - Avoid servers that are known to be insecure. harden-surveillance - Check services and/or servers automatically. harden-tools - Tools to enhance or analyze the security of the local system.

Die durch diese Pakete vorgenommenen Veränderungen basieren auf den im Securing Debian HOWTO beschriebenen Aktionen.

7.2 Securing Debian HOWTO

Im „Securing Debian HOWTO“ (http://www.debian.org/doc/manuals/securing-debian-howto/index.de.html - das englische Original finden Sie unter http://www.debian.org/doc/manuals/securing-debian-howto/index.en.html) - finden sich Informationen darüber, an welchen Stellen ein Debian System noch sicherer gemacht werden kann. Daher soll an dieser Stelle nur auf einige grundlegende Dinge hingewiesen werden.

Zunächst müssen Sie sich darüber im Klaren sein, dass ein System nur dann hundertprozentig sicher ist, wenn keinerlei Dienste auf diesem nach außen hin über das Netzwerk angeboten werden oder sogar gar keine Verbindung zu einem Netzwerk besteht. Dies ist heutzutage in der Realität natürlich unsinnig und nicht durchsetzbar; das System würde so nicht sinnvoll benutzbar sein.

Sie sollten sich auch darüber im Klaren sein, dass die Anforderungen an die Sicherheit stark vom Einsatz des Systems abhängig sind. Ein Privatnutzer wird deutlich andere Anforderungen stellen als der Administrator eines Firewall-Systems. Viele Einstellungen, die die Sicherheit eines Systems erhöhen, vereinfachen den Gebrauch nicht unbedingt; auch hier ist zwischen Benutzbarkeit und Paranoia abzuwägen.

Die softwareseitigen Einstellungen an einem Debian System, die zur Verbesserung der Sicherheit dienen, können nicht den physikalischen Zugriff auf das System verhindern. Wenn von Systemsicherheit die Rede ist, muss auch bedacht werden, dass zu einem völlig sicheren System auch gehört, dass der Zugriff auf die Hardware unterbunden wird. Ein Angreifer, der Zugriff auf die Hardware hat, kann beispielsweise über eine Boot-CD oder -Diskette Zugriff auf das System erlangen. Auch ein BIOS- oder LILO-Passwort kann den Diebstahl der Festplatte nicht verhindern. Ein völlig sicheres System gehört also hinter gut verschlossene Türen. Doch Betrachtungen zur Hardwaresicherung sollen an dieser Stelle nicht weiter verfolgt werden.

Allgemein formuliert, sollten folgende Punkte beachtet werden:

Entscheiden Sie, welche Dienste auf dem System benötigt werden, und beschränken Sie den Einsatz genau auf diese Dienste - nicht mehr und nicht weniger. Nicht benötigte Dienste sollten auf dem System nicht installiert oder zumindest deaktiviert sein. Weiterhin sollten die benutzten Ports durch eine Firewall auf dem System freigegeben bzw. die Ports von nicht genutzten Diensten über die Firewall gesperrt werden.

Verwendete Dienste sind abzusichern, so dass bei einem erfolgreichen Angriff auf diesen Dienst nicht das gesamte System kompromittiert wird (beispielsweise durch die Verwendung einer chroot-Umgebung).

Auf dem System sollten nur die notwendigen Benutzerkonten angelegt und aktiviert sein. Aktive Benutzerkonten sind durch entsprechende Beschränkungen (Unix-Zugriffsrechte, Quota, ACLs) abzusichern.

Setzen Sie Programme ein, die einen unbefugten Zugriff auf das System erkennen und melden, so dass geeignete Gegenmaßnahmen ergriffen werden können.

Bereits vor der Installation können einige Maßnahmen zur Sicherheit des Systems getroffen werden. Das Debian Installationsprogramm enthält ebenfalls einige Punkte, an denen die Sicherheit des Systems verbessert werden kann.

Bevor ein Betriebssystem auf einem neuen Computer installiert wird, sollte ein BIOS-Passwort gesetzt werden, und die Booteinstellungen sollten so gewählt werden, dass ein Systemstart von Diskette nicht möglich ist. Nach der Installation sollte darauf geachtet werden, dass so schnell wie möglich auch der Start von CD-ROM abgeschaltet wird.

Ein weiterer Vorteil dieser Einstellungen zeigt sich, wenn das System als Server in einem Rechenzentrum betrieben wird. Es wäre nicht das erste Mal, dass eine vergessene Diskette im Laufwerk einen erfolgreichen Reboot eines Systems verhindert; und das wird sehr ärgerlich, wenn ein direkter Zugriff auf das System nur mit einer längeren Anfahrt möglich ist.

Die Einteilung des verfügbaren Festplattenplatzes hängt von der Verwendung des Systems ab. Hierzu sollten Sie einige Dinge beachten:

Jede Partition, auf die die Benutzer des Systems Schreibzugriff haben, sollte auf einer eigenen Partition liegen, beispielsweise in den Bereichen /home und /tmp. Dies verhindert, dass ein Benutzer das Root-Dateisystem (/) unbenutzbar macht und das gesamte System in einen instabilen Zustand bringt. Es bleibt natürlich ein gewisser Platz (meist 5%, dieser Wert kann mit tunefs angepasst werden) für den Administrator reserviert, doch kann so anderen Benutzern das Arbeiten mit dem System unmöglich gemacht werden.

Es sollte für jeden Bereich, der automatisch mit Daten gefüllt wird (beispielsweise /var und hier insbesondere das Verzeichnis /var/log), eine eigene Partition vorgesehen werden. Auf Debian Systemen sollte /var großzügiger bemessen werden, da unter /var/cache/apt/archives Pakete temporär abgelegt werden, wenn die Installation über das Netz erfolgt. Weiterhin finden sich unter /var/lib/dpkg viele Dateien, die für das Paketmanagement benötigt werden.

Wenn Software installiert werden soll, die nicht in der Debian Distribution enthalten ist, sollten auch diese Bereiche auf eigenen Partitionen liegen; diese werden dann bei einer Neuinstallation des Systems nicht überschrieben. Nach dem „File Hierarchy Standard“ (FHS) sind dies /opt oder /usr/local.

Während der Installation wird nach einem Passwort für den Administrator (root) gefragt. Zusätzlich kann ein Konto für einen „normalen“ Benutzer dem System hinzugefügt werden; für diesen ist dann ebenfalls ein Passwort einzugeben. Auch wenn es möglich ist, hier ein sehr einfaches Passwort zu verwenden, so ist dies natürlich nicht empfehlenswert. Die Auswahl eines guten Passworts ist auf vielen Webseiten im Netz beschrieben, es sind dabei nur einige einfache Regeln zu beachten. Eine Suche nach „auswahl gutes passwort“ mittels Google führt schnell zum Erfolg.

Grundsätzlich sollte auf jedem System neben dem Zugangskonto für den Administrator auch mindestens ein solches Konto für einen Benutzer angelegt werden, das nicht über alle Rechte verfügt. Dieses Benutzerkonto sollte für grundsätzlich alle Logins bzw. Arbeiten auf dem System verwendet werden. Nur wenn für eine bestimmte Aufgabe die Zugriffsrechte nicht ausreichend sind, sollten Sie mit dem Kommando su die Identität wechseln. Nach Beendigung der Arbeiten als Administrator sollten Sie sich umgehend wieder als „normaler“ Benutzer im System bewegen.

Während der Installation erfolgt eine Abfrage, ob „Shadow Passwords“ aktiviert werden sollen. Wenn die Frage positiv beantwortet wird, werden die Passwörter in der Datei /etc/shadow verschlüsselt gespeichert. Diese Datei kann nur vom Administrator und

der Gruppe „shadow“ gelesen werden; somit kann kein Benutzer des Systems die verschlüsselten Passwörter lesen und versuchen, diese mit Hilfe einer Software zu entschlüsseln. Diese Einstellung kann später mit dem Programm shadowconfig rückgängig gemacht werden.

Weiterhin besteht die Möglichkeit, die Passwörter mit einer MD5-Verschlüsselung zu speichern; dies ist generell eine gute Idee, da so ein Angriff weiterhin erschwert wird und längere Passwörter möglich sind.

Wie bereits beschrieben, sollten nur die absolut notwendigen Dienste auf einem System aktiviert werden. Jeder neue Dienst schafft möglicherweise ein neues Sicherheitsloch, das vielleicht erst später zu einem Problem wird. Werden bestimmte Dienste nur selten benötigt, so können diese über die update-Kommandos (beispielsweise update-inetd) gezielt aktiviert und deaktiviert werden.

Lesen Sie die Debian Security-Mailinglisten. Informationen zu den verfügbaren Listen finden Sie unter http://lists.debian.org/debian-security-announce/. Dort ist auch beschrieben, wie Sie sich an einer solchen Liste an- und abmelden. Relevant sind in diesem Zusammenhang debian-security-announce, dort werden Sicherheitslücken bekannt gegeben, und Sie werden über Bugfixes dagegen informiert. Eine weitere Mailingliste zum Thema ist debian-security@lists.debian.org, dort werden alle Sicherheitsthemen rund um Debian behandelt.

Wenn Sie Meldungen über Sicherheitsupdates per E-Mail bekommen wollen, so senden Sie eine Mail an: debian-security-announce-request@lists.debian.org mit dem Wort „subscribe“ im Betreff der Mail. Die Anmeldung an dieser Mailingliste ist auch über die Webseite http://www.debian.org/MailingLists/subscribe möglich.

Auf dieser Mailingliste erhalten Sie nur sehr wenige Mails, Sie werden dort aber sehr schnell über Probleme mit Paketen informiert und erfahren eine Internet-Adresse, unter der eine fehlerbereinigte Version des Pakets zur Verfügung steht.

Nachdem das System mit allen benötigten Programmen eingerichtet ist, kann mit einigen weiteren Aktionen die Sicherheit des Systems weiter erhöht werden.

Jede Person, die Zugang zur Tastatur des Systems hat, kann eine Administrator-Shell bekommen und beispielsweise alle Passwörter ändern, indem am Bootprompt dateiname-des-bootkernels init=/bin/sh eingegeben wird. Um dies zu verhindern, kann ein Passwort für den Bootloader gesetzt werden. Dies kann global für alle Boot-Images geschehen oder individuell für jedes einzelne.

Wenn Lilo als Bootloader verwendet wird, muss die Datei /etc/lilo.conf um die Einträge password und restricted erweitert werden:

```
image=/boot/vmlinuz label=Linux read-only password=hackme restricted
```

Danach muss lilo noch einmal aufgerufen werden. Sorgen Sie dafür, dass die Datei /etc/lilo.conf nur vom Administrator gelesen werden kann, da das Passwort unverschlüsselt in der Konfigurationsdatei steht; dies erreichen Sie mit dem Kommando chmod 600 /etc/lilo.conf. Die Option restricted bewirkt, dass nur nach einem Passwort gefragt wird, wenn der Benutzer versucht, zusätzliche Parameter am Bootprompt anzugeben. Die Auswahl verschiedener, bereits in der Konfiguration eingetragener Kernel ist weiterhin möglich. Wird der Eintrag restricted weggelassen, fragt Lilo in jedem Fall nach einem Passwort.

Die genannten Einträge können am Anfang der Konfigurationsdatei allgemein gültig für alle Kernel in der Konfiguration angegeben werden oder aber innerhalb eines Abschnitts der Konfigurationsdatei nur für bestimmte Kernel.

Wird auf dem System GRUB verwendet, so müssen folgende Zeilen der Datei /boot/grub/menu.lst hinzugefügt werden:

```
timeout 3 password hackme
```

Die Option timeout sorgt nach der angegebenen Zeit dafür, dass der Standardeintrag gebootet wird.

Der von Debian Versionen vor 2.2 installierte MBR (Master Boot Record) wurde mit einer Option installiert, die es erlaubte, von Diskette zu booten, auch wenn dies sonst abgeschaltet war. Ob ein solcher MBR auf einem alten System heute noch installiert ist, lässt sich wie folgt prüfen:

Drücken Sie während des Startvorgangs die **SHIFT**-Taste; der MBR-Prompt sollte erscheinen.

Drücken Sie nun **f**, und das System startet von Diskette. Mit dieser kann ein Administratorzugang zum System erreicht werden.

Dieses Verhalten kann wie folgt verändert werden:

```
lilo -b /dev/hda
```

(wobei hda dem entsprechenden Devicenamen Ihres Systems angepasst werden muss).

Die Bootdisketten ab Debian Version 2.2 installieren lilo direkt in den MBR; hier tritt diese Lücke nicht auf.

Beim Mounten (Einhängen in das Dateisystem) von ext2-Partitionen gibt es diverse Optionen, die dem Kommando mount übergeben werden oder die direkt in die Datei /etc/fstab eingetragen werden können. Ein solcher Eintrag könnte beispielsweise wie folgt aussehen:

```
/dev/hda7    /tmp    ext2    defaults,nosuid,noexec,nodev    0    2
```

Die Optionen finden sich in der vierten Spalte. Die Option nosuid ignoriert gesetzte SUID- und GUID-Bits auf dieser Partition. Eine gesetzte Option noexec verhindert, dass auf dieser Partition befindliche Programme ausgeführt werden können, und nodev ignoriert Device-Dateien. Dabei ist zu beachten:

Dies bezieht sich nur auf ext2-Dateisysteme.

Auch solche Optionen können relativ leicht umgangen werden.

Hierzu ein Beispiel:

```
fr@sushi:/tmp# mount | grep tmp /dev/hda3 on /tmp type ext2
(rw,noexec,nosuid,nodev) fr@sushi:/tmp# ./date bash: ./date: Keine
```

```
Berechtigung fr@sushi:/tmp# /lib/ld-linux.so.2 ./date Sun Jul 29
14:40:32 CEST 2001
```

Ab der Kernelversion 2.6 ist die oben beschriebene Angriffsmöglichkeit nicht mehr gegeben.

Viele Tools, die von Hackern benutzt werden, versuchen im Verzeichnis /tmp Dateien anzulegen und diese auszuführen. Mit der Option noexec kann man dem Angreifer zumindest das Leben etwas schwerer machen.

Sobald eine neue Sicherheitslücke in einem Debian Paket oder einer Software dieses Pakets bekannt wird, wird von den Paket-Maintainern innerhalb weniger Stunden oder Tage ein Update der betroffenen Pakete bereitgestellt. Das aktualisierte Paket wird unter http://security.debian.org zur Verfügung gestellt.

Um auch die Sicherheitsupdates bei jeder Aktualisierung des Systems automatisch durchzuführen, muss folgende Zeile in die Datei /etc/apt/sources.list eingefügt werden:

```
deb http://security.debian.org/debian-security stable/updates \  main
contrib non-free
```

In Ländern, die den Import von kryptographischer Software nicht verbieten, kann zusätzlich folgende Zeile hinzugefügt werden:

```
deb http://security.debian.org/debian-non-US stable/non-US \  main
contrib non-free
```

Natürlich können auch die entsprechenden „ deb-src “ Zeilen hinzugefügt werden, wenn zusätzlich der Zugriff auf die Sourcen gewährleistet sein soll. Danach ist nur noch

ein apt-get update, gefolgt von einem apt-get upgrade, nötig, um das System auf den neuesten Stand zu bringen.

PAM erlauben es dem Systemadministrator auszuwählen, auf welche Weise die verschiedenen Programme eine Authentifizierung durchführen sollen. Hierzu muss jedes Programm an die Verwendung von PAM angepasst sein; dies ist seit Debian 2.2 für die meisten Programme der Fall. Ältere Versionen von Debian benutzen noch keine Authentifizierung über PAM. Für jedes Programm existiert im Verzeichnis /etc/pam.d eine eigene Konfigurationsdatei.

Mit PAM bietet sich die Möglichkeit, mehrere Authentifizierungsschritte vom Benutzer unbemerkt durchzuführen. Beispielsweise kann eine Authentifizierung sowohl gegen eine Datenbank als auch gegen die Datei /etc/passwd erfolgen.

Über PAM können viele Restriktionen auferlegt werden; genauso gut ist es aber möglich, das System weit zu öffnen und so Sicherheitslücken zu schaffen. Wenn Sie Einstellungen von PAM verändern, sollten Sie also größte Vorsicht walten lassen. Eine typische Konfigurationszeile enthält in der zweiten Spalte ein Kontrollfeld. Dieses sollte auf den Wert „required“ gesetzt werden, so dass bei einem Fehler in einem Modul ein Login verhindert wird.

Zuerst sollte die Unterstützung für MD5-verschlüsselte Passwörter aktiviert werden, um zu verhindern, dass ein Passwort leicht von einem Programm über eine Datenbank ermittelt werden kann. Auf Systemen vor der Debian Release „Sarge“ müssen die folgenden Zeilen in allen Dateien in /etc/pam.d/ hinzugefügt werden, die Zugang zu dem System erlauben. Ab der Debian Version „Sarge“ existiert die Datei /etc/pam.d/common-password, in der bereits die notwendigen Zeilen vorhanden sind, bei diesen Zeilen muss lediglich das Kommentarzeichen entfernt werden.

Auf älteren Systemen sind beispielsweise die Dateien login und ssh anzupassen.

```
password   required    pam_cracklib.so retry=3 minlen=12 difok=3
password   required    pam_unix.so use_authtok nullok md5
```

Der erste Eintrag lädt das Cracklib-PAM-Modul, das strengere Anforderungen an das verwendete Passwort stellt. Passwörter müssen mit diesem Modul mindestens 12 Zeichen haben; bei einer Passwortänderung müssen mindestens drei Zeichen verändert werden. Weiterhin werden nur drei Login-Versuche erlaubt.

Die zweite Zeile benutzt die Standard-Authentifizierung des Unix-Systems mit einer MD5-Verschlüsselung und erlaubt auch leere Passwörter. Die use_authtok Option wird benötigt, um das Passwort vom vorhergehenden Modul zu übernehmen.

Um den Zugriff so zu beschränken, dass der Benutzer root sich lediglich an einem lokalen Terminal anmelden kann, muss die folgende Zeile in der Datei /etc/pam.d/login aktiviert werden:

```
auth     requisite  pam_securetty.so
```

Weiterhin müssen die Terminals, von denen dem Benutzer root der Zugriff auf das System gewährt werden soll, in die Datei /etc/security/access.conf eingetragen werden. Um auch die eigentlichen Benutzer des Systems zu beschränken, beispielsweise in der Anzahl der gleichzeitigen Logins, muss noch die folgende Zeile aktiviert werden:

```
session  required   pam_limits.so
```

In der Datei /etc/pam.d/passwd ist nun noch die erste Zeile zu verändern. Dort muss die Option „md5“ eingetragen werden, um mit MD5 verschlüsselte Passwörter zu benutzen. Weiterhin kann die minimale Passwortlänge beispielsweise von 4 auf 6 Zeichen erhöht werden. Ebenso kann eine maximale Länge gesetzt werden, falls dies gewünscht ist. Schlussendlich sollte die Zeile in etwa wie folgt aussehen:

```
password   required   pam_unix.so nullok obscure min=6 max=11 md5
```

Wenn das Kommando su so geschützt werden soll, dass es nur von bestimmten Benutzern ausgeführt werden kann, muss zunächst eine neue Gruppe dem System hinzugefügt werden. Üblich ist es, hierzu die Gruppe „wheel“ zu verwenden, da diese üblicherweise noch nicht existiert und es somit unwahrscheinlich ist, dass bereits Dateien zu dieser Gruppe gehören. Dieser Gruppe fügen Sie das Benutzerkonto root sowie alle

Benutzerkonten hinzu, die das Kommando su ausführen können sollen. In der Datei /etc/pam/su ist dann folgender Eintrag zu ergänzen:

```
auth       requisite   pam_wheel.so group=wheel debug
```

Somit wird sichergestellt, dass nur Benutzer, die der Gruppe „wheel“ angehören, das Kommando su ausführen können. Alle anderen Benutzer bekommen eine entsprechende Meldung, wenn sie versuchen, dieses Kommando auszuführen.

Wenn nur bestimmten Benutzern eine Authentifizierung über PAM erlaubt werden soll, so ist dies relativ einfach über Dateien zu erreichen, in denen die Benutzer aufgeführt sind, denen der Login erlaubt oder verboten werden soll. Wenn beispielsweise nur dem Benutzer „fr“ der Login über ssh erlaubt werden soll, so muss dieser in die Datei /etc/sshusers-allowed eingetragen werden, und folgender Eintrag muss der Datei /etc/pam.d/ssh hinzugefügt werden:

```
auth       required    pam_listfile.so item=user sense=allow \
file=/etc/sshusers-allowed onerr=fail
```

Zu guter Letzt sind folgende Einträge der Datei /etc/pam.d/other hinzuzufügen:

```
 auth     required       pam_securetty.so auth     required
pam_unix_auth.so auth     required       pam_warn.so auth     required
pam_deny.so account  required       pam_unix_acct.so account  required
pam_warn.so account  required       pam_deny.so password required
pam_unix_passwd.so password required       pam_warn.so password
required       pam_deny.so session  required       pam_unix_session.so
session  required       pam_warn.so session  required       pam_deny.so
```

Diese Voreinstellungen sind erst einmal eine sinnvolle Vorgabe, denn es werden grundsätzlich zunächst alle PAM-Zugriffe verweigert.

Sie sollten auch einen aufmerksamen Blick in die Datei /etc/security/limits.conf werfen. Hier werden Ressourcen für die Benutzer des Systems festgelegt.

Generell sollten alle nicht benötigten Dienste auf einem System deaktiviert werden. Jeder laufende, nicht unbedingt benötigte Dienst stellt ein potenzielles Risiko dar. Dies betrifft beispielsweise die Dienste echo, charges, discard, daytime, time, talk, ntalk sowie die extrem unsicheren „r"-Kommandos wie rsh, rlogin und rcp. Für letztere ist es in jedem Fall besser, die Kommandos ssh und scp zu benutzen.

Nachdem die nicht benötigten Dienste deaktiviert sind, sollte geprüft werden, ob der Dienst inetd überhaupt noch benötigt wird. Dienste können natürlich auch als Dämon gestartet werden, statt inetd zu benutzen. „Denial of Service" (DoS)-Angriffe können auch auf den inetd als Ziel ausgeführt werden und so beispielsweise die Systemlast eines Rechners in die Höhe treiben. Wenn Sie trotzdem nicht auf den Einsatz eines solchen Dienstes verzichten können, so sollten Sie unter Umständen eine Alternative zu inetd einsetzen, die vielfältiger konfiguriert werden kann, beispielsweise den xinetd oder den rlinetd.

Veränderungen an der Datei /etc/inetd.conf können von Hand vorgenommen werden. Debian bietet aber eine einfach zu benutzende Alternative dazu: Mit dem Programm update-inetd können einzelne Dienste verändert werden.

Beispiel:

```
/usr/sbin/update-inetd --disable telnet /etc/init.d/inetd restart
```

Nach jeder Änderung ist der inetd noch neu zu starten.

Das Kommando update-inetd kennt noch viele andere Optionen, beispielsweise auch, um Einträge zu löschen:

```
sushi:/home/fr# update-inetd  Usage: update-inetd [OPTION] MODE
ARGUMENT  Options:   --version                output version
```

information and exit --help display this help and exit --verbose explain what is being done --debug enables debugging mode --multi allow multiple removes/disables --file FILENAME use FILENAME instead of /etc/inetd.conf --group GROUPNAME add entry to section GROUPNAME --comment-chars CHARACTERS use CHARACTERS as comment characters --pattern PATTERN use PATTERN to select a service Modes: --add ENTRY add ENTRY to /etc/inetd.conf --remove ENTRY remove ENTRY (regular expression) --enable SERVICE enable SERVICE in /etc/inetd.conf --disable SERVICE disable SERVICE in /etc/inetd.conf In order to prevent the shell from changing your ENTRY definition you have to quote the ENTRY using single or double quotes. You can use tabs (the tab character or \t) and spaces to separate the fields of the ENTRY. If you want to enable/disable more than one SERVICE you can use a comma separated list of services (no whitespace characters allowed).

Hier sollten einige Einstellungen zum Benutzer-Login und zur grundsätzlichen Konfiguration vorgenommen werden.

FAIL_DELAY 10

Diese Variable sollte auf einen höheren Wert gesetzt werden, um „Brute-Force“-Angriffe auf einem Terminal zu erschweren. Wenn ein falsches Passwort eingegeben wird, muss der Benutzer 10 Sekunden warten, bis ein neuer Login-Versuch gestartet werden kann. Dies frisst einiges an Zeit, wenn versucht wird, ein Passwort zu erraten. Diese Einstellung gilt nur, wenn getty benutzt wird; bei mingetty beispielsweise ist diese Einstellung ohne Wirkung.

FAILLOG_ENAB yes

Mit dieser Variablen werden fehlgeschlagene Logins im Logfile verzeichnet. Dies ist wichtig, wenn „Brute-Force“-Angriffe aufgezeichnet werden sollen.

```
LOG_UNKFAIL_ENAB    yes
```

Wenn die Variable FAILLOG_ENAB auf yes gesetzt wird, so sollte auch diese Variable auf yes gesetzt werden. Diese Einstellung schreibt auch unbekannte Benutzernamen bei einem Login-Versuch ins Logfile. Es ist darauf zu achten, dass die entsprechende Logdatei nicht von allen Benutzern gelesen werden kann, da Benutzer häufig anstelle des Benutzernamens das Passwort eingeben. Damit andere Benutzer die Logdatei nicht lesen können, sind die Zugriffsrechte beispielsweise auf 640 zu setzen.

```
SYSLOG_SU_ENAB      yes
```

Diese Einstellung verzeichnet die Benutzung des Kommandos su im Syslog.

```
SYSLOG_SG_ENAB      yes
```

Diese Einstellung erfüllt die gleiche Funktion wie die vorhergehende, jedoch für das Kommando sg (Ausführen eines Kommandos mit der ID einer anderen Benutzergruppe).

```
MD5_CRYPT_ENAB      yes
```

Wie schon beschrieben, reduzieren MD5-verschlüsselte Passwörter die Gefahr des Erschleichens eines Passwortes durch entsprechende Software. Wenn auf dem System noch „slink“ (Debian 2.1) eingesetzt wird, sollten Sie vor dem Aktivieren dieser Option einen Blick in die Dokumentation werfen. Ansonsten wird diese Einstellung über PAM realisiert.

```
PASS_MAX_LEN        50
```

Wenn MD5-Passwörter in der PAM-Konfiguration aktiviert sind, so sollte diese Variable auf den gleichen Wert wie dort gesetzt werden.

Diese Datei enthält eine Liste der Benutzer, die sich nicht per FTP-Protokoll einloggen dürfen. Benutzen Sie diese Datei nur, wenn Sie wirklich sicher sind, dass auf dem System auch tatsächlich ein FTP-Server laufen soll, da per FTP der Benutzername und das Passwort immer im Klartext übertragen werden. Wenn der benutzte FTP-Dämon PAM unterstützt, so können auch dort die Benutzer zugelassen oder ausgeschlossen werden.

TCP-Wrapper wurden entwickelt, als noch keine echten Paketfilter verfügbar waren, aber trotzdem eine Kontrolle notwendig wurde. Ein TCP-Wrapper erlaubt oder verbietet einem Rechner oder einer Domäne das Nutzen eines Dienstes. Nähere Informationen finden Sie in der Manpage hosts_access(5).

Im Folgenden sehen Sie vielleicht das kleinste System, um Einbruchsversuche zu registrieren. Auf alle Fälle sollte auf jedem System eine Firewall installiert sein, zusätzlich zu diesem TCP-Wrapper. Dieser kleine Eintrag in die Datei /etc/hosts.deny schickt bei jedem verweigerten Zugriff auf einen Service eine Mail an den Administrator.

```
ALL: ALL: spawn ( \   echo -e "\n\   TCP Wrappers\: Connection refused\n\   By\: $(uname -n)\n\   Process\: %d (pid %p)\n\   User\: %u\n\   Host\: %c\n\   Date\: $(date)\n\ " | /bin/mail -s "Connection to %d blocked" root)
```

Natürlich ist dieses Beispiel nicht perfekt: Bei vielen Verbindungen innerhalb einer kurzen Zeit werden natürlich auch entsprechend viele E-Mails gesendet, was wiederum einem DoS-Angriff gleichkommt.

Sollte es einmal notwendig sein, Arbeiten am System als Administrator durchzuführen, so kann das Kommando su benutzt werden, um die benötigten Rechte zu erlangen. Versuchen Sie, allen Benutzern klarzumachen, dass Arbeiten als Administrator nur in Ausnahmefällen gestattet sind. In jedem Fall ist ein Login als root zu vermeiden und stattdessen das Kommando su zu benutzen. Noch besser ist es jedoch, das Kommando su komplett zu entfernen und stattdessen das Kommando sudo zu benutzen, das eine ganze Reihe weiterer Funktionen bietet. Der Befehl su ist aber weithin bekannt, da er auch auf vielen anderen Unix-Systemen eingesetzt wird.

sudo erlaubt einem Benutzer, Kommandos unter der ID eines anderen Benutzers auszuführen, gegebenenfalls auch als Administrator. Wenn der Benutzer in der Datei /etc/sudoers aufgeführt ist und sich authentifiziert hat, können Kommandos ausgeführt werden, die ebenfalls in dieser Datei aufgeführt sind. Verletzungen dieser Regel, wie beispielsweise ein falsches Passwort oder die versuchte Ausführung eines unerlaubten Programms, werden aufgezeichnet und per E-Mail an den Administrator (root) geschickt.

Der Befehl chroot ist eine leistungsfähige Möglichkeit, um ein Programm, einen Dämon oder einen Benutzer zu beschränken. Man kann sich das wie in einem Gefängnis vorstellen, aus dem ein Ausbruch unmöglich ist (normalerweise... aber einige Leute schaffen es ja doch manchmal...). Wenn einem Benutzer nicht vollkommen vertraut wird, so kann für diesen eine chroot-Umgebung eingerichtet werden. Dies kann einiges an Festplattenplatz beanspruchen, wenn alle benötigten Binaries und Bibliotheken in diese Umgebung kopiert werden müssen. Aber wenn es dem Benutzer gelingt, Schaden anzurichten, so bleibt dieser auf die durch das Kommando chroot definierte Umgebung beschränkt.

Ein gutes Beispiel für eine solche Anwendung ist folgende: Die Authentifizierung erfolgt nicht gegen die Datei /etc/passwd, sondern gegen LDAP oder/und eine MySQL-Datenbank. Ein verwendeter FTP-Dämon benötigt das Binary und ein paar Bibliotheken. Hier bildet eine chroot-Umgebung eine exzellente Verbesserung der Sicherheit, falls eine Sicherheitslücke in diesem FTP-Dämon auftaucht. In diesem Fall ist lediglich die Benutzer-ID des FTP-Dämons betroffen und keine anderen Benutzer des Systems. Natürlich können auch viele andere Dienste von solch einer Umgebung profitieren.

Als Hinweis: Bisher wird bei keiner Debian Version eine chroot-Umgebung für die Dienste und Server verwendet.

Viele Funktionen des Kernels können während der Laufzeit verändert werden, beispielsweise indem mit dem Kommando echo ein Wert in die entsprechende Datei geschrieben wird, oder mit dem Kommando sysctl. Mit dem Kommando sysctl -A kann angezeigt werden, welche Einstellungen verändert werden können und welche Optionen verfügbar sind. In seltenen Fällen muss etwas verändert werden, aber auf diesem Weg kann die Sicherheit des Systems erhöht werden.

```
/proc/sys/net/ipv4/icmp_echo_ignore_broadcasts = 0
```

Wird diese Variable auf den Wert 1 gesetzt, so verhält sich das System nach außen wie ein Windows-System, wenn ein Broadcast-ping-Kommando das System erreicht.

```
/proc/sys/net/ipv4/icmp_echo_ignore_all = 0
```

Wenn ICMP-Pakete auf der vorgeschalteten Firewall nicht geblockt werden sollen, so ist diese Variable auf den Wert 0 zu setzen.

```
/proc/sys/net/ipv4/tcp_syncookies = 1
```

Diese Option ist ein zweischneidiges Schwert: Einerseits schützt sie gegen „Syn-Flooding“-Attacken, andererseits entspricht dies nicht den RFCs. Diese Option beschäftigt die angreifende Seite ebenso mit „Syn-Floods“, so dass diese gleichermaßen beschäftigt ist. Diese Option kann auch in /etc/network/options verändert werden, indem die Option syncookies auf yes gesetzt wird.

```
/proc/sys/net/ipv4/conf/all/log_martians = 1
```

Mit dieser Option werden Pakete mit unerlaubten Adressen, beispielsweise wegen eines fehlerhaften Routings, im Netzwerk protokolliert.

SVGAlib ist eine schöne Einrichtung für Liebhaber der Konsole. In der Vergangenheit sind jedoch immer wieder Sicherheitslücken bekannt geworden. So wurden Sicherheitslücken in `zgv` bekannt, mit denen Administratorrechte erlangt werden konnten. Wenn möglich, sollten Sie auf die Verwendung der SVGAlib verzichten.

Die Übertragung zwischen zwei Rechnern sollte auf keinen Fall mit Programmen wie `ftp` oder `rcp` erfolgen, da diese den Benutzernamen und das Passwort unverschlüsselt übertragen. Als sichere Alternative steht das Programm `scp` zur Verfügung, das im Paket `ssh` enthalten ist. Hier werden sowohl der Benutzername als auch das Passwort und auch die Daten selbst in verschlüsselter Form übertragen.

Eine saubere Definition von Benutzerquota ist wichtig, um zu verhindern, dass Benutzer ein komplettes Dateisystem mit Daten füllen können.

Es können grundsätzlich zwei verschiedene Quota-Systeme eingesetzt werden: benutzer- oder gruppenorientierte Quota. Dies ist bei der Planung zu berücksichtigen.

Einige Punkte sind bei der Benutzung von Quota zu beachten:

Quota sollten in der Summe so klein gewählt werden, dass nicht der gesamte Festplattenplatz belegt werden muss.

Quota sollten so groß gewählt werden, dass die Benutzer nicht bei der Arbeit behindert werden. Beispielsweise sollte das Spoolverzeichnis für Mail nicht zu knapp bemessen werden.

Quota müssen auf allen von Benutzern beschreibbaren Bereichen eingerichtet werden, beispielsweise `/home` und `/tmp`.

Für jede Partition bzw. jedes Verzeichnis, auf die bzw. das Benutzer Schreibzugriff haben, sollten Quota aktiviert werden. Für diese Bereiche ist ein sinnvoller Wert zu errechnen, der eine Balance zwischen Sicherheit und Benutzbarkeit des Systems schafft.

Doch nun zur Benutzung von Quota: Zunächst müssen Sie prüfen, ob die Quota-Unterstützung im Kernel aktiviert ist. Wenn dies nicht der Fall ist, muss ein neuer Kernel erzeugt werden. Danach ist sicherzustellen, dass das Paket quota installiert ist.

Quota werden aktiviert, indem in der Datei /etc/fstab der Eintrag für das entsprechende Dateisystem in der Spalte „Options“ um den Eintrag usrquota erweitert wird. Wenn statt Benutzer-Quota Gruppen-Quota benutzt werden sollen, lautet der Eintrag grpquota. Natürlich können auch beide gleichzeitig verwendet werden. Nun müssen im Root-Verzeichnis des Dateisystems leere Dateien quota.user und/oder quota.group erzeugt werden.

Das Quota-System muss nun neu gestartet werden. Dies geschieht durch die Kommandos /etc/init.d/quota stop und /etc/init.d/quota start. Nun können die gewünschten Grenzwerte gesetzt werden.

Um Quota für einen Benutzer (beispielsweise fr) zu setzen, wird das Kommando edquota -u fr benutzt. Gruppen-Quota werden mit dem Kommando edquota -g gruppe gesetzt. Nun können die Grenzwerte für „soft“ und „hard“ sowie für die Inodes gesetzt werden.

Einige Logdateien sind nach der Installation nicht perfekt. Zunächst ist es nicht notwendig, dass die Dateien /var/log/lastlog und /var/log/faillog von jedem Benutzer gelesen werden können. In der Datei lastlog sind Benutzer verzeichnet, die sich in der letzten Zeit am System angemeldet haben; in der Datei faillog finden sich fehlgeschlagene Loginversuche. Bei beiden Dateien sollten die Zugriffsrechte auf 660 verändert werden. Prüfen Sie genau, ob Logdateien mit unnötigen Zugriffsrechten versehen sind. Meist sind Lese- und Schreibrechte für den Administrator und für die Gruppe adm oder root ausreichend.

Debian wird mit einem täglichen Cronjob installiert, der in /etc/cron.daily/standard zu finden ist. Der Aufruf von /usr/sbin/checksecurity führt eine Überprüfung des Systems auf Änderungen des Flags setuid an allen Dateien auf dem System durch.

Um diese Überprüfung zu aktivieren, muss die Variable CHECKSECURITY_DISABLE in der Datei /etc/checksecurity.conf auf

FALSE gesetzt sein. Dies ist auch die Voreinstellung, so dass hier eigentlich keine Änderungen erforderlich sind.

Diese beiden Kommandos sind auf einem ext2-Dateisystem sehr sinnvoll. Die Attribute einer Datei können mit lsattr angezeigt und mit chattr verändert werden. Attribute unterscheiden sich von Zugriffsrechten! Es gibt viele verschiedene Attribute, hier werden nur die sicherheitsrelevanten aufgeführt. Zwei Attribute können nur vom Administrator gesetzt werden.

Zunächst wäre das „a"-Flag zu nennen. Wenn dieses Attribut gesetzt ist, können an die entsprechende Datei nur Daten angehängt (append) werden. Dieses Attribut kann auf einige Dateien in /var/log/ angewendet werden; beachten Sie jedoch, dass einige Dateien von Zeit zu Zeit „rotiert" werden (unter „rotieren" versteht man das tägliche Umkopieren und Umbenennen von Logdateien. Dabei werden Dateien vom Vortag komprimiert. Nach einer bestimmten Zeit, meist eine Woche, werden ältere Dateien gelöscht).

Das zweite wichtige Flag ist „i", das für „immutable" steht. Wenn dieses Flag gesetzt wird, kann eine Datei weder verändert noch gelöscht oder umbenannt werden. Auch kann kein Link auf diese Datei erzeugt werden. Wenn Benutzern die Einsicht in Logdateien verwehrt werden soll, so können Sie dieses Flag setzen und die Leserechte entfernen. Dies bietet auch eine etwas höhere Sicherheit gegen Eindringlinge, da diese sich sicher darüber wundern, dass die Datei nicht gelöscht werden kann. Trotzdem sollten Sie sich nicht darauf verlassen, dass ein Eindringling diese Funktion nicht kennt. In jedem Fall ist es ihm zu diesem Zeitpunkt bereits gelungen, in das System einzudringen...

Sind Sie sicher, dass die auf einem bereits seit Wochen oder Monaten ans Internet angeschlossenen System installierten Programme noch die ursprünglichen sind? Oder kann es sein, dass bereits das Programm /bin/login durch eine veränderte Variante ersetzt wurde, die einen unbemerkten Login als Administrator erlaubt...?

Die einzige Methode, ein System gegen solche Veränderungen zu schützen, ist, die installierten Dateien täglich, wöchentlich oder beispielsweise monatlich (je häufiger dieser Test durchgeführt wird, desto weniger Zeit bleibt einem Eindringling, auf dem System zu agieren) mittels einer Checksumme zu dokumentieren und diese mit vorab gespeicherten Werten zu vergleichen.

Pakete, die einen regelmäßigen Check der installierten Pakete ermöglichen, sind: sXid, AIDE (Advanced Intrusion Detection Environment) und Tripwire (die aktuelle Version befindet sich im Bereich non-free, eine der nächsten Versionen wird unter der GPL stehen).

Mit dem Paket debsums können alle MD5-Checksummen der installierten Pakete mit den Original-Checksummen der Dateien in den Ursprungspaketen der Distribution verglichen werden.

Mittels debsums -a können die Checksummen aller Pakete des Systems verglichen werden, wogegen Sie mit debsums <paketname> die Checksummen bestimmter Pakete vergleichen. Hierbei ist zu beachten, dass es einem versierten Eindringling durchaus möglich ist, auch diese Kontrolle der Checksummen zu beeinflussen. Die ursprünglichen Checksummen der Pakete liegen in den Dateien /var/lib/dpkg/info/<paketname>.md5sums und können vom Angreifer auch angepasst werden. Eine höhere Sicherheit bietet das Paket Tripwire.

Um die Sicherheit des Programms locate zu erhöhen, kann alternativ das Programm slocate verwendet werden. slocate ist eine verbesserte Version von locate aus dem GNU-Projekt. Wenn slocate verwendet wird, werden dem Benutzer nur Dateien angezeigt, auf die dieser Benutzer auch tatsächlich Zugriff hat. Weiterhin können per Konfiguration auch Dateien und Verzeichnisse gezielt ausgeschlossen werden, so dass diese nicht von locate in der Datenbank erfasst werden.

SSH sollte generell für alle Remote-Logins auf einem System eingesetzt werden. Wenn Sie noch telnetd einsetzen, so ändern Sie dies jetzt sofort. Heutzutage ist es sehr einfach, den Netzwerkverkehr mitzuschneiden und so an unverschlüsselte Benutzernamen und Passwörter zu gelangen. Die Verwendung von Programmen, die

keine verschlüsselte Kommunikation erlauben, verbietet sich somit von selbst. Auch der Einsatz von hochwertigen Netzwerkkompenenten wie Switches erlaubt Angreifern das „Mitlauschen“ auf den angeschlossenen Ports! Die meisten Switches schalten bei zu hoher Last einfach das „Switching“ ab und arbeiten wie ein normaler Hub! Verlassen Sie sich nie auf diese Komponenten, sondern sorgen Sie selbst für Sicherheit! Das Paket `ssh` kann mit dem Kommando `apt-get install ssh` schnell und einfach installiert werden.

Nun sollten alle Benutzer angewiesen werden, ausschließlich `ssh` und keinesfalls `telnet` zu benutzen. Deinstallieren Sie `telnet` nach Möglichkeit. Auch sollte ein Login als Administrator unmöglich gemacht werden. Um Administrator-Rechte zu erlangen, können die Kommandos `su` oder besser `sudo` benutzt werden.

Auch die Konfiguration des ssh-Dämons kann zur Erhöhung der Sicherheit noch verbessert werden. In der Datei `/etc/ssh/sshd_config` verbietet folgende Zeile einen Login via ssh als Administrator:

```
PermitRootLogin No
```

Dies verhindert, dass per Brute-Force-Angriff ein Login als Administrator möglich ist, da kein Login als Administrator erlaubt wird. Es sind nun zwei Login-Vorgänge notwendig (zunächst als Benutzer, dieser muss dann noch Administrator werden):

```
Listen 666
```

Wenn der Port, auf dem der ssh-Dämon läuft, verändert wird, so kann dies einen potenziellen Angreifer auch etwas beschäftigen. Es stehen aber verschiedene Netzwerktools zur Verfügung, mit deren Hilfe schnell und einfach ermittelt werden kann, auf welchem Port ein Dienst läuft. Verwenden Sie hier nicht zu viel Ehrgeiz...

```
PermitEmptyPasswords no
```

Leere Passwörter sollten ebenfalls verhindert werden.

```
AllowUsers alex bea fr
```

Mit dieser Option wird nur bestimmten Benutzern der Login via ssh erlaubt.

```
AllowGroups wheel admin
```

Gleichermaßen wird mit dieser Direktive nur bestimmten Gruppen der Zugriff per ssh erlaubt.

Um Benutzern oder Gruppen explizit den Zugriff zu verbieten, stehen die Optionen DenyUsers und DenyGroups zur Verfügung.

```
PasswordAuthentication no
```

Wird diese Option auf „no“ gesetzt, so ist der Login nur Benutzern gestattet, deren ssh-Key der Datei ~/.ssh/authorized_keys hinzugefügt wurde. Diese Einstellung ist sehr empfehlenswert! Bei Verwendung eines Keys wird das Passwort nicht mehr vom Client zum Server übermittelt. Die Autorisierung wird anhand eines Hash-Wertes geprüft, und der Client erhält lediglich die Freigabe für den Server - oder auch nicht.

Weiterhin ist der ssh-Dämon so einzustellen, dass auschließlich das Protokoll der Version 2 verwendet wird. Die Protokollversion 1 ist nach aktuellem Kenntnisstand als angreifbar anzusehen und sollte nicht mehr verwendet werden.

Alle diese Optionen beziehen sich auf die Konfigurationsdateien von OpenSSH. Momentan sind drei ssh-Pakete im Umlauf: ssh1, ssh2 und OpenSSH, das vom OpenBSD-Team entwickelt wurde. OpenSSH ist eine komplett freie Implementation eines ssh-Dämons, die sowohl ssh1 als auch ssh2 unterstützt. Wenn das Paket „ssh“ installiert wird, so wird das Paket OpenSSH gewählt.

Wenn auf dem System tatsächlich ein FTP-Server installiert werden muss, so ist sicherzustellen, dass die Benutzer sich in einer „chroot“-Umgebung bewegen. Diese hält den Benutzer in seinem Home-Verzeichnis gefangen; ansonsten könnte er sich frei im Dateisystem bewegen.

Mit der Option

```
DefaultRoot ~
```

im globalen Abschnitt der Datei /etc/proftpd.conf kann dies sichergestellt werden. Danach ist der Server mit /etc/init.d/proftpd restart von der Veränderung der Konfiguration zu unterrichten.

Um X-Anwendungen von einem Server aus auf einem Client darzustellen, ist zunächst auf dem Client das Öffnen der Anwendung durch den Server zu erlauben. Vielfach ist zu lesen, dass dies durch das Kommando „xhost +“ geschieht. Dies ist auch prinzipiell nicht falsch, erlaubt jedoch jedem System den Zugriff auf das X-Display. Besser ist es, den Zugriff nur von den gewünschten Systemen aus zu erlauben, indem der entsprechende Rechnername dem Kommando als Option mitgegeben wird, also beispielsweise xhost +sushi.

Eine deutlich sicherere Lösung ist es allerdings, die komplette Sitzung über ssh - und damit verschlüsselt - zu tunneln. Dies erfolgt automatisch, wenn eine ssh-Verbindung zu einem System aufgebaut wird. Soll diese Funktion abgeschaltet werden, so ist die Option X11Forwarding in der ssh-Konfiguration anzupassen. In Zeiten von ssh sollte auf die Verwendung von xhost komplett verzichtet werden.

Wenn keinerlei Zugriff auf den X-Server von anderen Systemen im Netz erlaubt werden soll, so ist es das Sicherste, dies bereits beim Start von X zu verhindern, indem der TCP-Port 6000 deaktiviert wird. Wenn X über das Kommando startx gestartet wird, so kann dies mit startx -- -nolisten tcp geschehen.

Wenn der Display-Manager (das Programm, das einen grafischen Login bereitstellt, beispielsweise XDM, KDM oder GDM) nur auf dem lokalen System benötigt wird, so ist sicherzustellen, dass XDMCP (X Display-Manager Control Protocol) deaktiviert ist. Wenn das Programm xdm benutzt wird, kann dies durch die Zeile

```
DisplayManager.requestPort:     0
```

in der Datei /etc/X11/xdm/xdm-config geschehen.

Die XDMCP-Unterstützung ist bei der Grundinstallation aller Display-Manager unter Debian deaktiviert.

Das Lesen bzw. Empfangen von E-Mail mittels POP3 ist das am häufigsten eingesetzte Protokoll ohne Verschlüsselung. Unabhängig davon, ob POP3 oder IMAP als Protokoll verwendet wird - beide benutzen Benutzernamen und Passwörter im Klartext, und auch die Daten werden unverschlüsselt übertragen. Als Alternative kann auch hier ssh verwendet werden, falls ein Shell-Account auf dem Mail-Server vorhanden ist.

Mittels fetchmail kann über ssh eine verschlüsselte Verbindung aufgebaut werden; hierzu eine entsprechende ~/.fetchmailrc:

```
poll my-imap-mailserver.org via "localhost"   with proto IMAP port
1236      user "ref" there with password "hackme" is alex here warnings
3600    folders      .Mail/debian    preconnect 'ssh -f -P -C -L 1236:my-
imap-mailserver.org:143 -l ref      my-imap-mailserver.org sleep 15 <
/dev/null > /dev/null'
```

Die wichtigste Zeile ist der „preconnect“-Eintrag. Dieser startet eine ssh-Session und installiert einen Tunnel, der automatisch die Verbindungen zum IMAP-Server auf Port 1236 weiterreicht, verschlüsselt. Dies nur als ein Beispiel, normalerweise sind einfachere Konfigurationen ausreichend. Alternativ kann fetchmail auch mit SSL benutzt werden.

Wenn verschlüsselte IMAP- und POP3-Server zur Verfügung gestellt werden sollen, so ist das Paket stunnel zu installieren. Die Dämons müssen dann über stunnel -p /etc/ssl/certs/stunnel.pem -d pop3s -l /usr/sbin/popd gestartet werden, wobei -l den gewünschten Dämon und -d den Port beschreibt. Die Option -p setzt das SSL-Zertifikat.

Mittlerweile sind auch POP- und IMAP-Server verfügbar, die über Verschlüsselungsfunktionen mittels SSL verfügen. Als Server-Programm für das POP-Protokoll wäre hier apop zu nennen.

Ein „Loghost“ ist ein zentraler Rechner, auf dem Daten aus dem Syslog-Dämon verschiedener Rechner gespeichert werden. Wenn ein Eindringling ein System geknackt hat, so ist es ihm unmöglich, die Spuren aus den Logdateien zu entfernen, außer er knackt auch noch den Loghost. Somit sollte speziell ein solcher Loghost gut abgesichert sein. Um ein System in einen Loghost umzuwandeln, muss lediglich der Syslog-Dämon mit der Option -r gestartet werden. Natürlich muss aber auch auf allen Rechnern, die nun die Daten auf diesem Loghost abliefern sollen, eine Anpassung erfolgen. Auf diesen Systemem sind in der Datei /etc/syslog.conf Einträge in der folgenden Form hinzuzufügen:

```
<facility>.<level>          @<loghost>
```

Das Feld <facility> kann dabei der einen Werte authpriv, cron, daemon, kern, lpr, mail, news, syslog, user, uucp oder local1 bis local7 annehmen. Als <level> kann alert, crit, err, warning, notice oder info angegeben werden. Hinter dem Zeichen „@“ ist der Hostname des Loghosts anzugeben.

Wenn generell alle Einträge auf dem entfernten System protokolliert werden sollen, so führt folgende Zeile zum Erfolg:

```
*.*                 @loghost
```

Im Idealfall wird man die Logdateien sowohl auf dem lokalen System als auch auf dem Loghost speichern, um durch Vergleichen der Dateien schneller zu einem Ergebnis zu kommen. Weitere Informationen finden Sie in den Manpages zu syslog(3), syslogd(8) und syslog.conf(5).

Auf einem unveränderten System läuft der Nameserver BIND nach der Installation mit den Rechten des Benutzers und der Gruppe „root“. BIND kann sehr leicht umkonfiguriert werden, so dass der Dienst unter einem anderen Benutzerkonto läuft. Leider kann er dann nicht mehr automatisch neue Netzwerkgeräte erkennen, die während des laufenden Betriebs hinzugefügt wurden, beispielsweise eine PCMCIA-Netzwerkkarte in einem Notebook oder auch virtuelle Netzwerk-Devices.

In der Datei README.Debian des Nameservers finden Sie weitere Informationen darüber, wie Sie BIND unter einem anderen Benutzerkonto zum Laufen bringen können. Wenn möglich, sollten Sie darauf verzichten, BIND mit Administratorrechten zu benutzen.

Um BIND mit einer anderen Benutzer-ID zu starten, muss zunächst ein neuer Benutzer und eine entsprechende Gruppe angelegt werden. Es kann beispielsweise als Benutzername und Gruppe der Name „named“ benutzt werden.

Hierzu sind folgende Kommandos notwendig:

```
addgroup named adduser --system --ingroup named named
```

Nun muss in der Datei /etc/init.d/bind der Eintrag

```
start-stop-daemon --start
```

in

```
start-stop-daemon --start --quiet --exec /usr/sbin/named -- -g named -u named
```

geändert werden.

Natürlich ist BIND nun noch mit /etc/init.d/bind stop und /etc/init.d/bind start neu zu starten. Dabei sollten im Syslog (/var/log/syslog) in etwa folgende Einträge auftauchen:

```
Jul  8 23:21:01 sushi named[12432]: group = named Jul  8 23:21:01 sushi named[12432]: user = named
```

Damit ist die Umstellung abgeschlossen. Idealerweise kann BIND nun noch in einer „chroot“-Umgebung betrieben werden.

Snort ist ein flexibler Packet-Sniffer, der verschiedenste Angriffe ermitteln kann. Hierzu gehören Buffer-Overflows, CGI-Angriffe, SMB-Angriffe und vieles mehr. Snort kann Alarmierungen in Echtzeit durchführen. Dieses Programm sollte auf jedem Router installiert sein, um jederzeit das Netzwerk zu überwachen. Die Installation erfolgt mit einem apt-get install snort; beantworten Sie die Fragen und werfen Sie dann einen Blick auf die Logdateien.

Unmittelbar nach jeder neuen Debian GNU-Installation, beispielsweise von CD, sollten die neuesten verfügbaren Security-Updates installiert werden. Durch die notwendigen Vorlaufzeiten bei der Produktion von CDs sind diese natürlich nicht immer auf dem neuesten Stand. Natürlich ist es nicht ausreichend, ein solches Update einmalig auszuführen, vielmehr müssen diese Updates in regelmäßigen Abständen durchgeführt werden. Dies verhindert, dass Software mit bekannten Sicherheitslücken über längere Zeit im laufenden Betrieb verwendet wird.

Zunächst sollten alle Netzwerkdienste, deren Passwörter im Klartext übertragen werden, deaktiviert bzw. gegen Versionen mit verschlüsselter Kommunikation ausgetauscht werden. Dies betrifft Dienste wie FTP, Telnet, NIS, RPC usw.

Auch sollten Sie auf die Verwendung von NIS (Network Information Service) verzichten. Bei einer fehlerhaften Konfiguration kann es leicht zu Sicherheitslücken kommen.

Auch sollten Sie RPC deaktivieren, sofern es möglich ist. Für diesen Dienst sind eine ganze Reihe von Sicherheitslücken bekannt. Natürlich wird gerade NFS häufig verwendet und stellt in vielen Netzen einen wichtigen Basisdienst dar. Hier gilt es, einen Kompromiss zwischen Sicherheit und Benutzbarkeit der Netzwerkdienste zu finden. Viele DDoS (Distributed Denial of Service)-Angriffe benutzen RPC-Sicherheitslücken, um Systeme in so genannte „Agents“ oder „Handler“ umzuwandeln.

Das Deaktivieren des Portmappers ist relativ einfach. Wie für jede Lösung gibt es auch hier verschiedene Wege. Auf einem Debian System ist der einfachste Weg sicherlich ein `update-rc.d portmap remove`. Dieses Kommando löscht jeden symbolischen Link auf den Portmapper in `/etc/rc${runlevel}.d/`. Dies kann natürlich auch auf herkömmlichem Wege von Hand erledigt werden. Eine weitere, nicht ganz elegante Möglichkeit ist es, die Zugriffsrechte so zu ändern, dass das Skript nicht mehr ausführbar ist. Dazu verwenden Sie `chmod 644 /etc/init.d/portmap`. Dies würde jedoch zu einer Fehlermeldung beim Start des Systems führen.

Natürlich ergibt es nur wenig Sinn, lediglich einen Teil der Dienste von unverschlüsselter auf verschlüsselte Kommunikation umzustellen; hier sollte der Systemadministrator konsequent durchgreifen. Generell sollten die Dienste `ftp`, `telnet`, `pop`, `imap` und `http` entfernt und durch die entsprechenden Dienste mit verschlüsselter Kommunikation (`ftp-ssl`, `telnet-ssl`, `pop-ssl`, `https`) ersetzt werden.

Im Netz sind einige Kernel-Patches verfügbar, die nicht Bestandteil des Standard-Linux-Kernels sind, dessen Sicherheit aber verbessern. Vor dem Einsatz ist im Einzelfall zu prüfen, ob der gewünschte Patch nicht schon in den benutzten Kernel eingeflossen ist. Auch erhebt die folgende Auflistung natürlich keinerlei Anspruch auf Vollständigkeit:

OpenWall Patch von „Solar Designer“. Eine Sammlung von Patches, die beispielsweise Links beschränken, FIFOs in /tmp/ unterbinden, das /proc-Dateisystem schützen, die Behandlung von Datei-Deskriptoren ändern und einiges andere verändern. Detaillierte Informationen finden Sie auf der Homepage http://www.openwall.com/linux/.

LIDS - Linux intrusion detection system von Huagang Xie und Philippe Biondi. Dieser Patch vereinfacht die Sicherung eines Linux-Systems. Jeder Prozess kann beschränkt werden, indem Lese- und Schreibberechtigungen auf Dateien vergeben werden können. Weiterhin können je Prozess „Capabilities“ gesetzt werden. Zum Einsatz von LIDS auf Debian GNU-Systemen gibt es eine spezielle Seite im Netz unter www.lids.org/. Dort findet sich neben einem vorkonfigurierten Kernel auch ein Paket mit dem Administrationstool.

POSIX Access Control Lists (ACLs) für Linux. Dieser Patch erweitert den Kernel um Access Control Lists (ACLs), eine Methode, die es gestattet, den Zugriff auf Dateien detaillierter zu beschränken, als es mit den üblichen Schemata möglich ist. http://acl.bestbits.at.

Eine wichtige Aufgabe des Systemadministrators ist die regelmäßige Überwachung aller Dateien im System. Ein Eindringling kann es nicht nur auf den Diebstahl von Daten abgesehen haben, auch das Verändern von Systemdateien (beispielsweise um neue Benutzer anzulegen) oder von Programmdateien (Austausch von Binaries durch veränderte Versionen) kann ein Ziel sein. Das Auffinden dieser Veränderungen ist nur möglich, wenn der Stand vor der Veränderung bekannt ist.

Debian unterstützt durch das ausgefeilte Paketsystem die Überprüfung der installierten Dateien. Als erster Einstieg kann das Programm cruft dienen.

cruft untersucht das komplette Dateisystem nach Dateien, die eigentlich nicht vorhanden sein sollten, beziehungsweise nach Dateien, die sich nicht mehr im Dateisystem finden lassen. Hierzu wird im Wesentlichen auf die Informationen aus den Dateien im Verzeichnis /var/lib/dpkg/info/ zugegriffen. cruft überwacht aber auch darüber hinausgehende Informationen wie beispielsweise die „alternatives“-Informationen, die lost+found-Verzeichnisse in einem ext2-Dateisystem und auch die Heimat-Verzeichnisse der Benutzer.

7.2 Securing Debian HOWTO

7.3 Weitere Möglichkeiten

7.2 Securing Debian HOWTO 7.4 PGP

Im Folgenden finden Sie einige Gedanken dazu, wie das bisher Gesagte weiterhin umgesetzt werden kann.

Das PAM-System ist durch den modularen Aufbau in der Lage, die verschiedensten Medien zur Authentifizierung zu nutzen. Wie wäre es mit einem Scanner für Fingerabdrücke oder einem Iris-Scanner?

Alle bisherigen Logdaten wurden auch in Dateien geschrieben. Diese können von einem Angreifer natürlich verändert oder gelöscht werden, auch wenn diese auf anderen Rechnern gespeichert werden. Logfiles, die auf einem Drucker mit Endlospapier ausgegeben werden, können nicht gelöscht werden!

Um das Löschen oder das Verändern von Dateien zu verhindern, kann ein komplettes System einmalig konfiguriert werden und dann auf eine bootfähige CD-ROM geschrieben werden. Natürlich sind so noch Angriffe auf das System möglich; es können aber keine Daten verändert oder zusätzliche Programme installiert werden. Für ein Firewall-System ist dies beispielsweise eine sinnvolle Möglichkeit, das System zu schützen.

Wenn möglich, sollten alle Kernel-Treiber nicht als Module übersetzt werden. Dann kann die Möglichkeit, Kernel-Module zu laden, komplett deaktiviert werden. So können viele Angriffe abgewehrt werden. Auch hier gilt: Nicht benutzte Funktionen sind abzuschalten.

Nach einem Einbruch gibt es nicht viel zu tun. Das System ist sofort vom Netz zu nehmen und komplett neu zu installieren. Einfach, nicht wahr? Natürlich gilt es herauszufinden, wie der Eindringling in das System eingedrungen ist. Dies geschieht in einer abgeschotteten Umgebung, also ohne Netzzugang für das betroffene System. Es sind zur späteren weiteren Analyse alle Daten auf einem geeigneten Medium zu sichern. Gegebenenfalls ist eine Meldung an ein CERT (Computer Emergency Response Team, in Deutschland beispielsweise das DFN-CERT http://www.cert.dfn.de/) zu erstellen und dort der Einbruch zu melden. Ist eine Strafverfolgung des Einbruchs vorgesehen oder geplant, so sollten Sie ggf. auf professionelle Unterstützung zurückgreifen.

Weiterhin sind auf dem neuen System alle notwendigen, vorab beschriebenen Sicherheitsvorkehrungen zu treffen.

Nach einem Einbruch auf einem System, beispielsweise durch ein über das Netzwerk gesnifftes Passwort, werden häufig so genannte „Rootkits“ installiert, die dem Angreifer einen Zugang mit Rechten des Administrators (root) erlauben. Es ist dabei über eine Sicherheitslücke, die dem Angreifer zunächst lediglich normale Benutzerrechte erlaubt, möglich, mittels bekannter Lücken weitere Zugriffsrechte zu erlangen.

Ein Rootkit wird dabei, wie der Name schon sagt, als „Bausatz“ in Form von Skripten geliefert. Rootkits sind in den verschiedensten Varianten im Internet verfügbar. Der Angreifer muss nicht zwingend über weit reichende Systemkenntnisse verfügen, ein erschlichener Benutzer-Account ist oft ausreichend, um ein Rootkit zu installieren. Angreifer, die solche Rootkits einsetzen, werden daher aufgrund des fehlenden Know-hows auch als „Script-Kiddies“ bezeichnet.

Rootkits haben dabei die (unangenehme) Eigenschaft, viele Arbeitsschritte bei einem Einbruch in ein System zu automatisieren. Es werden dabei zunächst Programme installiert und ausgetauscht, um Aktivitäten auf dem System zu verbergen. Dabei ist es beispielsweise üblich, zunächst /bin/login gegen eine modifizierte Variante auszutauschen. Die neue Version erlaubt Logins eines bestimmten Benutzers ohne oder mit einem bekannten Passwort. Natürlich erscheinen diese Logins in keinem Logfile. Weiterhin werden Programme, die Aufschlüsse über die Aktivität von Benutzern erlauben, durch veränderte Versionen ersetzt. Dies betrifft meist Programme wie ls (Anzeigen von Dateien und Zugriffsrechten) oder auch ps (Anzeigen von Prozessen eines Benutzers). Mit diesen neuen Programmen werden Aktivitäten des Angreifers verschleiert oder komplett verborgen.

Um ein solches Rootkit zu erkennen, können Sie das Paket chkrootkit (Check Rootkit) verwenden. Dieses kann ab der Debian Version 3.1 direkt via APT installiert werden.

Der Aufruf dieses Programms erfolgt wie üblich auf der Kommandozeile, es kann dabei die Option -q angegeben werden, um die Informationen über die durchgeführten Tests zu unterdrücken. Leider neigt chkrootkit dazu, mitunter Alarm zu schlagen, auch wenn kein Rootkit installiert ist. Bekannt ist dies bei Netzwerkinterfaces, die im Promiscous-Modus laufen (wenn beispielsweise tcpdump eingesetzt wird). Bekannt ist dieses Fehlverhalten auch bei Kerneln mit NPTL-Patch (ab Kernel 2.6 immer enthalten); dabei wird ein „LKM-Trojaner“ gemeldet. Weiterhin werden solche falsch-positiven Ergebnisse auch auf sehr langsamen Rechnern gemeldet.

Das Paket skdetect ist ein auf das „Suckit“-Rootkit spezialisiertes Programm. Einige Tests wurden hier genauer implementiert, die Erkennung ist aber nicht so umfassend.

7.3 Weitere Möglichkeiten

7.2 Securing Debian HOWTO 7.4 PGP

7.4 PGP

7.3 Weitere Möglichkeiten 7.5 GnuPG

PGP - Pretty Good Privacy (auf Deutsch etwa: ziemlich gute Privatsphäre) - war lange Jahre der Standard zur Verschlüsselung und wurde hauptsächlich zur Verschlüsselung von E-Mails verwendet (obwohl sich auch andere Daten damit verschlüsseln lassen). PGP benutzt ein Verfahren mit digitalen Schlüsseln und ermöglicht damit einen sicheren Austausch von Daten zwischen Personen, die sich noch nie getroffen haben. Durch die Verfügbarkeit der Quellcodes wurde PGP von Entwicklern in der ganzen Welt auf so ziemlich jedes Betriebssystem portiert.

PGP verwendet die Algorithmen IDEA und RSA, die leider lange Zeit durch Software-Patente geschützt waren. Diese Tatsache führte zur Entwicklung einer komplett freien Alternative, deren Entwicklung mittlerweile so weit fortgeschritten ist, dass wir auf eine weitere Beschreibung von PGP verzichten möchten und auf GnuPG verweisen. Mittlerweile ist dieses Patent aber abgelaufen, so dass auch GnuPG die RSA-Verschlüsselung nutzen kann.

7.4 PGP

◁ 7.3 Weitere Möglichkeiten △ 7.5 GnuPG ▷

7.5 GnuPG

7.4 PGP Kapitel 8. Weitere Informationen

GnuPG steht für „GNU Privacy Guard“ und ist ein kompletter, freier Ersatz für PGP. Der Algorithmus IDEA wird nicht benutzt, so dass GnuPG ohne Einschränkungen benutzt werden kann. GnuPG implementiert den OpenPGP-Standard, der aus den Datenformaten von PGP 5.x und 6.x entwickelt wurde. Die Homepage von GnuPG finden Sie unter folgender URL: http://www.gnupg.org/.

Folgende Eigenschaften zeichnen GnuPG aus:

Vollständiger PGP-Ersatz

Es werden keine patentierten Verfahren verwendet.

GPL - GNU General Public License

Kann als Filterprogramm verwendet werden.

Vollständige OpenPGP-Implementierung

Gegenüber PGP erweiterte Funktionalität und Sicherheitserweiterungen

Ver- und entschlüsselt PGP-5.x-Daten.

ElGamal (unterschreiben und verschlüsseln), DSA, 3DES, Blowfish, Twofish, CAST5, MD5, SHA-1

RIPE-MD-160 und TIGER werden unterstützt.

Einfache Implementierung neuer Verfahren durch Erweiterungsmodule

Die Benutzer-ID wird im Standardformat erzeugt.

Unterstützung für Schlüsselverfallsdatum

Benutzerführung in Englisch, Französisch, Deutsch, Italienisch, Polnisch, brasilianischem Portugiesisch, Russisch und Spanisch

Online-Hilfe

Wahlweise anonyme Adressaten

Eingebaute Unterstützung für Schlüsselserver (http://wwwkeys.pgp.net)

Um verschlüsselte E-Mails austauschen zu können, müssen Sie zuerst ein eigenes Schlüsselpaar erzeugen, das aus einem privaten und einem öffentlichen Schlüssel besteht. Mit einem geeigneten Mailprogramm (dieses muss die Verschlüsselung von Mails per PGP/GnuPG unterstützen, mutt oder spruce sind dazu in der Lage) können Sie dann verschlüsselte Mail austauschen. Lassen Sie sich von Ihrem Mailpartner hierzu seinen öffentlichen Schlüssel (public key) zuschicken, und schicken Sie ihm im Gegenzug Ihren.

Sie können nun Mails mit Ihrem privaten Schlüssel und dem öffentlichen Schlüssel Ihres Partners verschlüsseln. Im Gegenzug kann dieser die von Ihnen verschlüsselte Mail nur mit seinem privaten und Ihrem öffentlichen Schlüssel entschlüsseln.

Dieses Verfahren schützt jedoch noch nicht vor gefälschten Schlüsseln. Um sicherzugehen, dass ein öffentlicher Schlüssel auch wirklich von der entsprechenden Person stammt, können Sie Ihren Schlüssel von verschiedenen Leuten unterschreiben (signieren) lassen. Hierzu müssen Sie sich persönlich mit Leuten treffen, die GnuPG/PGP einsetzen und dieser Person Ihren öffentlichen Schlüssel übergeben (auf Diskette) sowie sich ausweisen. Wenn Sie niemanden kennen, der GnuPG/PGP einsetzt, wenden Sie sich an eine Linux User Group in Ihrer Nähe; dort finden Sie mit Sicherheit jemanden, der Wert auf sichere Kommunikation legt. Eine weitere Möglichkeit bietet die Zeitschrift c't an: Auf größeren Messen wie der CeBIT bietet die c't ebenfalls die Gelegenheit, Schlüssel zu signieren (http://www.heise.de/ct/pgpCA/).

Mit dem Kommando gpg --gen-key erzeugen Sie sich ein neues Schlüsselpaar:

```
bash-2.03$ gpg --gen-key gpg (GnuPG) 0.9.3; Copyright (C) 1999 Free
Software Foundation, Inc. This program comes with ABSOLUTELY
NO WARRANTY. This is free software, and you are welcome to
redistribute it under certain conditions. See the file COPYING for
details.  gpg: Warnung: Sensible Daten könnten auf Festplatte
ausgelagert werden.   Um dies zu vermeiden, kann das Programm
suid(root) installiert werden.   Bitte wenden Sie sich hierzu an den
Systemadministrator. gpg: /home/fr/.gnupg: Verzeichnis erzeugt gpg:
/home/fr/.gnupg/options: neue Optionendatei erstellt gpg:
/home/fr/.gnupg/secring.gpg: Schlüsselbund erstellt gpg:
/home/fr/.gnupg/pubring.gpg: Schlüsselbund erstellt gpg:
/home/fr/.gnupg/trustdb.gpg: trust-db erzeugt Bitte wählen Sie, welche
Art von Schlüssel Sie möchten:    (1) DSA und ElGamal (voreingestellt)
```

(2) DSA (nur signieren/beglaubigen) (4) ElGamal (signieren/beglaubigen und verschlüsseln) Ihre Auswahl?

Sie können hier zwischen verschiedenen Schlüsseln wählen; die voreingestellten Werte sind aber auf jeden Fall sinnvoll. Sie sollten diese einfach akzeptieren.

Der DSA Schlüssel wird 1024 Bits haben. Es wird ein neues ELG-E Schlüsselpaar erzeugt. kleinste Schlüssellänge ist 768 Bit standard Schlüssellänge ist 1024 Bit größte sinnvolle Schlüssellänge ist 2048 Bit Welche Schlüssellänge wünschen Sie? (1024)

Hier können Sie die Länge des zu erzeugenden Schlüssels festlegen. Längere Schlüssel sind schwerer zu „knacken“, allerdings benötigen diese auch etwas mehr Zeit, wenn man mit ihnen arbeitet. 1024 Bit sind ein sinnvoller Wert.

Die verlangte Schlüssellänge beträgt 1024 Bit Bitte wählen Sie, wie lange der Schlüssel gültig bleiben soll. 0 = Schlüssel verfällt nie <n> = Schlüssel verfällt nach n Tagen <n>w = Schlüssel verfällt nach n Wochen <n>m = Schlüssel verfällt nach n Monaten <n>y = Schlüssel verfällt nach n Jahren Der Schlüssel bleibt wie lange gültig? (0)

GnuPG unterstützt auch ein Verfallsdatum bei Schlüsseln. Vielleicht möchten Sie ja für die Mitarbeiter in Ihrer Firma, die nur Zeitverträge haben, verhindern, dass diese den Schlüssel auch später noch verwenden können. Für Ihren privaten Schlüssel benötigen Sie ein solches Verfallsdatum im Normalfall nicht.

Der Schlüssel verfällt nie. Ist dies richtig? (j/n) j

Sie benötigen eine User-ID, um Ihren Schlüssel eindeutig zu machen; das Programm baut diese User-ID aus Ihrem echten Namen, einem Kommentar und Ihrer E-Mail-Adresse in dieser Form auf: "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>" Ihr Name

("Vorname Nachname"): Frank Ronneburg E-Mail-Adresse: fr@leenuks.de Kommentar: Sie haben diese User-ID gewählt: "Frank Ronneburg <fr@leenuks.de>" Ändern: (N)ame, (K)ommentar, (E)-Mail oder (F)ertig/(B)eenden? f

Geben Sie hier mindestens Ihren Namen und Ihre E-Mail Adresse ein.

Sie benötigen ein Mantra, um den geheimen Schlüssel zu schützen. Geben Sie das Mantra ein: Geben Sie das Mantra nochmal ein:

Sie benötigen nun noch ein „Mantra“, also ein Passwort für Ihren privaten, geheimen Schlüssel.

Wir müssen eine ganze Menge Zufallszahlen erzeugen. Sie können dies unterstützen, indem Sie z.B. in einem anderen Fenster/Konsole irgendetwas tippen oder irgendwelche anderen Programme benutzen.
+++++.....+++++.....+++++.....++++++++++.....+++++
.....+++++.....+++++....++++++++++++++++++
++.+++++...+++++.....+++++....+++++....+++++
++++++++++++++...+++++.+++++.....+++++..+++++...+++++.....+
++++>+++++.+++++.....+++++ Wir müssen eine ganze Menge Zufallszahlen erzeugen. Sie können dies unterstützen, indem Sie z.B. in einem anderen Fenster/Konsole irgendetwas tippen oder irgendwelche anderen Programme benutzen.
.++++++++++.....+++++.....++++++++++.....
..+++++.....++++++++++.+++++.+++++.....+++++.++++++++++++
+.. ...+++++.....+++++.+++++...+++++.....+++++.+++
++....+++++.....++++++++++.....++++++++++.....+++++
>.....+++++.....>..+++++..<.....+++++...
.....>.....+++++.....+++++ Öffentlichen und geheimen Schlüssel erzeugt und signiert. bash-2.03$

GnuPG erzeugt nun aus Ihren Angaben ein Schlüsselpaar und legt dieses im Verzeichnis ~/.gnupg in Ihrem Heimat-Verzeichnis ab. Damit ist die eigentliche Erzeugung des Schlüssels schon abgeschlossen.

In diesem Abschnitt werden die grundlegenden Schritte der Schlüsselverwaltung beschrieben. Eine ausführliche Anleitung, auch zu allen anderen Funktionen von GnuPG, finden Sie in der deutschen Übersetzung des „GNU Privacy Handbook“ unter http://www.gnupg.org/gph/de/manual/.

Um mit anderen zu kommunizieren, müssen Sie untereinander Ihre öffentlichen Schlüssel austauschen. Dies kann per E-Mail, über einen Key-Server oder auch per persönlicher Übergabe erfolgen. Zum Auflisten der Schlüssel in Ihrem öffentlichen Schlüsselbund verwenden Sie die Befehlszeilen-Option --list-keys. Es werden alle Schlüssel angezeigt; die Suche kann beispielsweise durch einen Namen eingegrenzt werden.

```
fr@surimi:~$ gpg --list-keys Ronneburg pub 1024D/887EB817 2000-12-12 Frank Ronneburg <fr@tischbahn.de> uid Frank Ronneburg <fr@leenuks.de> uid Frank Ronneburg <fr@peace-love-and-linux.de> sub 1024g/B91B2CCC 2000-12-12 sub 2048g/AF72F22D 2002-03-08
```

Wenn Sie bereits weitere öffentliche Schlüssel oder weitere IDs hinzugefügt haben, wird diese Liste länger ausfallen. Beispielhaft findet sich in dieser Auflistung auch ein alter PGP-Key.

Um jemandem Ihren öffentlichen Schlüssel zu schicken, müssen Sie diesen zunächst exportieren. Hierzu benutzt man die Kommandozeilen-Option --export. Zur

Identifikation des zu exportierenden öffentlichen Schlüssels dient entweder die Schlüssel-ID oder irgendein Teil der Benutzer-ID.

```
gpg --output Ronneburg.gpg --export Ronneburg
```

Der Schlüssel wird in einem binären Format exportiert, doch kann dies unerwünscht sein, wenn Sie den Schlüssel per E-Mail verschicken oder auf einer WWW-Seite veröffentlichen wollen. GnuPG unterstützt daher die Kommandozeilen-Option --armor, die bewirkt, dass der Schlüssel im ASCII-Format ausgegeben wird. (Im Allgemeinen kann jede Ausgabe von GnuPG - beispielsweise Schlüssel, verschlüsselte Dokumente oder Unterschriften - im ASCII-Format dargestellt werden, indem man die Option --armor hinzufügt.)

```
fr@surimi:~$ gpg --armor --export Ronneburg -----BEGIN PGP
PUBLIC KEY BLOCK----- Version: GnuPG v1.0.4 (GNU/Linux)
Comment: Weitere Infos: siehe http://www.gnupg.org
mQGiBDo2SPYRBACkgxUYL30WWgBFybJWKc8eENKDC/8fWbm
sKLVFlmvayYu8ykeW
GiiUvh6zKhidoa7Vve796kq9N1v5VRvy4qKDMUvLlLA/4k4OHZg0r
KcQIIOzuGba
d3dGGCUcNqFYjlgnML2NP40+Kr4Rs6MAWk7gqZ4IofP4n9VmgOi
l1WzMmwCghZMB
orTO/atjGfWz/m30IgbpSisD/iw6mKPYdz7PJB50jCn6bpZt7dFAaQVJ
rlfSPS4J
iRUGPghVEPSfNtNd3N1PymRWv3j1CYC41Nl92wxLI8QFXdQv44s
mNdao7YDSdpkR
4+y+pWBCeitXqCj/9eVOVGvEM+mLmsvpg/M/qa6a+EF/iJW+3Mb6
3xmrumeHGBD6
GUllA/0f7B9u2Hqx4078QWZmxycpD8XI43Yq3r1pkHb28BryMi6tE/
qkfsgxywlD
Blnl6WTpzBxgsMLdm7BNz/B41fk8AnCAtnftoOa8+Q0TKodrDtLkr
XdW0z0RltNZ  ...
2NSr7iQUlIhGBBgRAgAGBQI6Nkj6AAoJEF+d1jKIfrgXJBwAnj038
AMs+feNTOyP
```

```
qkdlhcd+kHHTAJ0dKZe7keaAIJkz4FJQhBhrk/bHpw== =yDka -----
END PGP PUBLIC KEY BLOCK-----
```

Ein öffentlicher Schlüssel kann zu Ihrem öffentlichen Schlüsselbund hinzugefügt werden, und zwar mit der Option: --import.

```
fr@surimi:~$  gpg --import /tmp/frank.gpg  gpg: Schlüssel 887EB817:
Öffentlicher Schlüssel importiert gpg: Anzahl insgesamt bearbeiteter
Schlüssel: 1 gpg:            importiert: 1
```

Wenn ein Schlüssel einmal importiert ist, sollte er auf Authentizität überprüft werden. GnuPG arbeitet mit einem wirksamen und flexiblen Vertrauensmodell, bei dem Sie nicht jeden Schlüssel, den Sie importieren, persönlich zu authentifizieren brauchen. Einige Schlüssel können dies jedoch erfordern. Ein Schlüssel wird dadurch authentifiziert, dass Sie den Fingerabdruck des Schlüssels überpüfen und dann den Schlüssel unterschreiben, um seine Gültigkeit zu bestätigen. Der Fingerabdruck eines Schlüssels kann mit der Befehlszeilen-Option --fingerprint geprüft werden. Um aber den Schlüssel zu bestätigen, muss die Option --edit-key verwendet werden.

```
fr@surimi:~$ gpg --edit-key Borgert  gpg (GnuPG) 1.0.4; Copyright
(C) 2000 Free Software Foundation, Inc. This program comes with
ABSOLUTELY NO WARRANTY. This is free software, and you are
welcome to redistribute it under certain conditions. See the file
COPYING for details.   pub  1024R/9B668109  erstellt: 1998-06-28
verfällt: niemals Vertrauen: -/q (1)  W. Borgert <debacle@knorke.in-
berlin.de>  Befehl> fpr pub  1024R/9B668109 1998-06-28 W. Borgert
<debacle@knorke.in-berlin.de>        Fingerabdruck: 6F 74 32 AB 53
DD 09 F1  3B 69 E6 3F 79 8A 70 53
```

Um den Fingerabdruck zu überprüfen, müssen Sie den Eigentümer des Schlüssels kontaktieren und die Fingerabdrücke vergleichen. Sie können persönlich oder per Telefon mit ihm sprechen oder auf beliebigem anderen Wege kommunizieren, solange nur garantiert ist, dass es sich um den rechtmäßigen Eigentümer handelt. Stimmen beide Fingerabdrücke überein, dann können Sie sicher sein, dass Sie eine echte Kopie des öffentlichen Schlüssels haben.

Nach dem Prüfen des Fingerabdrucks können Sie den Schlüssel unterschreiben, um ihn zu authentifizieren. Da die Schlüsselüberprüfung ein Schwachpunkt in der Kryptographie mit öffentlichem Schlüssel ist, sollten Sie äußerste Sorgfalt walten lassen und den Fingerabdruck eines Schlüssels immer gemeinsam mit dem Eigentümer prüfen, bevor Sie den Schlüssel unterschreiben.

```
Befehl> sign            pub  1024R/9B668109  erstellt: 1998-06-28
verfällt: niemals Vertrauen: -/q       Fingerabdruck: 6F 74 32 AB 53 DD
09 F1  3B 69 E6 3F 79 8A 70 53       W. Borgert <debacle@knorke.in-
berlin.de>  Sind Sie wirklich sicher, dass Sie vorstehenden Schlüssel
mit Ihrem Schlüssel beglaubigen wollen: "Frank Ronneburg
<fr@leenuks.de>"  Wirklich unterschreiben?
```

Sie können sich jederzeit vergewissern, welche Unterschrift Sie hinzugefügt haben. Jede Benutzer-ID auf dem Schlüssel hat dann sowohl eine oder mehrere Eigenbeglaubigungen als auch eine Unterschrift von jedem Benutzer, der den Schlüssel authentifiziert hat.

```
Befehl> check uid  Frank Ronneburg <fr@leenuks.de> sig!
887EB817 2000-12-12   [Eigenbeglaubigung] sig!       9B668109 2001-
01-26   W. Borgert <debacle@knorke.in-berlin.de> sig!       801EA932
2001-04-02   Martin Schulze <joey@infodrom.north.de> sig!
13282FF2 2000-12-18   Werner Heuser (none) <wehe@debian.org>
sig!       EF439690 2000-12-21   Michael Piefel <piefel@informatik.hu-
ber sig!       258D8781 2001-03-31   Michael Bramer
<grisu@debian.org> sig!       DA4A1116 2001-04-04   Bernhard Reiter
<bernhard@intevation.de> sig!       86574ACA 2001-04-04   Georg C.
F. Greve <greve@gnu.org> sig!       496A1827 2001-04-06   Sebastian
Rittau <srittau@jroger.in-berl sig!       9C6D5E59 2001-04-10   Karl
Bartel <karlb@gmx.net>  Befehl>
```

Im Idealfall wird ein Schlüssel durch persönliche Übergabe an Ihre Korrespondenzpartner weitergegeben. In der Praxis werden jedoch Schlüssel oft per E-Mail oder über irgendein anderes elektronisches Kommunikationsmittel weitergegeben. Die Weitergabe per E-Mail ist durchaus annehmbar, wenn Sie nur einige wenige Korrespondenzpartner haben. Wenn Sie viele Korrespondenzpartner haben, könnten Sie beispielsweise Ihre(n) öffentlichen Schlüssel auf Ihrer Homepage im Web publizieren. Das setzt jedoch voraus, dass Ihre Korrespondenzpartner auch wissen, wo sie Ihre(n) Schlüssel finden können.

Um dieses Problem zu lösen, gibt es Key-Server, die öffentliche Schlüssel sammeln und weitergeben. Ein bei dem Server eingegangener öffentlicher Schlüssel wird entweder der Datenbank des Servers hinzugefügt oder mit Ihrem eventuell schon vorhandenen Schlüssel zusammengeführt. Wenn eine Anfrage nach einem Schlüssel beim Server eingeht, durchsucht dieser seine Datenbank und sendet den angeforderten öffentlichen Schlüssel zurück, wenn er ihn gefunden hat.

Ein oder mehr Schlüssel können unter Verwendung der Kommandozeilen-Option **--send-keys** an den Key-Server geschickt werden. Die Option erwartet eine Schlüssel-ID oder Benutzer-ID als Argument und schickt die so spezifizierten Schlüssel an den Key-Server. Der Key-Server, an den die Schlüssel geschickt werden sollen, wird durch die Kommandozeilenoption **--keyserver** spezifiziert. In ähnlicher Weise wird die Option **--recv-key** benutzt, um Schlüssel von einem Key-Server zu holen. Doch müssen Sie hier den Schlüssel mit einer Schlüssel-ID spezifizieren.

Hier ein Beispiel, mit dem ich meinen Public-Key auf dem Server aktualisiere.

```
fr@surimi:~$ gpg --send-key 887EB817 gpg: sende Schlüssel
887EB817 auf den hkp-Server subkeys.pgp.net gpg: sende Schlüssel
887EB817 auf den hkp-Server pool.sks-keyservers.net
```

Hier wird ein bereits vorhandener Key aktualisiert:

```
fr@surimi:~$ gpg --recv-key 887EB817 gpg: fordere Schlüssel
887EB817 von hkp-Server subkeys.pgp.net an gpg: Schlüssel
887EB817: "Frank Ronneburg <fr@tischbahn.de>" nicht geändert gpg:
Anzahl insgesamt bearbeiteter Schlüssel: 1 gpg:
unverändert: 1 gpg:                    neue Signaturen: 128
```

Weltweit gibt es eine Vielzahl bekannter Key-Server. Die größeren Key-Server synchronisieren sich wechselseitig. Am besten benutzen Sie einen gut erreichbaren Key-Server im Internet und tauschen dann regelmäßig über diesen Schlüssel aus.

Das Einbinden von GnuPG in das Mail-Programm mutt ist durch ein bei mutt mitgeliefertes Beispiel sehr einfach zu realisieren. Wenn Sie keine individuellen Einstellungen benötigen, reicht das Einfügen der Zeile

```
source /usr/share/doc/mutt/examples/gpg.rc
```

in der Datei ~/.muttrc aus. Natürlich können Sie auch eine Kopie der Beispieldatei in Ihrem Heimat-Verzeichnis erzeugen und die Werte an Ihre Bedürfnisse anpassen.

7.5 GnuPG

7.4 PGP Kapitel 8. Weitere Informationen

Kapitel 8. Weitere Informationen

7.5 GnuPG 8.2 Programmfehler (Bugs)

Inhaltsverzeichnis

In einer technischen Dokumentation lässt es sich nur schwer vermeiden, vollkommen auf englischsprachige Begriffe oder auf „Kunstwörter“ aus dem Wortschatz der „Community“ zu verzichten. Begriffe wie: „Hacker“, „FAQ“ oder „AFAIK“ finden sich aber auch in HOWTOs oder auf Mailinglisten.

In Netz gibt es die verschiedensten Sammlungen von üblichen Abkürzungen. Gute Einstiegspunkte für die weitere Lektüre sind:

V.E.R.A - Verzeichnis EDV-Relevanter Akronyme

FOLDOC - Free On-Line Dictionary Of Computing

Jargon - The New Hacker's Dictionary

Hier einige Erläuterungen zu den gebräuchlichsten Begriffen, insbesondere im Zusammenhang mit Debian:

AM - Application Maintainer, eine Person, die einen NM durch den NM-Prozess geleitet

BOD - Board Of Directors

BTS - Bug Tracking System

DAD - Debian Acronym Dictionary, Verzeichnis mit Debian Begriffen

DAM - Debian Account Manager, ein Debian Entwickler, der für das Einrichten neuer Benutzeraccounts zuständig ist

DD - Debian Developer, ein Debian Entwickler

DDP - Debian Documentation Project; in diesem Projekt wird die Dokumentation zu Debian erstellt und verwaltet.

DDR - Debian Developers Reference, Richtlinien für die Debian Entwicklung

DDTC - Debian Description Translation Client

DDTP - Debian Description Translation Project

DDTS - Debian Description Translation Server

DFSG - Debian Free Software Guidelines; siehe Debian Free Software Guidelines

DMUP - Debian Machine Use Policies, Richtlinien für die Benutzung der Debian Entwicklungsmaschinen

DPL - Debian Project Leader, die Person, die dem Projekt vorsteht und es nach außen repräsentiert

DSA - Debian Security Advisory oder Debian System Administration

DWN - Debian Weekly News, wöchentliche Neuigkeiten des Projekts

FHS - File Hierarchy Standard

FTBFS - Failure To Build From Source

IANADD - I Am Not A Debian Developer

ITA - Intend to Adopt, d.h., ein Maintainer möchte ein Paket von einem anderen übernehmen. Siehe auch WNPP.

ITO - Intend to Orphan, ein Maintainer beabsichtigt, ein Paket zur Übernahme freizugeben bzw. verwaisen zu lassen. Sie auch WNPP.

ITP - Intend to Package. Wenn ein Maintainer ein neues Paket erstellen möchte, wird dies zunächst auf der Mailingliste debian-devel angekündigt. Siehe auch WNPP.

MIA - Missing in Action, d.h. ein Maintainer, der keine Bugs beseitigt oder nicht auf Mails antwortet

NM - New Maintainer, d.h. jemand, der Maintainer werden möchte oder es gerade geworden ist.

NMU - Non Maintainer Upload, d.h., eine andere Person als der Maintainer hat ein Paket auf einen Server gelegt.

OPN - Open Projects Network, ein Netzwerk von Rechnern, die das IRC-Netzwerk beherbergen

PTS - Package Tracking System

QA - Quality Assurance

QoT - Quality of Translation

RC - Release Candidate oder Release Critical

RFA - Request for Adoption, d.h., ein Entwickler möchte ein Paket abgeben und bittet einen anderen Maintainer, das Paket zu übernehmen.

RFB - Request for Boot

RFH - Request for Help

RFP - Request for Packaging, d.h., ein Entwickler möchte eine Software als Debian Paket zusammenstellen und auf dem Server ablegen.

RFS - Request For Sponsorship

RL - Real Life - Das Gegenteil des „Virtual Life“, das sich im Netz, also in IRC-Channeln, Newsgroups und auf Mailinglisten, abspielt.

RTFM - Read the Fine Manual - weist freundlich darauf hin, doch zunächst einen Blick in die Dokumentation zu werfen. Häufig auch als „Read the Fucking Manual“ übersetzt.

SPI - Software in the Public Interest, Inc.

T&S - Tasks and Skills

UTSL - Use the Source, Luke - entlehnt aus den „Star Wars“-Filmen; es wird gebeten, doch einen Blick in den Quellcode zu werfen.

WNPP - Work-Needing and Prospective Packages - Liste von Paketen, die zurzeit ohne hinreichende Betreuung sind.

Kapitel 8. Weitere Informationen

7.5 GnuPG 8.2 Programmfehler (Bugs)

8.2 Programmfehler (Bugs)

Debian GNU/Linux nimmt die Sicherheit sehr ernst. Der größte Teil der Probleme, die bekannt werden, ist innerhalb von 48 Stunden korrigiert.

Die Erfahrung zeigt, dass „Sicherheit durch Verschleierung“ (englisch: security trough obscurity) nicht funktioniert. Die öffentliche „Ent-Schleierung“ wird durch Freigabe des Source-Codes zu Programmen erreicht und erlaubt es, bessere Lösungen für Sicherheitsprobleme schneller zu finden.

Mängel in der Sicherheit werden bei Debian GNU/Linux so lange verfolgt, bis eine korrigierte Version in die Distribution eingefügt wird. Wenn Sie in einem stabilen Release eine höhere Versionsnummer finden, dann benutzen Sie die neuere Version.

Wenn Sie selbst einen Fehler in einem Programm in Ihrer Debian GNU/Linux-Distribution finden, sehen Sie zunächst nach, ob dieses Problem bereits bekannt ist. Sollte dies nicht der Fall sein, so können Sie das Programm bug benutzen, um diesen Fehler an die Debian Entwickler zu melden.

Für allerneueste Sicherheitsinformationen über Debian tragen Sie sich in die Mailingliste debian-security-announce ein. Informationen hierzu finden Sie auf den Webseiten des Debian-Projekts.

Die Debian GNU/Linux-Distribution verwendet ein Fehlerverfolgungssystem, in dem Details von Fehlern, die von Benutzern oder Entwicklern gemeldet wurden, gespeichert werden. Jeder gefundene Fehler bekommt eine Nummer zugewiesen und wird so lange gespeichert, bis der Fehler beseitigt ist.

Sie finden eine Liste aller bekannten Fehler unter http://www.debian.org/Bugs/ im Web.

8.2 Programmfehler (Bugs)

Kapitel 8. Weitere Informationen | 8.3 Debian Gesellschaftsvertrag

8.3 Debian Gesellschaftsvertrag

8.2 Programmfehler (Bugs) 8.4 Debian Richtlinien für freie Software

„Gesellschaftsvertrag“ mit der Gemeinschaft für Freie Software

Wir sind „Software In The Public Interest“-Hersteller des Debian GNU/Linux-Systems. Wir bieten diesen „Gesellschaftsvertrag“ der Gemeinschaft für Freie Software an. (Mit „Gemeinschaft für Freie Software“ werden alle Hersteller und Anwender freier Software bezeichnet.)

1. Debian wird 100% Freie Software bleiben

Wir versprechen, dass die Debian GNU/Linux-Distribution auch weiterhin vollständig aus freier Software bestehen wird. Da es viele verschiedene Auslegungen des Begriffs „freie Software“ gibt, haben wir weiter unten die Richtlinien aufgeführt, nach denen wir freie Software identifizieren. Trotzdem werden wir Anwender unterstützen, die nicht-freie Programme einsetzen oder entwickeln. Wir werden aber niemals das Gesamtsystem von nicht-freier Software abhängig machen.

2. Unser Beitrag zur Gemeinschaft für Freie Software

Wenn wir neue Komponenten des Debian Systems schreiben, so werden wir sie als freie Software lizenzieren. Wir werden das bestmögliche System erstellen, so daß freie Software weit verbreitet und genutzt wird. Wir werden Korrekturen, Verbesserungen, Anwenderwünsche usw. an die ursprünglichen („upstream“) Autoren weiterleiten, deren Programme in unser System integriert wurden.

3. Wir werden Probleme nicht verbergen

Wir werden unsere Fehlerdatenbank für alle Zeiten öffentlich betreiben. Fehlermeldungen, die von Anwendern online abgeschickt werden, werden augenblicklich für andere sichtbar.

4. Unsere Prioritäten sind unsere Anwender und freie Software

Wir orientieren uns an den Bedürfnissen unserer Anwender und der Gemeinschaft für Freie Software. Ihre Interessen stehen an erster Stelle. Wir werden unsere Nutzer bei ihrer Arbeit mit den verschiedensten Rechnerumgebungen unterstützen. Wir haben nichts dagegen, dass kommerzielle Software auf Debian Systemen eingesetzt wird. Außerdem erlauben wir anderen, eine erweiterte („Value-Added“) Distribution zu erstellen, die Debian und kommerzielle Software enthält, ohne dafür irgendwelche Gebühren zu erheben. Um diese Ziele zu erreichen, werden wir ein integriertes System von hoher Qualität und 100% freier Software anbieten, die die gerade beschriebene Nutzung nicht durch rechtliche Einschränkungen wie z. B. durch Lizenzverträge verhindert.

5. Programme, die nicht unseren Standards für freie Software genügen

Wir wissen, dass einige unserer Anwender unbedingt Programme einsetzen müssen, die nicht den Debian Richtlinien für freie Software entsprechen. Für solche Programme

haben wird die zusätzlichen Bereiche „contrib“ und „non-free“ auf unserem FTP-Archiv eingerichtet. Die Software in diesen Verzeichnissen ist nicht Bestandteil des Debian Systems, wurde aber trotzdem für den Einsatz in einem Debian System vorbereitet. Wir empfehlen den CD-Herstellern, die jeweiligen Lizenzbestimmungen der Programmpakete in diesen Verzeichnissen zu studieren und selbst zu entscheiden, ob Sie die Programme mit Ihren CDs verteilen dürfen. Obwohl die Programme aus „non-free“ nicht Bestandteil der Debian Distribution sind, unterstützen wir ihren Einsatz und bieten Infrastruktur für diese nicht freien Programme an, z.B. unsere Fehlerdatenbank und die Mailinglisten.

8.3 Debian Gesellschaftsvertrag

8.2 Programmfehler (Bugs) 8.4 Debian Richtlinien für freie Software

8.4 Debian Richtlinien für freie Software

8.3 Debian Gesellschaftsvertrag 8.5 Creative Commons Lizenz (by-nd)

(englisch: Debian Free Software Guidelines)

1. Unbeschränkte Weitergabe

Ein Bestandteil der Debian Distribution darf durch seine Lizenz nicht verhindern, dass irgend jemand diese Software als Bestandteil einer Software-Distribution, die Programme aus den verschiedensten Quellen enthält, verkauft oder weitergibt. Die Lizenz darf keine Abgaben oder sonstige Leistungen für einen solchen Verkauf fordern.

2. Quellcode

Das Programm muss im Quellcode vorliegen, und es muss die Weitergabe sowohl im Quellcode als auch in kompilierter Form erlaubt sein.

3. Weiterführende Arbeiten

Die Lizenz muss Veränderungen und weiterführende Arbeiten gestatten und es erlauben, dass diese unter den gleichen Lizenzbedingungen weitergegeben werden dürfen wie die Original-Software.

4. Integrität des ursprünglichen Quellcodes

Die Lizenz darf die Weitergabe von verändertem Quellcode nur dann verbieten, wenn sie die Weitergabe von sogenannten Patch-Dateien mit dem Quellcode erlaubt, die dazu dienen, das Programm vor seiner Herstellung zu modifizieren. Die Lizenz muss ausdrücklich die Weitergabe der aus dem veränderten Quellcode erzeugten Programme erlauben. Die Lizenz darf fordern, dass die veränderten Programme einen anderen Namen oder eine andere Versionsnummer tragen müssen.

(Dies ist ein Kompromiss. Die Debian Gruppe ermutigt alle Autoren, Veränderungen an Dateien sowohl im Quellcode als auch in Binärform zu erlauben.)

5. Keine Diskriminierung von Personen oder Gruppen

Die Lizenz darf keine Person oder Gruppe von Personen diskriminieren.

6. Keine Diskriminierung von Einsatzbereichen

Die Lizenz darf keine Einschränkungen hinsichtlich des Einsatzbereichs vornehmen. Beispielsweise darf sie nicht verhindern, dass das Programm geschäftlich oder für genetische Forschungen verwendet wird.

7. Weitergabe der Lizenz

Die mit einem Programm verbundenen Rechte müssen für alle gelten, die das Programm erhalten, ohne dass es für sie notwendig ist, eine zusätzliche Lizenz zu erwerben.

8. Keine spezielle Lizenz für Debian

Die mit dem Programm verbundenen Rechte dürfen nicht davon abhängig sein, dass das Programm Teil des Debian Systems ist. Falls das Programm aus der Debian Distribution herausgenommen wird und ohne Debian genutzt oder vertrieben werden soll, ansonsten aber im Rahmen der Programmlizenz bleibt, so müssen alle Parteien, die das Programm bekommen, die gleichen Rechte haben, wie sie im Zusammenhang mit dem Debian System gewährt wurden.

9. Keine Auswirkungen auf andere Programme

Die Lizenz darf keine Beschränkungen besitzen, die Auswirkungen auf andere Software hat, die mit diesem Programm weitergegeben wird. Beispielsweise darf die Lizenz nicht vorschreiben, dass alle anderen Programme auf dem gleichen Medium freie Software sein müssen.

10. Beispiellizenzen

Die „GPL“- „BSD“- und „Artistic“-Lizenzen sind Beispiele für Lizenzen, die wir als „frei“ betrachten.

Dies ist die deutsche Übersetzung von „Debian's social contract with the free software community“. In Zweifelsfällen ist das englische Original maßgeblich. Es ist beispielsweise unter www.debian.org/social_contract.de.html verfügbar.

8.4 Debian Richtlinien für freie Software

◁ 8.3 Debian Gesellschaftsvertrag △ 8.5 Creative Commons Lizenz (by-nd) ▷

8.5 Creative Commons Lizenz (by-nd)

8.4 Debian Richtlinien für freie Software 8.6 Creative Commons Lizenz (by-nc-nd)

Namensnennung - Keine Bearbeitung - Version 2.0

CREATIVE COMMONS IST KEINE RECHTSANWALTSGESELLSCHAFT UND LEISTET KEINE RECHTSBERATUNG. DIE WEITERGABE DIESES LIZENZENTWURFES FÜHRT ZU KEINEM MANDATSVERHÄLTNIS. CREATIVE COMMONS ERBRINGT DIESE INFORMATIONEN OHNE GEWÄHR. CREATIVE COMMONS ÜBERNIMMT KEINE GEWÄHRLEISTUNG FÜR DIE GELIEFERTEN INFORMATIONEN UND SCHLIESST DIE HAFTUNG FÜR SCHÄDEN AUS, DIE SICH AUS IHREM GEBRAUCH ERGEBEN.

Lizenzvertrag

DAS URHEBERRECHTLICH GESCHÜTZTE WERK ODER DER SONSTIGE SCHUTZGEGENSTAND (WIE UNTEN BESCHRIEBEN) WIRD UNTER DEN BEDINGUNGEN DIESER CREATIVE COMMONS PUBLIC LICENSE („CCPL" ODER „LIZENZVERTRAG") ZUR VERFÜGUNG GESTELLT. DER SCHUTZGEGENSTAND IST DURCH DAS URHEBERRECHT UND/ODER EINSCHLÄGIGE GESETZE GESCHÜTZT.

DURCH DIE AUSÜBUNG EINES DURCH DIESEN LIZENZVERTRAG GEWÄHRTEN RECHTS AN DEM SCHUTZGEGENSTAND ERKLÄREN SIE SICH MIT DEN LIZENZBEDINGUNGEN RECHTSVERBINDLICH EINVERSTANDEN. DER LIZENZGEBER RÄUMT IHNEN DIE HIER BESCHRIEBENEN RECHTE UNTER DER VORAUSSETZUNG EIN, DASS SIE SICH MIT DIESEN VERTRAGSBEDINGUNGEN EINVERSTANDEN ERKLÄREN.

Definitionen

Unter einer *Bearbeitung* wird eine Übersetzung oder andere Bearbeitung des Werkes verstanden, die Ihre persönliche geistige Schöpfung ist. Eine freie Benutzung des Werkes wird nicht als Bearbeitung angesehen.

Unter den *Lizenzelementen* werden die folgenden Lizenzcharakteristika verstanden, die vom Lizenzgeber ausgewählt und in der Bezeichnung der Lizenz genannt werden: *Namensnennung*, *Nicht-kommerziell*, *Weitergabe unter gleichen Bedingungen*.

Unter dem *Lizenzgeber* wird die natürliche oder juristische Person verstanden, die den Schutzgegenstand unter den Bedingungen dieser Lizenz anbietet.

Unter einem *Sammelwerk* wird eine Sammlung von Werken, Daten oder anderen unabhängigen Elementen verstanden, die aufgrund der Auswahl oder Anordnung der Elemente eine persönliche geistige Schöpfung ist. Darunter fallen auch solche Sammelwerke, deren Elemente systematisch oder methodisch angeordnet und einzeln mit Hilfe elektronischer Mittel oder auf andere Weise zugänglich sind (Datenbankwerke). Ein Sammelwerk wird im Zusammenhang mit dieser Lizenz nicht als Bearbeitung (wie oben beschrieben) angesehen.

Mit *Sie* und *Ihnen* ist die natürliche oder juristische Person gemeint, die die durch diese Lizenz gewährten Nutzungsrechte ausübt und die zuvor die Bedingungen dieser Lizenz im Hinblick auf das Werk nicht verletzt hat oder die die ausdrückliche Erlaubnis des

Lizenzgebers erhalten hat, die durch diese Lizenz gewährten Nutzungsrechte trotz einer vorherigen Verletzung auszuüben.

Unter dem *Schutzgegenstand* wird das Werk oder Sammelwerk oder das Schutzobjekt eines verwandten Schutzrechts, das Ihnen unter den Bedingungen dieser Lizenz angeboten wird, verstanden.

Unter dem *Urheber* wird die natürliche Person verstanden, die das Werk geschaffen hat.

Unter einem *verwandten Schutzrecht* wird das Recht an einem anderen urheberrechtlichen Schutzgegenstand als einem Werk verstanden, zum Beispiel einer wissenschaftlichen Ausgabe, einem nachgelassenen Werk, einem Lichtbild, einer Datenbank, einem Tonträger, einer Funksendung, einem Laufbild oder einer Darbietung eines ausübenden Künstlers.

Unter dem *Werk* wird eine persönliche geistige Schöpfung verstanden, die Ihnen unter den Bedingungen dieser Lizenz angeboten wird.

Schranken des Urheberrechts. Diese Lizenz lässt sämtliche Befugnisse unberührt, die sich aus den Schranken des Urheberrechts, aus dem Erschöpfungsgrundsatz oder anderen Beschränkungen der Ausschließlichkeitsrechte des Rechtsinhabers ergeben.

Lizenzierung. Unter den Bedingungen dieses Lizenzvertrages räumt Ihnen der Lizenzgeber ein lizenzgebührenfreies, räumlich und zeitlich (für die Dauer des Urheberrechts oder verwandten Schutzrechts) unbeschränktes einfaches Nutzungsrecht ein, den Schutzgegenstand in der folgenden Art und Weise zu nutzen:

den Schutzgegenstand in körperlicher Form zu verwerten, insbesondere zu vervielfältigen, zu verbreiten und auszustellen;

den Schutzgegenstand in unkörperlicher Form öffentlich wiederzugeben, insbesondere vorzutragen, aufzuführen und vorzuführen, öffentlich zugänglich zu machen, zu senden, durch Bild- und Tonträger wiederzugeben sowie Funksendungen und öffentliche Zugänglichmachungen wiederzugeben;

den Schutzgegenstand auf Bild- oder Tonträger aufzunehmen, Lichtbilder davon herzustellen, weiterzusenden und in dem in a. und b. genannten Umfang zu verwerten;

Die genannten Nutzungsrechte können für alle bekannten Nutzungsarten ausgeübt werden. Die genannten Nutzungsrechte beinhalten das Recht, solche Veränderungen an dem Werk vorzunehmen, die technisch erforderlich sind, um die Nutzungsrechte für alle Nutzungsarten wahrzunehmen. Insbesondere sind davon die Anpassung an andere Medien und auf andere Dateiformate umfasst.

Beschränkungen. Die Einräumung der Nutzungsrechte gemäß Ziffer 3 erfolgt ausdrücklich nur unter den folgenden Bedingungen:

Sie dürfen den Schutzgegenstand ausschließlich unter den Bedingungen dieser Lizenz vervielfältigen, verbreiten oder öffentlich wiedergeben, und Sie müssen stets eine Kopie oder die vollständige Internetadresse in Form des Uniform-Resource-Identifier (URI)

dieser Lizenz beifügen, wenn Sie den Schutzgegenstand vervielfältigen, verbreiten oder öffentlich wiedergeben. Sie dürfen keine Vertragsbedingungen anbieten oder fordern, die die Bedingungen dieser Lizenz oder die durch sie gewährten Rechte ändern oder beschränken. Sie dürfen den Schutzgegenstand nicht unterlizenzieren. Sie müssen alle Hinweise unverändert lassen, die auf diese Lizenz und den Haftungsausschluss hinweisen. Sie dürfen den Schutzgegenstand mit keinen technischen Schutzmaßnahmen versehen, die den Zugang oder den Gebrauch des Schutzgegenstandes in einer Weise kontrollieren, die mit den Bedingungen dieser Lizenz im Widerspruch stehen. Die genannten Beschränkungen gelten auch für den Fall, dass der Schutzgegenstand einen Bestandteil eines Sammelwerkes bildet; sie verlangen aber nicht, dass das Sammelwerk insgesamt zum Gegenstand dieser Lizenz gemacht wird. Wenn Sie ein Sammelwerk erstellen, müssen Sie - soweit dies praktikabel ist - auf die Mitteilung eines Lizenzgebers oder Urhebers hin aus dem Sammelwerk jeglichen Hinweis auf diesen Lizenzgeber oder diesen Urheber entfernen. Wenn Sie den Schutzgegenstand bearbeiten, müssen Sie - soweit dies praktikabel ist- auf die Aufforderung eines Rechtsinhabers hin von der Bearbeitung jeglichen Hinweis auf diesen Rechtsinhaber entfernen.

Wenn Sie den Schutzgegenstand oder ein Sammelwerk vervielfältigen, verbreiten oder öffentlich wiedergeben, müssen Sie alle Urhebervermerke für den Schutzgegenstand unverändert lassen und die Urheberschaft oder Rechtsinhaberschaft in einer der von Ihnen vorgenommenen Nutzung angemessenen Form anerkennen, indem Sie den Namen (oder das Pseudonym, falls ein solches verwendet wird) des Urhebers oder Rechteinhabers nennen, wenn dieser angegeben ist. Dies gilt auch für den Titel des Schutzgegenstandes, wenn dieser angeben ist, sowie - in einem vernünftigerweise durchführbaren Umfang - für die mit dem Schutzgegenstand zu verbindende Internetadresse in Form des Uniform-Resource-Identifier (URI), wie sie der Lizenzgeber angegeben hat, sofern dies geschehen ist, es sei denn, diese Internetadresse verweist nicht auf den Urhebervermerk oder die Lizenzinformationen zu dem Schutzgegenstand. Ein solcher Hinweis kann in jeder angemessenen Weise erfolgen, wobei jedoch bei einer Datenbank oder einem Sammelwerk der Hinweis zumindest an gleicher Stelle und in ebenso auffälliger Weise zu erfolgen hat wie vergleichbare Hinweise auf andere Rechtsinhaber.

Obwohl die gemäß Ziffer 3 gewährten Nutzungsrechte in umfassender Weise ausgeübt werden dürfen, findet diese Erlaubnis ihre gesetzliche Grenze in den Persönlichkeitsrechten der Urheber und ausübenden Künstler, deren berechtigte geistige und persönliche Interessen bzw. deren Ansehen oder Ruf nicht dadurch gefährdet werden dürfen, dass ein Schutzgegenstand über das gesetzlich zulässige Maß hinaus beeinträchtigt wird.

Gewährleistung. Sofern dies von den Vertragsparteien nicht anderweitig schriftlich vereinbart ist, bietet der Lizenzgeber keine Gewährleistung für die erteilten Rechte, außer für den Fall, dass Mängel arglistig verschwiegen wurden. Für Mängel anderer Art, insbesondere bei der mangelhaften Lieferung von Verkörperungen des Schutzgegenstandes, richtet sich die Gewährleistung nach der Regelung, die die Person, die Ihnen den Schutzgegenstand zur Verfügung stellt, mit Ihnen außerhalb dieser Lizenz vereinbart, oder - wenn eine solche Regelung nicht getroffen wurde - nach den gesetzlichen Vorschriften.

Haftung. Über die in Ziffer 5 genannte Gewährleistung hinaus haftet Ihnen der Lizenzgeber nur für Vorsatz und grobe Fahrlässigkeit.

Vertragsende

Dieser Lizenzvertrag und die durch ihn eingeräumten Nutzungsrechte enden automatisch bei jeder Verletzung der Vertragsbedingungen durch Sie. Für natürliche und juristische Personen, die von Ihnen eine Datenbank oder ein Sammelwerk unter diesen Lizenzbedingungen erhalten haben, gilt die Lizenz jedoch weiter, vorausgesetzt, diese natürlichen oder juristischen Personen erfüllen sämtliche Vertragsbedingungen. Die Ziffern 1, 2, 5, 6, 7 und 8 gelten bei einer Vertragsbeendigung fort.

Unter den oben genannten Bedingungen erfolgt die Lizenz auf unbegrenzte Zeit (für die Dauer des Schutzrechts). Dennoch behält sich der Lizenzgeber das Recht vor, den Schutzgegenstand unter anderen Lizenzbedingungen zu nutzen oder die eigene Weitergabe des Schutzgegenstandes jederzeit zu beenden, vorausgesetzt, dass solche Handlungen nicht dem Widerruf dieser Lizenz dienen (oder jeder anderen Lizenzierung, die auf Grundlage dieser Lizenz erfolgt ist oder erfolgen muss) und diese Lizenz wirksam bleibt, bis Sie unter den oben genannten Voraussetzungen endet.

Schlussbestimmungen

Jedes Mal, wenn Sie den Schutzgegenstand vervielfältigen, verbreiten oder öffentlich wiedergeben, bietet der Lizenzgeber dem Erwerber eine Lizenz für den Schutzgegenstand unter denselben Vertragsbedingungen an, unter denen er Ihnen die Lizenz eingeräumt hat.

Sollte eine Bestimmung dieses Lizenzvertrages unwirksam sein, so wird die Wirksamkeit der übrigen Lizenzbestimmungen dadurch nicht berührt, und an die Stelle der unwirksamen Bestimmung tritt eine Ersatzregelung, die dem mit der unwirksamen Bestimmung angestrebten Zweck am nächsten kommt.

Nichts soll dahingehend ausgelegt werden, dass auf eine Bestimmung dieses Lizenzvertrages verzichtet oder einer Vertragsverletzung zugestimmt wird, so lange ein solcher Verzicht oder eine solche Zustimmung nicht schriftlich vorliegen und von der verzichtenden oder zustimmenden Vertragspartei unterschrieben sind.

Dieser Lizenzvertrag stellt die vollständige Vereinbarung zwischen den Vertragsparteien hinsichtlich des Schutzgegenstandes dar. Es gibt keine weiteren ergänzenden Vereinbarungen oder mündlichen Abreden im Hinblick auf den Schutzgegenstand. Der Lizenzgeber ist an keine zusätzlichen Abreden gebunden, die aus irgendeiner Absprache mit Ihnen entstehen könnten. Der Lizenzvertrag kann nicht ohne eine übereinstimmende schriftliche Vereinbarung zwischen dem Lizenzgeber und Ihnen abgeändert werden.

Auf diesen Lizenzvertrag findet das Recht der Bundesrepublik Deutschland Anwendung.

CREATIVE COMMONS IST KEINE VERTRAGSPARTEI DIESES LIZENZVERTRAGES UND ÜBERNIMMT KEINERLEI GEWÄHRLEISTUNG FÜR DAS WERK. CREATIVE COMMONS IST IHNEN ODER DRITTEN GEGENÜBER NICHT HAFTBAR FÜR SCHÄDEN JEDWEDER ART. UNGEACHTET DER VORSTEHENDEN ZWEI (2) SÄTZE HAT CREATIVE COMMONS ALLE

RECHTE UND PFLICHTEN EINES LIZENZGEBERS, WENN SICH CREATIVE COMMONS AUSDRÜCKLICH ALS LIZENZGEBER BEZEICHNET.

AUSSER FÜR DEN BESCHRÄNKTEN ZWECK EINES HINWEISES AN DIE ÖFFENTLICHKEIT, DASS DAS WERK UNTER DER CCPL LIZENZIERT WIRD, DARF KEINE VERTRAGSPARTEI DIE MARKE "CREATIVE COMMONS" ODER EINE ÄHNLICHE MARKE ODER DAS LOGO VON CREATIVE COMMONS OHNE VORHERIGE GENEHMIGUNG VON CREATIVE COMMONS NUTZEN. JEDE GESTATTETE NUTZUNG HAT IN ÜBEREINSTIMMUNG MIT DEN JEWEILS GÜLTIGEN NUTZUNGSBEDINGUNGEN FÜR MARKEN VON CREATIVE COMMONS ZU ERFOLGEN, WIE SIE AUF DER WEBSITE ODER IN ANDERER WEISE AUF ANFRAGE VON ZEIT ZU ZEIT ZUGÄNGLICH GEMACHT WERDEN.

CREATIVE COMMONS KANN UNTER http://creativecommons.org KONTAKTIERT WERDEN.

8.5 Creative Commons Lizenz (by-nd)

8.4 Debian Richtlinien für freie Software 8.6 Creative Commons Lizenz (by-nc-nd)

8.6 Creative Commons Lizenz (by-nc-nd)

8.5 Creative Commons Lizenz (by-nd) 8.7 GNU Public License (deutsche Übersetzung)

Namensnennung - Nicht-kommerziell - Keine Bearbeitung - Version 2.0

CREATIVE COMMONS IST KEINE RECHTSANWALTSGESELLSCHAFT UND LEISTET KEINE RECHTSBERATUNG. DIE WEITERGABE DIESES LIZENZENTWURFES FÜHRT ZU KEINEM MANDATSVERHÄLTNIS. CREATIVE COMMONS ERBRINGT DIESE INFORMATIONEN OHNE GEWÄHR. CREATIVE COMMONS ÜBERNIMMT KEINE GEWÄHRLEISTUNG FÜR DIE GELIEFERTEN INFORMATIONEN UND SCHLIESST DIE HAFTUNG FÜR SCHÄDEN AUS, DIE SICH AUS IHREM GEBRAUCH ERGEBEN.

Lizenzvertrag

DAS URHEBERRECHTLICH GESCHÜTZTE WERK ODER DER SONSTIGE SCHUTZGEGENSTAND (WIE UNTEN BESCHRIEBEN) WIRD UNTER DEN BEDINGUNGEN DIESER CREATIVE COMMONS PUBLIC LICENSE („CCPL" ODER „LIZENZVERTRAG") ZUR VERFÜGUNG GESTELLT. DER SCHUTZGEGENSTAND IST DURCH DAS URHEBERRECHT UND/ODER EINSCHLÄGIGE GESETZE GESCHÜTZT.

DURCH DIE AUSÜBUNG EINES DURCH DIESEN LIZENZVERTRAG GEWÄHRTEN RECHTS AN DEM SCHUTZGEGENSTAND ERKLÄREN SIE SICH MIT DEN LIZENZBEDINGUNGEN RECHTSVERBINDLICH EINVERSTANDEN. DER LIZENZGEBER RÄUMT IHNEN DIE HIER BESCHRIEBENEN RECHTE UNTER DER VORAUSSETZUNG EIN, DASS SIE SICH MIT DIESEN VERTRAGSBEDINGUNGEN EINVERSTANDEN ERKLÄREN.

Definitionen

Unter einer *Bearbeitung* wird eine Übersetzung oder andere Bearbeitung des Werkes verstanden, die Ihre persönliche geistige Schöpfung ist. Eine freie Benutzung des Werkes wird nicht als Bearbeitung angesehen.

Unter den *Lizenzelementen* werden die folgenden Lizenzcharakteristika verstanden, die vom Lizenzgeber ausgewählt und in der Bezeichnung der Lizenz genannt werden: *Namensnennung*, *Nicht-kommerziell*, *Weitergabe unter gleichen Bedingungen*.

Unter dem *Lizenzgeber* wird die natürliche oder juristische Person verstanden, die den Schutzgegenstand unter den Bedingungen dieser Lizenz anbietet.

Unter einem *Sammelwerk* wird eine Sammlung von Werken, Daten oder anderen unabhängigen Elementen verstanden, die aufgrund der Auswahl oder Anordnung der Elemente eine persönliche geistige Schöpfung ist. Darunter fallen auch solche Sammelwerke, deren Elemente systematisch oder methodisch angeordnet und einzeln mit Hilfe elektronischer Mittel oder auf andere Weise zugänglich sind (Datenbankwerke). Ein Sammelwerk wird im Zusammenhang mit dieser Lizenz nicht als Bearbeitung (wie oben beschrieben) angesehen.

Mit *Sie* und *Ihnen* ist die natürliche oder juristische Person gemeint, die die durch diese Lizenz gewährten Nutzungsrechte ausübt und die zuvor die Bedingungen dieser Lizenz im Hinblick auf das Werk nicht verletzt hat oder die die ausdrückliche Erlaubnis des

Lizenzgebers erhalten hat, die durch diese Lizenz gewährten Nutzungsrechte trotz einer vorherigen Verletzung auszuüben.

Unter dem *Schutzgegenstand* wird das Werk oder Sammelwerk oder das Schutzobjekt eines verwandten Schutzrechts, das Ihnen unter den Bedingungen dieser Lizenz angeboten wird, verstanden.

Unter dem *Urheber* wird die natürliche Person verstanden, die das Werk geschaffen hat.

Unter einem *verwandten Schutzrecht* wird das Recht an einem anderen urheberrechtlichen Schutzgegenstand als einem Werk verstanden, zum Beispiel einer wissenschaftlichen Ausgabe, einem nachgelassenen Werk, einem Lichtbild, einer Datenbank, einem Tonträger, einer Funksendung, einem Laufbild oder einer Darbietung eines ausübenden Künstlers.

Unter dem *Werk* wird eine persönliche geistige Schöpfung verstanden, die Ihnen unter den Bedingungen dieser Lizenz angeboten wird.

Schranken des Urheberrechts. Diese Lizenz lässt sämtliche Befugnisse unberührt, die sich aus den Schranken des Urheberrechts, aus dem Erschöpfungsgrundsatz oder anderen Beschränkungen der Ausschließlichkeitsrechte des Rechtsinhabers ergeben.

Lizenzierung. Unter den Bedingungen dieses Lizenzvertrages räumt Ihnen der Lizenzgeber ein lizenzgebührenfreies, räumlich und zeitlich (für die Dauer des Urheberrechts oder verwandten Schutzrechts) unbeschränktes einfaches Nutzungsrecht ein, den Schutzgegenstand in der folgenden Art und Weise zu nutzen:

den Schutzgegenstand in körperlicher Form zu verwerten, insbesondere zu vervielfältigen, zu verbreiten und auszustellen;

den Schutzgegenstand in unkörperlicher Form öffentlich wiederzugeben, insbesondere vorzutragen, aufzuführen und vorzuführen, öffentlich zugänglich zu machen, zu senden, durch Bild- und Tonträger wiederzugeben sowie Funksendungen und öffentliche Zugänglichmachungen wiederzugeben;

den Schutzgegenstand auf Bild- oder Tonträger aufzunehmen, Lichtbilder davon herzustellen, weiterzusenden und in dem in a. und b. genannten Umfang zu verwerten;

Die genannten Nutzungsrechte können für alle bekannten Nutzungsarten ausgeübt werden. Die genannten Nutzungsrechte beinhalten das Recht, solche Veränderungen an dem Werk vorzunehmen, die technisch erforderlich sind, um die Nutzungsrechte für alle Nutzungsarten wahrzunehmen. Insbesondere sind davon die Anpassung an andere Medien und auf andere Dateiformate umfasst.

Beschränkungen. Die Einräumung der Nutzungsrechte gemäß Ziffer 3 erfolgt ausdrücklich nur unter den folgenden Bedingungen:

Sie dürfen den Schutzgegenstand ausschließlich unter den Bedingungen dieser Lizenz vervielfältigen, verbreiten oder öffentlich wiedergeben, und Sie müssen stets eine Kopie oder die vollständige Internetadresse in Form des Uniform-Resource-Identifier (URI)

dieser Lizenz beifügen, wenn Sie den Schutzgegenstand vervielfältigen, verbreiten oder öffentlich wiedergeben. Sie dürfen keine Vertragsbedingungen anbieten oder fordern, die die Bedingungen dieser Lizenz oder die durch sie gewährten Rechte ändern oder beschränken. Sie dürfen den Schutzgegenstand nicht unterlizenzieren. Sie müssen alle Hinweise unverändert lassen, die auf diese Lizenz und den Haftungsausschluss hinweisen. Sie dürfen den Schutzgegenstand mit keinen technischen Schutzmaßnahmen versehen, die den Zugang oder den Gebrauch des Schutzgegenstandes in einer Weise kontrollieren, die mit den Bedingungen dieser Lizenz im Widerspruch stehen. Die genannten Beschränkungen gelten auch für den Fall, dass der Schutzgegenstand einen Bestandteil eines Sammelwerkes bildet; sie verlangen aber nicht, dass das Sammelwerk insgesamt zum Gegenstand dieser Lizenz gemacht wird. Wenn Sie ein Sammelwerk erstellen, müssen Sie - soweit dies praktikabel ist - auf die Mitteilung eines Lizenzgebers oder Urhebers hin aus dem Sammelwerk jeglichen Hinweis auf diesen Lizenzgeber oder diesen Urheber entfernen. Wenn Sie den Schutzgegenstand bearbeiten, müssen Sie - soweit dies praktikabel ist- auf die Aufforderung eines Rechtsinhabers hin von der Bearbeitung jeglichen Hinweis auf diesen Rechtsinhaber entfernen.

Sie dürfen die in Ziffer 3 gewährten Nutzungsrechte in keiner Weise verwenden, die hauptsächlich auf einen geschäftlichen Vorteil oder eine vertraglich geschuldete geldwerte Vergütung abzielt oder darauf gerichtet ist. Erhalten Sie im Zusammenhang mit der Einräumung der Nutzungsrechte ebenfalls einen Schutzgegenstand, ohne dass eine vertragliche Verpflichtung hierzu besteht, so wird dies nicht als geschäftlicher Vorteil oder vertraglich geschuldete geldwerte Vergütung angesehen, wenn keine Zahlung oder geldwerte Vergütung in Verbindung mit dem Austausch der Schutzgegenstände geleistet wird (z.B. File-Sharing).

Wenn Sie den Schutzgegenstand oder ein Sammelwerk vervielfältigen, verbreiten oder öffentlich wiedergeben, müssen Sie alle Urhebervermerke für den Schutzgegenstand unverändert lassen und die Urheberschaft oder Rechtsinhaberschaft in einer der von Ihnen vorgenommenen Nutzung angemessenen Form anerkennen, indem Sie den Namen (oder das Pseudonym, falls ein solches verwendet wird) des Urhebers oder Rechteinhabers nennen, wenn dieser angegeben ist. Dies gilt auch für den Titel des Schutzgegenstandes, wenn dieser angeben ist, sowie - in einem vernünftigerweise durchführbaren Umfang - für die mit dem Schutzgegenstand zu verbindende Internetadresse in Form des Uniform-Resource-Identifier (URI), wie sie der Lizenzgeber angegeben hat, sofern dies geschehen ist, es sei denn, diese Internetadresse verweist nicht auf den Urhebervermerk oder die Lizenzinformationen zu dem Schutzgegenstand. Ein solcher Hinweis kann in jeder angemessenen Weise erfolgen, wobei jedoch bei einer Datenbank oder einem Sammelwerk der Hinweis zumindest an gleicher Stelle und in ebenso auffälliger Weise zu erfolgen hat wie vergleichbare Hinweise auf andere Rechtsinhaber.

Obwohl die gemäß Ziffer 3 gewährten Nutzungsrechte in umfassender Weise ausgeübt werden dürfen, findet diese Erlaubnis ihre gesetzliche Grenze in den Persönlichkeitsrechten der Urheber und ausübenden Künstler, deren berechtigte geistige und persönliche Interessen bzw. deren Ansehen oder Ruf nicht dadurch gefährdet werden dürfen, dass ein Schutzgegenstand über das gesetzlich zulässige Maß hinaus beeinträchtigt wird.

Gewährleistung. Sofern dies von den Vertragsparteien nicht anderweitig schriftlich vereinbart ist, bietet der Lizenzgeber keine Gewährleistung für die erteilten Rechte, außer für den Fall, dass Mängel arglistig verschwiegen wurden. Für Mängel anderer Art, insbesondere bei der mangelhaften Lieferung von Verkörperungen des Schutzgegenstandes, richtet sich die Gewährleistung nach der Regelung, die die Person, die Ihnen den Schutzgegenstand zur Verfügung stellt, mit Ihnen außerhalb dieser Lizenz vereinbart, oder - wenn eine solche Regelung nicht getroffen wurde - nach den gesetzlichen Vorschriften.

Haftung. Über die in Ziffer 5 genannte Gewährleistung hinaus haftet Ihnen der Lizenzgeber nur für Vorsatz und grobe Fahrlässigkeit.

Vertragsende

Dieser Lizenzvertrag und die durch ihn eingeräumten Nutzungsrechte enden automatisch bei jeder Verletzung der Vertragsbedingungen durch Sie. Für natürliche und juristische Personen, die von Ihnen eine Datenbank oder ein Sammelwerk unter diesen Lizenzbedingungen erhalten haben, gilt die Lizenz jedoch weiter, vorausgesetzt, diese natürlichen oder juristischen Personen erfüllen sämtliche Vertragsbedingungen. Die Ziffern 1, 2, 5, 6, 7 und 8 gelten bei einer Vertragsbeendigung fort.

Unter den oben genannten Bedingungen erfolgt die Lizenz auf unbegrenzte Zeit (für die Dauer des Schutzrechts). Dennoch behält sich der Lizenzgeber das Recht vor, den Schutzgegenstand unter anderen Lizenzbedingungen zu nutzen oder die eigene Weitergabe des Schutzgegenstandes jederzeit zu beenden, vorausgesetzt, dass solche Handlungen nicht dem Widerruf dieser Lizenz dienen (oder jeder anderen Lizenzierung, die auf Grundlage dieser Lizenz erfolgt ist oder erfolgen muss) und diese Lizenz wirksam bleibt, bis Sie unter den oben genannten Voraussetzungen endet.

Schlussbestimmungen

Jedes Mal, wenn Sie den Schutzgegenstand vervielfältigen, verbreiten oder öffentlich wiedergeben, bietet der Lizenzgeber dem Erwerber eine Lizenz für den Schutzgegenstand unter denselben Vertragsbedingungen an, unter denen er Ihnen die Lizenz eingeräumt hat.

Sollte eine Bestimmung dieses Lizenzvertrages unwirksam sein, so wird die Wirksamkeit der übrigen Lizenzbestimmungen dadurch nicht berührt, und an die Stelle der unwirksamen Bestimmung tritt eine Ersatzregelung, die dem mit der unwirksamen Bestimmung angestrebten Zweck am nächsten kommt.

Nichts soll dahingehend ausgelegt werden, dass auf eine Bestimmung dieses Lizenzvertrages verzichtet oder einer Vertragsverletzung zugestimmt wird, so lange ein solcher Verzicht oder eine solche Zustimmung nicht schriftlich vorliegen und von der verzichtenden oder zustimmenden Vertragspartei unterschrieben sind.

Dieser Lizenzvertrag stellt die vollständige Vereinbarung zwischen den Vertragsparteien hinsichtlich des Schutzgegenstandes dar. Es gibt keine weiteren ergänzenden Vereinbarungen oder mündlichen Abreden im Hinblick auf den Schutzgegenstand. Der Lizenzgeber ist an keine zusätzlichen Abreden gebunden, die aus irgendeiner Absprache

mit Ihnen entstehen könnten. Der Lizenzvertrag kann nicht ohne eine übereinstimmende schriftliche Vereinbarung zwischen dem Lizenzgeber und Ihnen abgeändert werden.

Auf diesen Lizenzvertrag findet das Recht der Bundesrepublik Deutschland Anwendung.

CREATIVE COMMONS IST KEINE VERTRAGSPARTEI DIESES LIZENZVERTRAGES UND ÜBERNIMMT KEINERLEI GEWÄHRLEISTUNG FÜR DAS WERK. CREATIVE COMMONS IST IHNEN ODER DRITTEN GEGENÜBER NICHT HAFTBAR FÜR SCHÄDEN JEDWEDER ART. UNGEACHTET DER VORSTEHENDEN ZWEI (2) SÄTZE HAT CREATIVE COMMONS ALLE RECHTE UND PFLICHTEN EINES LIZENZGEBERS, WENN SICH CREATIVE COMMONS AUSDRÜCKLICH ALS LIZENZGEBER BEZEICHNET.

AUSSER FÜR DEN BESCHRÄNKTEN ZWECK EINES HINWEISES AN DIE ÖFFENTLICHKEIT, DASS DAS WERK UNTER DER CCPL LIZENZIERT WIRD, DARF KEINE VERTRAGSPARTEI DIE MARKE "CREATIVE COMMONS" ODER EINE ÄHNLICHE MARKE ODER DAS LOGO VON CREATIVE COMMONS OHNE VORHERIGE GENEHMIGUNG VON CREATIVE COMMONS NUTZEN. JEDE GESTATTETE NUTZUNG HAT IN ÜBEREINSTIMMUNG MIT DEN JEWEILS GÜLTIGEN NUTZUNGSBEDINGUNGEN FÜR MARKEN VON CREATIVE COMMONS ZU ERFOLGEN, WIE SIE AUF DER WEBSITE ODER IN ANDERER WEISE AUF ANFRAGE VON ZEIT ZU ZEIT ZUGÄNGLICH GEMACHT WERDEN.

CREATIVE COMMONS KANN UNTER http://creativecommons.org KONTAKTIERT WERDEN.

8.6 Creative Commons Lizenz (by-nc-nd)

8.5 Creative Commons Lizenz (by-nd) 8.7 GNU Public License (deutsche Übersetzung)

8.7 GNU Public License (deutsche Übersetzung)

8.6 Creative Commons Lizenz (by-nc-nd) 8.8. GNU Free Documentation License

Hinweis: am 29. Juni 2007 wurden die GPL und die LGPL in der Version 3 veröffentlicht. In dieser Version sind insbesondere Regelungen zu DRM (Digital Rights Management) und Software-Patenten hinzugekommen. Eine inoffizielle deutsche Übersetzung ist unter http://www.gnu.de/documents/gpl-3.0.de.html verfügbar. Es ist davon auszugehen, dass in den kommenden Monaten viele Projekte die GPLv3 einsetzen werden. Der englische Originaltext der GPLv3 kann unter http://www.gnu.org/licenses/gpl-3.0.html eingesehen werden. Im Folgenden finden Sie die deutsche Übersetzung der GPLv2.

Deutsche Übersetzung der GNU General Public License

Erstellt im Auftrag der S.u.S.E. GmbH [suse@suse.de] von Katja Lachmann Übersetzungen [na194@fim.uni-erlangen.de], überarbeitet von Peter Gerwinski [peter.gerwinski@uni-essen.de] (31. Oktober 1996)

Diese Übersetzung wird mit der Absicht angeboten, das Verständnis der GNU General Public License (GNU-GPL) zu erleichtern. Es handelt sich jedoch nicht um eine offizielle oder im rechtlichen Sinne anerkannte Übersetzung.

Die Free Software Foundation (FSF) ist nicht der Herausgeber dieser Übersetzung, und sie hat diese Übersetzung auch nicht als rechtskräftigen Ersatz für die Original-GNU-GPL anerkannt. Da die Übersetzung nicht sorgfältig von Anwälten überprüft wurde, können die Übersetzer nicht garantieren, daß die Übersetzung die rechtlichen Aussagen der GNU-GPL exakt wiedergibt. Wenn Sie sichergehen wollen, daß von Ihnen geplante Aktivitäten im Sinne der GNU-GPL gestattet sind, halten Sie sich bitte an die englischsprachige Originalversion.

Die Free Software Foundation möchte Sie darum bitten, diese Übersetzung nicht als offizielle Lizenzbedingungen für von Ihnen geschriebene Programme zu verwenden. Bitte benutzen Sie hierfür statt dessen die von der Free Software Foundation herausgegebene englischsprachige Originalversion.

This is a translation of the GNU General Public License into German. This translation is distributed in the hope that it will facilitate understanding, but it is not an official or legally approved translation.

The Free Software Foundation is not the publisher of this translation and has not approved it as a legal substitute for the authentic GNU General Public License. The translation has not been reviewed carefully by lawyers, and therefore the translator cannot be sure that it exactly represents the legal meaning of the GNU General Public License. If you wish to be sure whether your planned activities are permitted by the GNU General Public License, please refer to the authentic English version.

The Free Software Foundation strongly urges you not to use this translation as the official distribution terms for your programs; instead, please use the authentic English version published by the Free Software Foundation.

GNU General Public License

Deutsche Übersetzung der Version 2, Juni 1991

675 Mass Ave, Cambridge, MA 02139, USA

Diese Übersetzung ist kein rechtskräftiger Ersatz für die englischsprachige Originalversion!

Vorwort

Die meisten Softwarelizenzen sind daraufhin entworfen worden, Ihnen die Freiheit zu nehmen, die Software weiterzugeben und zu verändern. Im Gegensatz dazu soll Ihnen die GNU General Public License, die allgemeine öffentliche GNU-Lizenz, ebendiese Freiheit garantieren. Sie soll sicherstellen, daß die Software für alle Benutzer frei ist. Diese Lizenz gilt für den Großteil der von der Free Software Foundation herausgegebenen Software und für alle anderen Programme, deren Autoren Ihr Werk dieser Lizenz unterstellt haben. Auch Sie können diese Möglichkeit der Lizenzierung für Ihre Programme anwenden. (Ein anderer Teil der Software der Free Software Foundation unterliegt statt dessen der GNU Lesser General Public License, der allgemeinen öffentlichen GNU-Lizenz für Bibliotheken.)

Die Bezeichnung „freie“ Software bezieht sich auf Freiheit, nicht auf den Preis. Unsere Lizenzen sollen Ihnen die Freiheit garantieren, Kopien freier Software zu verbreiten (und etwas für diesen Service zu berechnen, wenn Sie möchten), die Möglichkeit, die Software im Quelltext zu erhalten oder den Quelltext auf Wunsch zu bekommen. Die Lizenzen sollen garantieren, daß Sie die Software ändern oder Teile davon in neuen freien Programmen verwenden dürfen - und daß Sie wissen, daß Sie dies alles tun dürfen.

Um Ihre Rechte zu schützen, müssen wir Einschränkungen machen, die es jedem verbieten, Ihnen diese Rechte zu verweigern oder Sie aufzufordern, auf diese Rechte zu verzichten. Aus diesen Einschränkungen folgen bestimmte Verantwortlichkeiten für Sie, wenn Sie Kopien der Software verbreiten oder sie verändern.

Beispielsweise müssen Sie den Empfängern alle Rechte gewähren, die Sie selbst haben, wenn Sie - kostenlos oder gegen Bezahlung - Kopien eines solchen Programms verbreiten. Sie müssen sicherstellen, daß auch Sie den Quelltext erhalten beziehungsweise erhalten können. Und sie müssen ihnen diese Bedingungen zeigen, damit die Benutzer die Rechte kennen.

Wir schützen Ihre Rechte in zwei Schritten: (1) Wir stellen die Software unter ein Urheberrecht (Copyright), und (2) wir bieten Ihnen diese Lizenz an, die Ihnen das Recht gibt, die Software zu vervielfältigen, zu verbreiten und/oder zu verändern.

Um die Autoren und uns zu schützen, wollen wir darüberhinaus sicherstellen, daß jeder erfährt, daß für diese Freie Software keinerlei Garantie besteht. Wenn die Software von jemand anderem modifiziert und weitergegeben wird, möchten wir, daß die Empfänger

wissen, daß sie nicht das Original erhalten haben, damit von anderen verursachte Probleme nicht den Ruf des ursprünglichen Autors schädigen.

Schließlich und endlich ist jedes freie Programm permanent durch Software-Patente bedroht. Wir möchten die Gefahr ausschließen, daß Distributoren eines freien Programms individuell Patente lizensieren - mit dem Ergebnis, daß das Programm proprietär würde. Um dies zu verhindern, haben wir klargestellt, daß jedes Patent entweder für freie Benutzung durch jedermann lizenziert werden muß oder überhaupt nicht lizenziert werden darf.

Es folgen die genauen Bedingungen für die Vervielfältigung, Verbreitung und Bearbeitung:

Bedingungen für die Vervielfältigung, Verbreitung und Bearbeitung

Paragraph 0. Diese Lizenz gilt für jedes Programm und jedes andere Werk, in dem ein entsprechender Vermerk des Copyright-Inhabers darauf hinweist, daß das Werk unter den Bestimmungen dieser General Public License verbreitet werden darf. Im Folgenden wird jedes derartige Programm oder Werk als „das Programm“ bezeichnet; die Formulierung „auf dem Programm basierendes Werk“ bezeichnet das Programm sowie jegliche Bearbeitung des Programms im urheberrechtlichen Sinne, also ein Werk, welches das Programm, auch auszugsweise, sei es unverändert oder verändert und/oder in eine andere Sprache übersetzt, enthält. (im Folgenden wird die Übersetzung ohne Einschränkung als „Bearbeitung“ eingestuft.) Jeder Lizenznehmer wird im folgenden als „Sie“ angesprochen.

Andere Handlungen als Vervielfältigung, Verbreitung und Bearbeitung werden von dieser Lizenz nicht berührt; sie fallen nicht in Ihren Anwendungsbereich. Der Vorgang der Ausführung des Programms wird nicht eingeschränkt, und die Ausgaben des Programms unterliegen dieser Lizenz nur, wenn der Inhalt ein auf dem Programm basierendes Werk darstellt (unabhängig davon, daß die Ausgabe durch die Ausführung des Programms erfolgte). Ob dies zutrifft, hängt von den Funktionen des Programms ab.

Paragraph 1. Sie dürfen auf beliebigen Medien unveränderte Kopien des Quelltextes des Programms, wie Sie ihn erhalten haben, anfertigen und verbreiten. Voraussetzung hierfür ist, daß Sie mit jeder Kopie einen entsprechenden Copyright-Vermerk sowie einen Haftungsausschluß veröffentlichen, alle Vermerke, die sich auf diese Lizenz und das Fehlen einer Garantie beziehen, unverändert lassen und desweiteren allen anderen Empfängern des Programms zusammen mit dem Programm eine Kopie dieser Lizenz zukommen lassen.

Sie dürfen für den eigentlichen Kopiervorgang eine Gebühr verlangen. Wenn Sie es wünschen, dürfen Sie auch gegen Entgelt eine Garantie für das Programm anbieten.

Paragraph 2. Sie dürfen Ihre Kopie(n) des Programms oder eines Teils davon verändern, wodurch ein auf dem Programm basierendes Werk entsteht; Sie dürfen derartige Bearbeitungen unter den Bestimmungen von Paragraph 1 vervielfältigen und verbreiten, vorausgesetzt, daß zusätzlich alle folgenden Bedingungen erfüllt werden:

(a)

Sie müssen die veränderten Dateien mit einem auffälligen Vermerk versehen, der auf die von Ihnen vorgenommene Modifizierung und das Datum jeder Änderung hinweist.

(b)

Sie müssen dafür sorgen, daß jede von Ihnen verbreitete oder veröffentlichte Arbeit, die ganz oder teilweise von dem Programm oder Teilen davon abgeleitet ist, Dritten gegenüber als Ganzes unter den Bedingungen dieser Lizenz ohne Lizenzgebühren zur Verfügung gestellt wird.

(c)

Wenn das veränderte Programm normalerweise bei der Ausführung interaktiv Kommandos einliest, müssen Sie dafür sorgen, daß es, wenn es auf dem üblichsten Wege für solche interaktive Nutzung gestartet wird, eine Meldung ausgibt oder ausdruckt, die einen geeigneten Copyright-Vermerk enthält sowie einen Hinweis, daß es keine Gewährleistung gibt (oder anderenfalls, daß Sie Garantie leisten), und daß die Benutzer das Programm unter diesen Bedingungen weiter verbreiten dürfen. Auch muß der Benutzer darauf hingewiesen werden, wie er eine Kopie dieser Lizenz ansehen kann. (Ausnahme: Wenn das Programm selbst interaktiv arbeitet, aber normalerweise keine derartige Meldung ausgibt, muß Ihr auf dem Programm basierendes Werk auch keine solche Meldung ausgeben).

Diese Anforderungen betreffen das veränderte Werk als Ganzes. Wenn identifizierbare Abschnitte des Werkes nicht von dem Programm abgeleitet sind und vernünftigerweise selbst als unabhängige und eigenständige Werke betrachtet werden können, dann erstrecken sich diese Lizenz und Ihre Bedingungen nicht auf diese Abschnitte, wenn sie als eigenständige Werke verbreitet werden. Wenn Sie jedoch dieselben Abschnitte als Teil eines Ganzen verbreiten, das ein auf dem Programm basierendes Werk darstellt, dann muß die Verbreitung des Ganzen nach den Bedingungen dieser Lizenz erfolgen, deren Bedingungen für weitere Lizenznehmer somit auf die Gesamtheit ausgedehnt werden - und damit auf jeden einzelnen Teil, unabhängig vom jeweiligen Autor.

Somit ist es nicht die Absicht dieses Abschnittes, Rechte für Werke in Anspruch zu nehmen oder zu beschneiden, die komplett von Ihnen geschrieben wurden; vielmehr ist es die Absicht, die Rechte zur Kontrolle der Verbreitung von Werken, die auf dem Programm basieren oder unter seiner auszugsweisen Verwendung zusammengestellt worden sind, auszuüben.

Ferner bringt ein einfaches Zusammenstellen eines anderen Werkes, das nicht auf dem Programm basiert, zusammen mit dem Programm oder einem auf dem Programm basierenden Werk auf ein- und demselben Speicher- oder Vertriebsmedium das andere Werk nicht in den Anwendungsbereich dieser Lizenz.

Paragraph 3. Sie dürfen das Programm (oder ein darauf basierendes Werk gemäß Paragraph 2) als Objectcode oder in ausführbarer Form unter den Bedingungen von Paragraph 1 und 2 vervielfältigen und verbreiten - vorausgesetzt, daß Sie außerdem eine der folgenden Leistungen erbringen:

(a)

Liefern Sie das Programm zusammen mit dem vollständigen zugehörigen maschinenlesbaren Quelltext auf einem für den Datenaustausch üblichen Medium aus, wobei die Verteilung unter den Bedingungen der Paragraphen 1 und 2 erfolgen muß. Oder:

(b)

Liefern Sie das Programm zusammen mit einem mindestens drei Jahre lang gültigen schriftlichen Angebot aus, jedem Dritten eine vollständige maschinenlesbare Kopie des Quelltextes zur Verfügung zu stellen - zu nicht höheren Kosten als denen, die durch den physikalischen Kopiervorgang anfallen -, wobei der Quelltext unter den Bedingungen der Paragraphen 1 und 2 auf einem für den Datenaustausch üblichen Medium weitergegeben wird. Oder:

(c)

Liefern Sie das Programm zusammen mit dem schriftlichen Angebot der Zurverfügungstellung des Quelltextes aus, das Sie selbst erhalten haben. (Diese Alternative ist nur für nicht-kommerzielle Verbreitung zulässig und nur, wenn Sie das Programm als Objectcode oder in ausführbarer Form mit einem entsprechenden Angebot gemäß Absatz b erhalten haben.)

Unter dem Quelltext eines Werkes wird diejenige Form des Werkes verstanden, die für Bearbeitungen vorzugsweise verwendet wird. Für ein ausführbares Programm bedeutet „der komplette Quelltext“ der Quelltext aller im Programm enthaltenen Module einschließlich aller zugehörigen Modulschnittstellen-Definitionsdateien sowie der zur Compilation und Installation verwendeten Scripts. Als besondere Ausnahme jedoch braucht der verteilte Quelltext nichts von dem zu enthalten, was üblicherweise (entweder als Quelltext oder in binärer Form) zusammen mit den Hauptkomponenten des Betriebssystems (Kernel, Compiler usw.) geliefert wird, unter dem das Programm läuft - es sei denn, diese Komponente selbst gehört zum ausführbaren Programm.

Wenn die Verbreitung eines ausführbaren Programms oder des Objectcodes dadurch erfolgt, daß der Kopierzugriff auf eine dafür vorgesehene Stelle gewährt wird, so gilt die Gewährung eines gleichwertigen Zugriffs auf den Quelltext als Verbreitung des Quelltextes, auch wenn Dritte nicht dazu gezwungen sind, den Quelltext zusammen mit dem Objectcode zu kopieren.

Paragraph 4. Sie dürfen das Programm nicht vervielfältigen, verändern, weiter lizenzieren oder verbreiten, sofern es nicht durch diese Lizenz ausdrücklich gestattet ist. Jeder anderweitige Versuch der Vervielfältigung, Modifizierung, Weiterlizenzierung und Verbreitung ist nichtig und beendet automatisch Ihre Rechte unter dieser Lizenz. Jedoch werden die Lizenzen Dritter, die von Ihnen Kopien oder Rechte unter dieser Lizenz erhalten haben, nicht beendet, solange diese die Lizenz voll anerkennen und befolgen.

Paragraph 5. Sie sind nicht verpflichtet, diese Lizenz anzunehmen, da Sie sie nicht unterzeichnet haben. Jedoch gibt Ihnen nichts anderes die Erlaubnis, das Programm oder von ihm abgeleitete Werke zu verändern oder zu verbreiten. Diese Handlungen sind gesetzlich verboten, wenn Sie diese Lizenz nicht anerkennen. Indem Sie das Programm (oder ein darauf basierendes Werk) verändern oder verbreiten, erklären Sie Ihr

Einverständnis mit dieser Lizenz und mit allen Ihren Bedingungen bezüglich der Vervielfältigung, Verbreitung und Veränderung des Programms oder eines darauf basierenden Werkes.

Paragraph 6. Jedesmal, wenn Sie das Programm (oder ein auf dem Programm basierendes Werk) weitergeben, erhält der Empfänger automatisch vom ursprünglichen Lizenzgeber die Lizenz, das Programm entsprechend den hier festgelegten Bestimmungen zu vervielfältigen, zu verbreiten und zu verändern. Sie dürfen keine weiteren Einschränkungen der Durchsetzung der hierin zugestandenen Rechte des Empfängers vornehmen. Sie sind nicht dafür verantwortlich, die Einhaltung dieser Lizenz durch Dritte durchzusetzen.

Paragraph 7. Sollten Ihnen infolge eines Gerichtsurteils, des Vorwurfs einer Patentverletzung oder aus einem anderen Grunde (nicht auf Patentfragen begrenzt) Bedingungen (durch Gerichtsbeschluß, Vergleich oder anderweitig) auferlegt werden, die den Bedingungen dieser Lizenz widersprechen, so befreien Sie diese Umstände nicht von den Bestimmungen dieser Lizenz. Wenn es Ihnen nicht möglich ist, das Programm unter gleichzeitiger Beachtung der Bedingungen in dieser Lizenz und Ihrer anderweitigen Verpflichtungen zu verbreiten, dann dürfen Sie als Folge das Programm überhaupt nicht verbreiten. Wenn zum Beispiel ein Patent nicht die gebührenfreie Weiterverbreitung des Programms durch diejenigen erlaubt, die das Programm direkt oder indirekt von Ihnen erhalten haben, dann besteht der einzige Weg, sowohl das Patentrecht als auch diese Lizenz zu befolgen, darin, ganz auf die Verbreitung des Programms zu verzichten.

Sollte sich ein Teil dieses Paragraphen als ungültig oder unter bestimmten Umständen nicht durchsetzbar erweisen, so soll dieser Paragraph seinem Sinne nach angewandt werden; im übrigen soll dieser Paragraph als Ganzes gelten.

Zweck dieses Paragraphen ist nicht, Sie dazu zu bringen, irgendwelche Patente oder andere Eigentumsansprüche zu verletzen oder die Gültigkeit solcher Ansprüche zu bestreiten; dieser Paragraph hat einzig den Zweck, die Integrität des Verbreitungssystems der freien Software zu schützen, das durch die Praxis öffentlicher Lizenzen verwirklicht wird. Viele Leute haben großzügige Beiträge zu dem großen Angebot der mit diesem System verbreiteten Software im Vertrauen auf die konsistente Anwendung dieses Systems geleistet; es liegt am Autor/Geber, zu entscheiden, ob er die Software mittels irgendeines anderen Systems verbreiten will; ein Lizenznehmer hat auf diese Entscheidung keinen Einfluß.

Dieser Paragraph ist dazu gedacht, deutlich klarzustellen, was als Konsequenz aus dem Rest dieser Lizenz betrachtet wird.

Paragraph 8. Wenn die Verbreitung und/oder die Benutzung des Programms in bestimmten Staaten entweder durch Patente oder durch urheberrechtlich geschützte Schnittstellen eingeschränkt ist, kann der Urheberrechtsinhaber, der das Programm unter diese Lizenz gestellt hat, eine explizite geographische Begrenzung der Verbreitung angeben, in der diese Staaten ausgeschlossen werden, so daß die Verbreitung nur innerhalb und zwischen den Staaten erlaubt ist, die nicht ausgeschlossen sind. In einem solchen Fall beinhaltet diese Lizenz die Beschränkung, als wäre sie in diesem Text niedergeschrieben.

Paragraph 9. Die Free Software Foundation kann von Zeit zu Zeit überarbeitete und/oder neue Versionen der General Public License veröffentlichen. Solche neuen Versionen werden vom Grundprinzip her der gegenwärtigen entsprechen, können aber im Detail abweichen, um neuen Problemen und Anforderungen gerecht zu werden.

Jede Version dieser Lizenz hat eine eindeutige Versionsnummer. Wenn in einem Programm angegeben wird, daß es dieser Lizenz in einer bestimmten Versionsnummer oder „jeder späteren Version“ („any later version“ unterliegt, so haben Sie die Wahl, entweder den Bestimmungen der genannten Version zu folgen oder denen jeder beliebigen späteren Version, die von der Free Software Foundation veröffentlicht wurde. Wenn das Programm keine Versionsnummer angibt, können Sie eine beliebige Version wählen, die je von der Free Software Foundation veröffentlicht wurde.

Paragraph 10. Wenn Sie den Wunsch haben, Teile des Programms in anderen freien Programmen zu verwenden, deren Bedingungen für die Verbreitung anders sind, schreiben Sie an den Autor, um ihn um die Erlaubnis zu bitten. Für Software, die unter dem Copyright der Free Software Foundation steht, schreiben Sie an die Free Software Foundation; wir machen zu diesem Zweck gelegentlich Ausnahmen. Unsere Entscheidung wird von den beiden Zielen geleitet werden, zum einen den freien Status aller von unserer freien Software abgeleiteten Werke zu erhalten und zum anderen das gemeinschaftliche Nutzen und Wiederverwenden von Software im allgemeinen zu fördern.

Keine Gewährleistung

Paragraph 11. Da das Programm ohne jegliche Kosten lizenziert wird, besteht keinerlei Gewährleistung für das Programm, soweit dies gesetzlich zulässig ist. Sofern nicht anderweitig schriftlich bestätigt, stellen die Copyright-Inhaber und/oder Dritte das Programm so zur Verfügung, „wie es ist“ ohne irgendeine Gewährleistung, weder ausdrücklich noch implizit, einschließlich - aber nicht begrenzt auf - Marktreife oder Verwendbarkeit für einen bestimmten Zweck. Das volle Risiko bezüglich Qualität und Leistungsfähigkeit des Programms liegt bei Ihnen. Sollte sich das Programm als fehlerhaft herausstellen, liegen die Kosten für notwendigen Service, Reparatur oder Korrektur bei Ihnen.

Paragraph 12. In keinem Fall, außer wenn durch geltendes Recht gefordert oder schriftlich zugesichert, ist irgendein Copyright-Inhaber oder irgendein Dritter, der das Programm wie oben erlaubt modifiziert oder verbreitet hat, Ihnen gegenüber für irgendwelche Schäden haftbar, einschließlich jeglicher allgemeiner oder spezieller Schäden, Schäden durch Seiteneffekte (Nebenwirkungen) oder Folgeschäden, die aus der Benutzung des Programms oder der Unbenutzbarkeit des Programms folgen (einschließlich - aber nicht beschränkt auf - Datenverluste, fehlerhafte Verarbeitung von Daten, Verluste, die von Ihnen oder anderen getragen werden müssen, oder dem Unvermögen des Programms, mit irgendeinem anderen Programm zusammenzuarbeiten), selbst wenn ein Copyright-Inhaber oder Dritter über die Möglichkeit solcher Schäden unterrichtet worden war.

Ende der Bedingungen

Anhang: Wie Sie diese Bedingungen auf Ihre neuen Programme anwendbar machen

Wenn Sie ein neues Programm entwickeln und wollen, daß es von größtmöglichem Nutzen für die Allgemeinheit ist, dann erreichen Sie das am besten, indem Sie es zu freier Software machen, die jeder unter diesen Bestimmungen weiterverbreiten und verändern kann.

Um dies zu erreichen, fügen Sie die folgenden Anmerkungen zu Ihrem Programm hinzu. Am sichersten ist es, sie an den Anfang einer jeden Quelldatei zu stellen, um den Gewährleistungsausschluß möglichst deutlich darzustellen; außerdem sollte jede Datei mindestens eine „Copyright"-Zeile besitzen sowie einen kurzen Hinweis darauf, wo die vollständige Lizenz gefunden werden kann.

[eine Zeile mit dem Programmnamen und einer kurzen Beschreibung] Copyright (C) 19[yy] [Name des Autors]

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Auf Deutsch: [eine Zeile mit dem Programmnamen und einer kurzen Beschreibung] Copyright (C) 19[jj] [Name des Autors]

Dieses Programm ist Freie Software. Sie können es unter den Bedingungen der GNU General Public License, wie von der Free Software Foundation herausgegeben, weitergeben und/oder modifizieren, entweder unter Version 2 der Lizenz oder (wenn Sie es wünschen) jeder späteren Version.

Die Veröffentlichung dieses Programms erfolgt in der Hoffnung, daß es Ihnen von Nutzen sein wird, aber OHNE JEDE GEWÄHRLEISTUNG - sogar ohne die implizite Gewährleistung der MARKTREIFE oder der EIGNUNG FÜR EINEN BESTIMMTEN ZWECK. Details finden Sie in der GNU General Public License.

Sie sollten eine Kopie der GNU General Public License zusammen mit diesem Programm erhalten haben. Falls nicht, schreiben Sie an die Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Fügen Sie auch einen kurzen Hinweis hinzu, wie Sie elektronisch und per Brief erreichbar sind.

Wenn Ihr Programm interaktiv ist, sorgen Sie dafür, daß es nach dem Start einen kurzen Vermerk ausgibt:

Gnomovision version 69, Copyright (C) 19[yy] [Name des Autors] Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

Auf Deutsch:

Gnomovision Version 69, Copyright (C) 19[jj] [Name des Autors] Für Gnomovision besteht KEINERLEI GARANTIE; geben Sie `show w' für Details ein. Gnomovision ist freie Software, die Sie unter bestimmten Bedingungen weitergeben dürfen; geben Sie `show c' für Details ein.

Die hypothetischen Kommandos `show w' und `show c' sollten die entsprechenden Teile der GNU-GPL anzeigen. Natürlich können die von Ihnen verwendeten Kommandos anders heißen als `show w' und `show c'; es könnten auch Mausklicks oder Menüpunkte sein - was immer am besten in Ihr Programm paßt.

Soweit vorhanden, sollten Sie auch Ihren Arbeitgeber (wenn Sie als Programmierer arbeiten) oder Ihre Schule einen Copyright-Verzicht für das Programm unterschreiben lassen. Hier ein Beispiel; ändern Sie bitte die Namen:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program „Gnomovision“ (which makes passes at compilers) written by James Hacker.

[Unterschrift von Ty Coon], 1 April 1989 Ty Coon, President of Vice

Auf Deutsch:

Die Yoyodyne GmbH erhebt keinerlei urheberrechtlichen Anspruch auf das Programm „Gnomovision“ (einem Schrittmacher für Compiler), geschrieben von James Hacker.

[Unterschrift von Ty Coon], 1. April 1989 Ty Coon, Vizepräsident

Diese General Public License gestattet nicht die Einbindung des Programms in proprietäre Programme. Ist Ihr Programm eine Funktionsbibliothek, so kann es sinnvoller sein, das Linken proprietärer Programme mit dieser Bibliothek zu gestatten. Wenn Sie dies tun wollen, sollten Sie die GNU Library General Public License anstelle dieser Lizenz verwenden.

8.7 GNU Public License (deutsche Übersetzung)

8.6 Creative Commons Lizenz (by-nc-nd) 8.8. GNU Free Documentation License

8.8. GNU Free Documentation License

8.7 GNU Public License (deutsche Übersetzung) Kapitel 9. Historische Dokumente

Version 1.2, November 2002

PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "quote" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship

could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any

other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

State on the Title page the name of the publisher of the Modified Version, as the publisher.

Preserve all the copyright notices of the Document.

Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

Include an unaltered copy of this License.

Preserve the section Entitled “History”, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.

Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.

Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties-for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or

publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

8.8. GNU Free Documentation License

8.7 GNU Public License (deutsche Übersetzung) Kapitel 9. Historische Dokumente

Kapitel 9. Historische Dokumente

8.8. GNU Free Documentation License | Kapitel 10. Impressum

Inhaltsverzeichnis

BABYL OPTIONS: Version: 5 Labels: Note: This is the header of an rmail file. Note: If you are seeing it in rmail, Note: it means the file has no messages in it. 0, unseen,, *** EOOH *** From: ast@cs.vu.nl (Andy Tanenbaum) Newsgroups: comp.os.minix Subject: LINUX is obsolete Date: 29 Jan 92 12:12:50 GMT Organization: Fac. Wiskunde & Informatica, Vrije Universiteit, Amsterdam I was in the U.S. for a couple of weeks, so I haven't commented much on LINUX (not that I would have said much had I been around), but for what it is worth, I have a couple of comments now. As most of you know, for me MINIX is a hobby, something that I do in the evening when I get bored writing books and there are no major wars, revolutions, or senate hearings being televised live on CNN. My real job is a professor and researcher in the area of operating systems. As a result of my occupation, I think I know a bit about where operating are going in the next decade or so. Two aspects stand out: 1. MICROKERNEL VS MONOLITHIC SYSTEM Most older operating systems are monolithic, that is, the whole operating system is a single a.out file that runs in 'kernel mode.' This binary contains the process management, memory management, file system and the rest. Examples of such systems are UNIX, MS-DOS, VMS, MVS, OS/360, MULTICS, and many more. The alternative is a microkernel-based system, in which most of the OS runs as separate processes, mostly outside the kernel. They communicate by message passing. The kernel's job is to handle the message passing, interrupt handling, low-level process management, and possibly the I/O. Examples of this design are the RC4000, Amoeba, Chorus, Mach, and the not-yet-released Windows/NT. While I could go into a long story here about the relative merits of the two designs, suffice it to say that among the people who actually design operating systems, the debate is essentially over. Microkernels have won. The only real argument for monolithic systems was performance, and there is now enough evidence showing that microkernel systems can be just as fast as monolithic systems (e.g., Rick Rashid has published papers comparing Mach 3.0 to monolithic

systems) that it is now all over but the shoutin`. MINIX is a microkernel-based system. The file system and memory management are separate processes, running outside the kernel. The I/O drivers are also separate processes (in the kernel, but only because the brain-dead nature of the Intel CPUs makes that difficult to do otherwise). LINUX is a monolithic style system. This is a giant step back into the 1970s. That is like taking an existing, working C program and rewriting it in BASIC. To me, writing a monolithic system in 1991 is a truly poor idea. 2. PORTABILITY Once upon a time there was the 4004 CPU. When it grew up it became an 8008. Then it underwent plastic surgery and became the 8080. It begat the 8086, which begat the 8088, which begat the 80286, which begat the 80386, which begat the 80486, and so on unto the N-th generation. In the meantime, RISC chips happened, and some of them are running at over 100 MIPS. Speeds of 200 MIPS and more are likely in the coming years. These things are not going to suddenly vanish. What is going to happen is that they will gradually take over from the 80x86 line. They will run old MS-DOS programs by interpreting the 80386 in software. (I even wrote my own IBM PC simulator in C, which you can get by FTP from ftp.cs.vu.nl = 192.31.231.42 in dir minix/simulator.) I think it is a gross error to design an OS for any specific architecture, since that is not going to be around all that long. MINIX was designed to be reasonably portable, and has been ported from the Intel line to the 680x0 (Atari, Amiga, Macintosh), SPARC, and NS32016. LINUX is tied fairly closely to the 80x86. Not the way to go. Don`t get me wrong, I am not unhappy with LINUX. It will get all the people who want to turn MINIX in BSD UNIX off my back. But in all honesty, I would suggest that people who want a **MODERN** "free" OS look around for a microkernel-based, portable OS, like maybe GNU or something like that. Andy Tanenbaum (ast@cs.vu.nl) P.S. Just as a random aside, Amoeba has a UNIX emulator (running in user space), but it is far from complete. If there are any people who would like to work on that, please let me know. To run Amoeba you need a few 386s, one of which needs 16M, and all of which need the WD Ethernet card. 0, unseen,, *** EOOH *** From: meggin@epas.utoronto.ca (David Megginson) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 29 Jan 92 14:12:12 GMT Organization: University of

Toronto - EPAS Nntp-Posting-Host: epas.utoronto.ca I would like to at least look at LINUX, but I cannot, since I run a 68000-based machine. In any case, it is nice having the kernel independent, since patches like the multi-threaded FS patch don't have to exist in a different version for each CPU. I second everything AST said, except that I would like to see the kernel _more_ independent from everything else. Why does the Intel architecture _not_ allow drivers to be independent programs? I also don't like the fact that the kernel, mm and fs share the same configuration files. Since they _are_ independent, they should have more of a sense of independence. David ## David Megginson meggin@epas.utoronto.ca Centre for Medieval Studies david@doe.utoronto.ca University of Toronto 39 Queen's Park Cr. E. ## 0, unseen,, *** EOOH *** From: ast@cs.vu.nl (Andy Tanenbaum) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 29 Jan 92 18:03:01 GMT Organization: Fac. Wiskunde & Informatica, Vrije Universiteit, Amsterdam In article <1992Jan29.141212.29636@epas.toronto.edu> meggin@epas.utoronto.ca (David Megginson) writes: > >Why does the >Intel architecture _not_ allow drivers to be independent programs? The drivers have to read and write the device registers in I/O space, and this cannot be done in user mode on the 286 and 386. If it were possible to do I/O in a protected way in user space, all the I/O tasks could have been user programs, like FS and MM. Andy Tanenbaum (ast@cs.vu.nl) 0, unseen,, *** EOOH *** From: gt0178a@prism.gatech.EDU (Jim Burns) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 30 Jan 92 03:39:48 GMT Organization: Georgia Institute of Technology in article <12605@star.cs.vu.nl>, ast@cs.vu.nl (Andy Tanenbaum) says: > The drivers have to read and write the device registers in I/O space, and > this cannot be done in user mode on the 286 and 386. If it were possible > to do I/O in a protected way in user space, all the I/O tasks could have > been user programs, like FS and MM. The standard way of doing that is to trap on i/o space protection violations, and emulate the i/o for the user. -- BURNS,JIM (returned student) Georgia Institute of Technology, 30178 Georgia Tech Station,

Atlanta Georgia, 30332 | Internet: gt0178a@prism.gatech.edu uucp: ...!{decvax,hplabs,ncar,purdue,rutgers}!gatech!prism!gt0178a 0, unseen,, *** EOOH *** From: joe@jshark.rn.com Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Summary: Is this for real? Date: 31 Jan 92 12:55:21 GMT Organization: a blip in entropy In article <12605@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes: >In article <1992Jan29.141212.29636@epas.toronto.edu> meggin@epas.utoronto.ca (David Megginson) writes: >> >>Why does the >>Intel architecture _not_ allow drivers to be independent programs? > >The drivers have to read and write the device registers in I/O space, and >this cannot be done in user mode on the 286 and 386. If it were possible >to do I/O in a protected way in user space, [[We must be talking about protected mode]] *THIS IS UNTRUE* The Intel architecture supports independent tasks, each of which can be given a "i/o privilege level". The convenient approach, used by iRMX(?), is to "build" a load image ("root" device driver, kernel, MM and FS). Once booted, these could be replaced by loadable tasks from disc (or network...) and given a suitable privilege level. The '386 additionally allows each task to have an "i/o permissions bitmap" which specifies exactly which ports can be used. (See "80386 Programmers Reference Manual", chapter 8) > all the I/O tasks could have >been user programs, like FS and MM. Do you really mean "user programs" and not "separate tasks" ?? Separate tasks, possibly privileged, I'll agree with. User level programs may be ok for teaching operating system principles, or on toy computers :-) But a "production" system? Not on my machines! >Andy Tanenbaum (ast@cs.vu.nl) joe. -- joe@jshark.rn.com uunet!nstar!jshark!joe 0, unseen,, *** EOOH *** From: drew@anchor.cs.colorado.edu (Drew Eckhardt) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 2 Feb 92 12:17:44 GMT Organization: University of Colorado, Boulder Nntp-Posting-Host: anchor.cs.colorado.edu In article <12605@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes: >In article <1992Jan29.141212.29636@epas.toronto.edu> meggin@epas.utoronto.ca (David Megginson) writes: >> >>Why does the >>Intel architecture _not_ allow drivers to be independent programs? > >The drivers have to read and write the device registers in

I/O space, and >this cannot be done in user mode on the 286 and 386. If it were possible >to do I/O in a protected way in user space, all the I/O tasks could have >been user programs, like FS and MM. > >Andy Tanenbaum (ast@cs.vu.nl) Every 386 TSS has an iopermission bitmap. If the CPL is of a lower priveledge level than IOPL, the io permissions bitmap is consulted, allowing protection on a port by port basis. 0, unseen,, *** EOOH *** From: jonathan@ukmug.uk.mugnet.org (Jonathan Allen) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 3 Feb 92 07:43:00 GMT Organization: MUGNET UK Backbone (UKMUG) In article <12605@star.cs.vu.nl>, ast@cs.vu.nl (Andy Tanenbaum) wrote: > In article <1992Jan29.141212.29636@epas.toronto.edu> meggin@epas.utoronto.ca (David Megginson) writes: >> >>Why does the >>Intel architecture _not_ allow drivers to be independent programs? > > The drivers have to read and write the device registers in I/O space, and > this cannot be done in user mode on the 286 and 386. If it were possible > to do I/O in a protected way in user space, all the I/O tasks could have > been user programs, like FS and MM. Surely this could have been done by a minute task just to read/write a given port address in one message ? The security could have been checked like everything else using the process table ... Sure it would not have been at all efficient, but would have given the independance at a price. Jonathan 0, unseen,, *** EOOH *** From: ts@cup.portal.com (Tim W Smith) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 7 Feb 92 01:30:59 GMT Organization: The Portal System (TM) Andy Tanenbaum (ast@cs.vu.nl) writes: > The drivers have to read and write the device registers in I/O space, and > this cannot be done in user mode on the 286 and 386. If it were possible > to do I/O in a protected way in user space, all the I/O tasks could have > been user programs, like FS and MM. On the 386, you could run the drivers in V86 mode, which sort of counts as user mode and allows access to I/O registers if the kernel sets things up to allow this. Tim Smith 0, unseen,, *** EOOH *** From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 29 Jan 92 23:14:26 GMT Organization: University of Helsinki Well, with a subject like this, I'm afraid I'll have to reply. Apologies to minix-

users who have heard enough about linux anyway. I'd like to be able to just "ignore the bait", but ... Time for some serious flamefesting! In article <12595@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes: > >I was in the U.S. for a couple of weeks, so I haven't commented much on >LINUX (not that I would have said much had I been around), but for what >it is worth, I have a couple of comments now. > >As most of you know, for me MINIX is a hobby, something that I do in the >evening when I get bored writing books and there are no major wars, >revolutions, or senate hearings being televised live on CNN. My real >job is a professor and researcher in the area of operating systems. You use this as an excuse for the limitations of minix? Sorry, but you loose: I've got more excuses than you have, and linux still beats the pants of minix in almost all areas. Not to mention the fact that most of the good code for PC minix seems to have been written by Bruce Evans. Re 1: you doing minix as a hobby - look at who makes money off minix, and who gives linux out for free. Then talk about hobbies. Make minix freely available, and one of my biggest gripes with it will disappear. Linux has very much been a hobby (but a serious one: the best type) for me: I get no money for it, and it's not even part of any of my studies in the university. I've done it all on my own time, and on my own machine. Re 2: your job is being a professor and researcher: That's one hell of a good excuse for some of the brain-damages of minix. I can only hope (and assume) that Amoeba doesn't suck like minix does. >1. MICROKERNEL VS MONOLITHIC SYSTEM True, linux is monolithic, and I agree that microkernels are nicer. With a less argumentative subject, I'd probably have agreed with most of what you said. From a theoretical (and aesthetical) standpoint linux looses. If the GNU kernel had been ready last spring, I'd not have bothered to even start my project: the fact is that it wasn't and still isn't. Linux wins heavily on points of being available now. > MINIX is a microkernel-based system. [deleted, but not so that you > miss the point] LINUX is a monolithic style system. If this was the only criterion for the "goodness" of a kernel, you'd be right. What you don't mention is that minix doesn't do the micro-kernel thing very well, and has problems with real multitasking (in the kernel). If I had made an OS that had problems with a multithreading filesystem, I wouldn't be so fast to condemn others: in fact, I'd do my damndest to make others forget

about the fiasco. [yes, I know there are multithreading hacks for minix, but they are hacks, and bruce evans tells me there are lots of race conditions] >2. PORTABILITY "Portability is for people who cannot write new programs" -me, right now (with tongue in cheek) The fact is that linux is more portable than minix. What? I hear you say. It's true - but not in the sense that ast means: I made linux as conformant to standards as I knew how (without having any POSIX standard in front of me). Porting things to linux is generally /much/ easier than porting them to minix. I agree that portability is a good thing: but only where it actually has some meaning. There is no idea in trying to make an operating system overly portable: adhering to a portable API is good enough. The very /idea/ of an operating system is to use the hardware features, and hide them behind a layer of high-level calls. That is exactly what linux does: it just uses a bigger subset of the 386 features than other kernels seem to do. Of course this makes the kernel proper unportable, but it also makes for a /much/ simpler design. An acceptable trade-off, and one that made linux possible in the first place. I also agree that linux takes the non-portability to an extreme: I got my 386 last January, and linux was partly a project to teach me about it. Many things should have been done more portably if it would have been a real project. I'm not making overly many excuses about it though: it was a design decision, and last april when I started the thing, I didn't think anybody would actually want to use it. I'm happy to report I was wrong, and as my source is freely available, anybody is free to try to port it, even though it won't be easy. Linus PS. I apologise for sometimes sounding too harsh: minix is nice enough if you have nothing else. Amoeba might be nice if you have 5-10 spare 386's lying around, but I certainly don't. I don't usually get into flames, but I'm touchy when it comes to linux :) 0, unseen,, *** EOOH *** From: ast@cs.vu.nl (Andy Tanenbaum) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 30 Jan 92 13:44:34 GMT Organization: Fac. Wiskunde & Informatica, Vrije Universiteit, Amsterdam In article <1992Jan29.231426.20469@klaava.Helsinki.FI> torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds) writes: >You use this [being a professor] as an excuse for the limitations of minix? The limitations of MINIX relate at least partly to my being a professor: An explicit design goal was to make it run on cheap hardware so

students could afford it. In particular, for years it ran on a regular 4.77 MHZ PC with no hard disk. You could do everything here including modify and recompile the system. Just for the record, as of about 1 year ago, there were two versions, one for the PC (360K diskettes) and one for the 286/386 (1.2M). The PC version was outselling the 286/386 version by 2 to 1. I don't have figures, but my guess is that the fraction of the 60 million existing PCs that are 386/486 machines as opposed to 8088/286/680x0 etc is small. Among students it is even smaller. Making software free, but only for folks with enough money to buy first class hardware is an interesting concept. Of course 5 years from now that will be different, but 5 years from now everyone will be running free GNU on their 200 MIPS, 64M SPARCstation-5. >Re 2: your job is being a professor and researcher: That's one hell of a >good excuse for some of the brain-damages of minix. I can only hope (and >assume) that Amoeba doesn't suck like minix does. Amoeba was not designed to run on an 8088 with no hard disk. >If this was the only criterion for the "goodness" of a kernel, you'd be >right. What you don't mention is that minix doesn't do the micro-kernel >thing very well, and has problems with real multitasking (in the >kernel). If I had made an OS that had problems with a multithreading >filesystem, I wouldn't be so fast to condemn others: in fact, I'd do my >damndest to make others forget about the fiasco. A multithreaded file system is only a performance hack. When there is only one job active, the normal case on a small PC, it buys you nothing and adds complexity to the code. On machines fast enough to support multiple users, you probably have enough buffer cache to insure a hit cache hit rate, in which case multithreading also buys you nothing. It is only a win when there are multiple processes actually doing real disk I/O. Whether it is worth making the system more complicated for this case is at least debatable. I still maintain the point that designing a monolithic kernel in 1991 is a fundamental error. Be thankful you are not my student. You would not get a high grade for such a design :-) >The fact is that linux is more portable than minix. What? I hear you >say. It's true - but not in the sense that ast means: I made linux as >conformant to standards as I knew how (without having any POSIX standard >in front of me). Porting things to linux is generally /much/ easier >than porting them to minix. MINIX was designed before POSIX, and is now being (slowly)

POSIXized as everyone who follows this newsgroup knows. Everyone agrees that user-level standards are a good idea. As an aside, I congratulate you for being able to write a POSIX-conformant system without having the POSIX standard in front of you. I find it difficult enough after studying the standard at great length. My point is that writing a new operating system that is closely tied to any particular piece of hardware, especially a weird one like the Intel line, is basically wrong. An OS itself should be easily portable to new hardware platforms. When OS/360 was written in assembler for the IBM 360 25 years ago, they probably could be excused. When MS-DOS was written specifically for the 8088 ten years ago, this was less than brilliant, as IBM and Microsoft now only too painfully realize. Writing a new OS only for the 386 in 1991 gets you your second 'F' for this term. But if you do real well on the final exam, you can still pass the course. Prof. Andrew S. Tanenbaum (ast@cs.vu.nl) 0, unseen,, *** EOOH *** From: feustel@netcom.COM (David Feustel) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 30 Jan 92 18:57:28 GMT Organization: DAFCO - An OS/2 Oasis ast@cs.vu.nl (Andy Tanenbaum) writes: >I still maintain the point that designing a monolithic kernel in 1991 is >a fundamental error. Be thankful you are not my student. You would not >get a high grade for such a design :-) That's ok. Einstein got lousy grades in math and physics. -- David Feustel N9MYI, 1930 Curdes Ave, Fort Wayne, IN 46805. (219)482-9631 feustel@netcom.com === NBC News: GE's Advertising And Public Relations Agency === 0, unseen,, *** EOOH *** From: meggin@epas.utoronto.ca (David Megginson) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 30 Jan 92 19:58:50 GMT Organization: University of Toronto - EPAS Nntp-Posting-Host: epas.utoronto.ca In article <1992Jan30.185728.26477feustel@netcom.COM> feustel@netcom.COM (David Feustel) writes: >ast@cs.vu.nl (Andy Tanenbaum) writes: > > >>I still maintain the point that designing a monolithic kernel in 1991 is >>a fundamental error. Be thankful you are not my student. You would not >>get a high grade for such a design :-) > >That's ok. Einstein got lousy grades in math and physics. And Dan Quayle got low grades in political science. I think that there are more Dan Quayles than Einsteins out there... ;-) David

David Megginson meggin@epas.utoronto.ca Centre for Medieval Studies david@doe.utoronto.ca University of Toronto 39 Queen's Park Cr. E. ## ######## 0, unseen,, *** EOOH *** From: feustel@netcom.COM (David Feustel) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 30 Jan 92 23:15:05 GMT Organization: DAFCO - An OS/2 Oasis meggin@epas.utoronto.ca (David Megginson) writes: >In article <1992Jan30.185728.26477feustel@netcom.COM> feustel@netcom.COM (David Feustel) writes: >>ast@cs.vu.nl (Andy Tanenbaum) writes: >> >> >>>I still maintain the point that designing a monolithic kernel in 1991 is >>>a fundamental error. Be thankful you are not my student. You would not >>>get a high grade for such a design :-) >> >>That's ok. Einstein got lousy grades in math and physics. >And Dan Quayle got low grades in political science. I think that there >are more Dan Quayles than Einsteins out there... ;-) But the Existence of Linux suggests that we may have more of an Einstein than a Quail here. -- David Feustel N9MYI, 1930 Curdes Ave, Fort Wayne, IN 46805. (219)482-9631 feustel@netcom.com === NBC News: GE's Advertising And Public Relations Agency === 0, unseen,, *** EOOH *** From: pete@ohm.york.ac.uk (-Pete French.) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 31 Jan 92 09:49:37 GMT Organization: Electronics Department, University of York, UK in article <1992Jan30.195850.7023@epas.toronto.edu>, meggin@epas.utoronto.ca (David Megginson) says: > Nntp-Posting-Host: epas.utoronto.ca > > In article <1992Jan30.185728.26477feustel@netcom.COM> feustel@netcom.COM (David Feustel) writes: >> >>That's ok. Einstein got lousy grades in math and physics. > > And Dan Quayle got low grades in political science. I think that there > are more Dan Quayles than Einsteins out there... ;-) What a horrible thought ! But on the points about microkernel v monolithic, isnt this partly an artifact of the language being used ? MINIX may well be designed as a microkernel system, but in the end you still end up with a large monolithic chunk of binary data that gets loaded in as "the OS". Isnt it written as separate programs simply because C does not support the

idea of multiple processes within a single piece of monolithic code. Is there any real difference between a microkernel written as several pieces of C and a monolithic kernel written in something like OCCAM ? I would have thought that in this case the monolithic design would be a better one than the micorkernel style since with the advantage of inbuilt language concurrency the kernel could be made even more modular than the MINIX one is. Anyone for MINOX :-) -bat. -- -Pete French. (the -bat.) / Adaptive Systems Engineering / 0, unseen,, *** EOOH *** From: peter@ferranti.com (peter da silva) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 3 Feb 92 16:22:32 GMT Organization: Xenix Support, FICC In article <1992Jan31.094937.3726@ohm.york.ac.uk> pete@ohm.york.ac.uk (-Pete French.) writes: > But on the points about microkernel v monolithic, isnt this partly an > artifact of the language being used ? I doubt it. [isn't MINIX] > written as separate programs simply because C does not support the idea > of multiple processes within a single piece of monolithic code. C doesn't support formatted I/O either, but it can be implemented quite effectively in C. So can concurrent processes. I've done it, in fact. The resulting code is 90% portable (the 10% being the code that handles the context switch). -- -- Peter da Silva, Ferranti International Controls Corporation -- Sugar Land, TX 77487-5012; +1 713 274 5180 -- "Have you hugged your wolf today?" 0, unseen,, *** EOOH *** From: kt4@prism.gatech.EDU (Ken Thompson) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 3 Feb 92 23:07:54 GMT Organization: Georgia Institute of Technology viewpoint may be largely unrelated to its usefulness. Many if not most of the software we use is probably obsolete according to the latest design criteria. Most users could probably care less if the internals of the operating system they use is obsolete. They are rightly more interested in its performance and capabilities at the user level. I would generally agree that microkernels are probably the wave of the future. However, it is in my opinion easier to implement a monolithic kernel. It is also easier for it to turn into a mess in a hurry as it is modified.

Regards, Ken -- Ken Thompson GTRI, Ga. Tech, Atlanta Ga. 30332 Internet:!kt4@prism.gatech.edu uucp:...!{allegra,amd,hplabs,ut-ngp}!gatech!prism!kt4 "Rowe's Rule: The odds are five to six that the

light at the end of the tunnel is the headlight of an oncoming train." -- Paul Dickson 0, unseen,, *** EOOH *** From: kevin@taronga.taronga.com (Kevin Brown) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 4 Feb 92 08:08:42 GMT Organization: University of Houston In article <47607@hydra.gatech.EDU> kt4@prism.gatech.EDU (Ken Thompson) writes: >viewpoint may be largely unrelated to its usefulness. Many if not >most of the software we use is probably obsolete according to the >latest design criteria. Most users could probably care less if the >internals of the operating system they use is obsolete. They are >rightly more interested in its performance and capabilities at the >user level. > >I would generally agree that microkernels are probably the wave of >the future. However, it is in my opinion easier to implement a >monolithic kernel. It is also easier for it to turn into a mess in >a hurry as it is modified. How difficult is it to structure the source tree of a monolithic kernel such that most modifications don't have a large negative impact on the source? What sorts of pitfalls do you run into in this sort of endeavor, and what suggestions do you have for dealing with them? I guess what I'm asking is: how difficult is it to organize the source such that most changes to the kernel remain localized in scope, even though the kernel itself is monolithic? I figure you've got years of experience with monolithic kernels :-), so I'd think you'd have the best shot at answering questions like these. >Ken Thompson GTRI, Ga. Tech, Atlanta Ga. 30332 Internet:!kt4@prism.gatech.edu >uucp:...!{allegra,amd,hplabs,ut-ngp}!gatech!prism!kt4 >"Rowe's Rule: The odds are five to six that the light at the end of the >tunnel is the headlight of an oncoming train." -- Paul Dickson

Kevin Brown 0, unseen,, *** EOOH *** From: rburns@finess.Corp.Sun.COM (Randy Burns) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 30 Jan 92 20:33:07 GMT Organization: Sun Microsystems, Mt. View, Ca. NNTP-Posting-Host: finess.corp.sun.com In article <12615@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes: >In article <1992Jan29.231426.20469@klaava.Helsinki.FI> torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds) writes: >Of course 5 years from now that will be different, but 5 years from now

>everyone will be running free GNU on their 200 MIPS, 64M SPARCstation-5. Well, I for one would _love_ to see this happen. >>The fact is that linux is more portable than minix. What? I hear you >>say. It's true - but not in the sense that ast means: I made linux as >>conformant to standards as I knew how (without having any POSIX standard >>in front of me). Porting things to linux is generally /much/ easier >>than porting them to minix. ..... >My point is that writing a new operating system that is closely tied to any >particular piece of hardware, especially a weird one like the Intel line, >is basically wrong. First off, the parts of Linux tuned most finely to the 80x86 are the Kernel and the devices. My own sense is that even if Linux is simply a stopgap measure to let us all run GNU software, it is still worthwhile to have a a finely tuned kernel for the most numerous architecture presently in existance. > An OS itself should be easily portable to new hardware >platforms. Well, the only part of Linux that isn't portable is the kernel and drivers. Compare to the compilers, utilities, windowing system etc. this is really a small part of the effort. Since Linux has a large degree of call compatibility with portable OS's I wouldn't complain. I'm personally very grateful to have an OS that makes it more likely that some of us will be able to take advantage of the software that has come out of Berkeley, FSF, CMU etc. It may well be that in 2-3 years when ultra cheap BSD variants and Hurd proliferate, that Linux will be obsolete. Still, right now Linux greatly reduces the cost of using tools like gcc, bison, bash which are useful in the development of such an OS. 0, unseen,, *** EOOH *** From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 31 Jan 92 10:33:23 GMT Organization: University of Helsinki In article <12615@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes: >The limitations of MINIX relate at least partly to my being a professor: >An explicit design goal was to make it run on cheap hardware so students >could afford it. All right: a real technical point, and one that made some of my comments inexcusable. But at the same time you shoot yourself in the foot a bit: now you admit that some of the errors of minix were that it was too portable: including machines that weren't really designed to run unix. That assumption lead to the fact that minix now cannot easily be extended to have things like paging, even for

machines that would support it. Yes, minix is portable, but you can rewrite that as "doesn't use any features", and still be right. >A multithreaded file system is only a performance hack. Not true. It's a performance hack /on a microkernel/, but it's an automatic feature when you write a monolithic kernel - one area where microkernels don't work too well (as I pointed out in my personal mail to ast). When writing a unix the "obsolete" way, you automatically get a multithreaded kernel: every process does it's own job, and you don't have to make ugly things like message queues to make it work efficiently. Besides, there are people who would consider "only a performance hack" vital: unless you have a cray-3, I'd guess everybody gets tired of waiting on the computer all the time. I know I did with minix (and yes, I do with linux too, but it's /much/ better). >I still maintain the point that designing a monolithic kernel in 1991 is >a fundamental error. Be thankful you are not my student. You would not >get a high grade for such a design :-) Well, I probably won't get too good grades even without you: I had an argument (completely unrelated - not even pertaining to OS's) with the person here at the university that teaches OS design. I wonder when I'll learn :) >My point is that writing a new operating system that is closely tied to any >particular piece of hardware, especially a weird one like the Intel line, >is basically wrong. But /my/ point is that the operating system /isn't/ tied to any processor line: UNIX runs on most real processors in existence. Yes, the /implementation/ is hardware-specific, but there's a HUGE difference. You mention OS/360 and MS-DOG as examples of bad designs as they were hardware-dependent, and I agree. But there's a big difference between these and linux: linux API is portable (not due to my clever design, but due to the fact that I decided to go for a fairly- well-thought-out and tested OS: unix.) If you write programs for linux today, you shouldn't have too many surprises when you just recompile them for Hurd in the 21st century. As has been noted (not only by me), the linux kernel is a miniscule part of a complete system: Full sources for linux currently runs to about 200kB compressed - full sources to a somewhat complete developement system is at least 10MB compressed (and easily much, much more). And all of that source is portable, except for this tiny kernel that you can (provably: I did it) re-write totally from scratch in less than a year without having /any/ prior knowledge. In fact the /whole/ linux kernel

is much smaller than the 386-dependent things in mach: i386.tar.Z for the current version of mach is well over 800kB compressed (823391 bytes according to nic.funet.fi). Admittedly, mach is "somewhat" bigger and has more features, but that should still tell you something.

Linus 0, unseen,, *** EOOH *** Newsgroups: comp.os.minix From: kevin@nuchat.sccsi.com (Kevin Brown) Subject: Re: LINUX is obsolete Organization: Where??? Date: Fri, 31 Jan 1992 07:43:47 GMT Sorry, but I just can't resist this thread...:-) In article <12615@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes: >In article <1992Jan29.231426.20469@klaava.Helsinki.FI> torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds) writes: >>You use this [being a professor] as an excuse for the limitations of minix? >The limitations of MINIX relate at least partly to my being a professor: >An explicit design goal was to make it run on cheap hardware so students >could afford it. In particular, for years it ran on a regular 4.77 MHZ PC >with no hard disk. And an explicit design goal of Linux was to take advantage of the special features of the 386 architecture. So what exactly is your point? Different design goals get you different designs. You ought to know that. >You could do everything here including modify and recompile >the system. Just for the record, as of about 1 year ago, there were two >versions, one for the PC (360K diskettes) and one for the 286/386 (1.2M). >The PC version was outselling the 286/386 version by 2 to 1. I don't have >figures, but my guess is that the fraction of the 60 million existing PCs that >are 386/486 machines as opposed to 8088/286/680x0 etc is small. Among students >it is even smaller. I find it very interesting that you claim here that Minix was designed primarily for cheap hardware (in particular, the IBM PC/XT with no hard disk) and yet elsewhere have also mentioned the virtues of being portable across hardware platforms. Well, if you insist on designing the thing with the lowest common denominator as your basis, that's fine, but of course the end result will be less than pretty unless designed *very* carefully. >Making software free, but only for folks with enough money >to buy first class hardware is an interesting concept. Except that Linux was designed more for the purposes of the designer than anything else. If I were writing an OS, I'd design it to suit myself, too. It's just that Linus was nice enough to share his code with the rest of us. >Of course 5 years from now that will be

different, but 5 years from now >everyone will be running free GNU on their 200 MIPS, 64M SPARCstation-5. Maybe. But by then, the 386/486 will probably be where the PC is now: everyone will have one and they'll be dirt cheap. The timing will be about right. In which case Linux will fit right in, wouldn't you say? >>Re 2: your job is being a professor and researcher: That's one hell of a >>good excuse for some of the brain-damages of minix. I can only hope (and >>assume) that Amoeba doesn't suck like minix does. >Amoeba was not designed to run on an 8088 with no hard disk. Here's a question for you: as a general rule, when you go to design an operating system, do you design it for specific capabilities and then run it on whatever hardware will do the job, or do you design it with the hardware as a target and fit the capabilities to the hardware? With respect to Minix, it seems you did the latter, but I don't know whether or not you did that with Amoeba. >>If this was the only criterion for the "goodness" of a kernel, you'd be >>right. What you don't mention is that minix doesn't do the micro-kernel >>thing very well, and has problems with real multitasking (in the >>kernel). If I had made an OS that had problems with a multithreading >>filesystem, I wouldn't be so fast to condemn others: in fact, I'd do my >>damndest to make others forget about the fiasco. >A multithreaded file system is only a performance hack. Bull. A multithreaded file system has a completely different design than a single-threaded file system and has different design criteria than a single-threaded file system. >When there is only one job active, the normal case on a small PC, it buys >you nothing and adds complexity to the code. If there is only going to be one job active anyway then *why bother with multitasking at all*???? If you're going to implement multitasking, then don't do a halfway job of it. On the other hand, if you're going to assume that there will be only one job active anyway, then don't bother with multitasking (after all, it *does* complicate things :-). >On machines fast enough to >support multiple users, you probably have enough buffer cache to insure a >hit cache hit rate, in which case multithreading also buys you nothing. Maybe. Multiple users means multiple things being done simultaneously. I wouldn't bet on the buffer cache buying you so much that multithreading makes no difference. It's one thing if the users are doing something simple, like editing a file. It's another thing if they're compiling, reading news, or

other things that touch lots of different files. >It is only a win when there are multiple processes actually doing real disk >I/O. Which happens a *lot* when you're running multiple users. Or when you're a machine hooked up to the net and handling news traffic. >Whether it is worth making the system more complicated for this case is >at least debatable. Oh, come on. How tough is it to implement a multi-threaded file system? All you need is a decent *buffered* (preferably infinitely so) message-passing system and a way to save your current state when you send out a request to the device driver(s) to perform some work (and obviously some way to restore that state). Minix has the second via the setjmp()/ longjmp() mechanism, but lacks the former in a serious way. >I still maintain the point that designing a monolithic kernel in 1991 is >a fundamental error. Not if you're trying to implement the system call semantics of Unix in a reasonably simple and elegant way. >Be thankful you are not my student. You would not >get a high grade for such a design :-) Why not? What's this big thing against monolithic kernels? There are certain classes of problems for which a monolithic kernel is a more appropriate design than a microkernel architecture. I think implementing Unix semantics with a minimum of fuss is one such problem. Unless you can suggest an elegant way to terminate a system call upon receipt of a signal from within a microkernel OS? >>The fact is that linux is more portable than minix. What? I hear you >>say. It's true - but not in the sense that ast means: I made linux as >>conformant to standards as I knew how (without having any POSIX standard >>in front of me). Porting things to linux is generally /much/ easier >>than porting them to minix. >MINIX was designed before POSIX, and is now being (slowly) POSIXized as >everyone who follows this newsgroup knows. Everyone agrees that user-level >standards are a good idea. As an aside, I congratulate you for being able >to write a POSIX-conformant system without having the POSIX standard in front >of you. I find it difficult enough after studying the standard at great length. > >My point is that writing a new operating system that is closely tied to any >particular piece of hardware, especially a weird one like the Intel line, >is basically wrong. Weird as the Intel line may be, it's *the* most popular line, by several times. So it's not like it's *that* big a loss. And Intel hardware is at least relatively cheap to come by, regardless of what

your students might tell you (why do you think they all own PCs?)... >An OS itself should be easily portable to new hardware >platforms. As long as you don't sacrifice too much in the way of performance or architectural elegance in order to gain this. Unfortunately, that's *exactly* what happened with Minix: in attempting to implement it on hardware of the lowest caliber, you ended up having to make design decisions with respect to the architecture and implementation that have made vintage Minix unusable as anything more than a personal toy operating system. For example: why didn't you implement a system call server as a layer between the file system and user programs? My guess: you didn't have enough memory on the target machine to do it. Put another way: you hit your original goal right on target, and are to be applauded for that. But in doing so, you missed a lot of other targets that wouldn't have been hard to hit as well, with some consideration of them. I think. But I wasn't there when you were making the decisions, so it's real hard for me to say for sure. I'm speaking from hindsight, but you had the tough problem of figuring out what to do without such benefit. Now, *modified* Minix is usable. Add a bigger buffer cache. Modify it so that it can take advantage of 386 protected mode. Fix the tty driver so that it will give you multiple consoles. Fix the rs232 driver to deal with DCD/DTR and do the right thing when carrier goes away. Fix the pipes so that read and write requests don't fail just because they happen to be bigger than the size of a physical pipe. Add shared text segments so you maximize the use of your RAM. Fix the scheduler so that it deals with character I/O bound processes in a reasonable way. >When OS/360 was written in assembler for the IBM 360 >25 years ago, they probably could be excused. When MS-DOS was written >specifically for the 8088 ten years ago, this was less than brilliant, as >IBM and Microsoft now only too painfully realize. Yeah, right. Just what hardware do you think they'd like to port DOS to, anyway? I can't think of any. I don't think IBM or Microsoft are regretting *that* particular aspect of DOS. Rather, they're probably regretting the fact that it was written for the address space provided by the 8088. MS-DOS isn't less than brilliant because it was written for one machine architecture. It's less than brilliant because it doesn't do anything well, *regardless* of its portability or lack thereof. >Writing a new OS only for the >386 in 1991 gets you your second 'F' for this term. But if you

do real well >on the final exam, you can still pass the course. He made his code freely redistributable. *You* didn't even do that. Just for that move alone, he scores points in my book. Of course, the distribution technology available to him is much better than what was available when you did Minix, so it's hard to fault you for that... But I must admit, Minix is still one hell of a bargain, and I would never hesitate to recommend it to anyone who wants to learn something about Unix and operating systems in general. As a working operating system (i.e., one intended for a multi-user environment), however, I'd hesitate to recommend it, except that there really aren't any good alternatives (except Linux, of course, at least tentatively. I can't say for sure, since I haven't checked out Linux yet), since it doesn't have the performance capabilities that a working operating system needs. >Prof. Andrew S. Tanenbaum (ast@cs.vu.nl) Kevin Brown 0, unseen,, *** EOOH *** Newsgroups: comp.os.minix From: kevin@taronga.taronga.com (Kevin Brown) Subject: Re: LINUX is obsolete Organization: Sorely lacking. Date: Mon, 3 Feb 1992 05:12:58 GMT It has been brought to my attention that my last posting was exceedingly harsh. Having reread it, I'm inclined to agree. Dr. Tanenbaum claims that the microkernel architecture is the way to go. He has a great deal more experience with operating systems than I have. It's an understatement that it's likely that there's some substance to his statement. :-) Many of the things I said in my previous posting were more a result of my philosophical viewpoint on operating systems and programming in general than experience. And the particular viewpoint I hold that's relevent to the discussion says that the method of implementation chosen depends on the design goals, and that there is no "wrong" or "right" way to do things that is independent of such goals. Thus, my statement that a monolithic kernel follows from some design goals, e.g. ease of implementation of the semantics of the Unix API. In particular, the ease of implementing things like signal handling, premature system call termination, etc. At least, that's the conclusion I come to when I think about the problem. My experience with Minix says that there are a number of things that should not go in a user process, things that are better left in the kernel. Things like memory allocation (which requires global knowledge of the hardware, something that a user process should, IMHO, not have) and signal

handling (which requires building stack frames). ÷³So from my point of view, the architecture of Minix is not ideal. While it may win in that it's a "microkernel" architecture, the division of functionality is not entirely to my liking. As is undoubtedly plainly obvious by now. :-) Despite that, Minix is quite usable in many ways as a personal operating system, i.e. one where there is usually only one person logged into the system. If I gave the impression that I thought it was unusable in general, then I apologize for that. However, as a *multiuser* operating system, þòi.e. an operating system designed to efficiently meet the needs of multiple users simultaneously while also performing batch operations, Minix is lacking, as far as I'm concerned. The main reason, of course, is the single-threaded file system (hereafter, STFS). Now, Dr. Tanenbaum may feel that a multi-threaded file system (hereafter, MTFS) is merely a performance hack. Perhaps he's right. Perhaps the architecture of a MTFS is sufficiently similar to that of a STFS that his assessment is correct. My vision of a MTFS may differ significantly from his, and this would explain why he and I seem to have a difference of opinion on this matter. Regardless of whether or not a MTFS is a "performance hack", for a *multiuser* operating system, I think there are a lot of good arguments that say that a MTFS is a *necessary* "performance hack". Provided, of course, that one does not have infinite buffer cache resources. :-) There are other things I feel Minix lacks as well. The ability to allocate memory in the kernel is one (such an ability would allow any user process, e.g. device drivers and the file system, to allocate memory dynamically. This is useful for doing things like resizing the buffer cache on the fly, etc. The ability to pass arbitrarily sized messages, optionally via shared memory, is another (such an ability might be limited by constraints like page size and such). However much Minix may be lacking from my standpoint, it is nevertheless a very useful and welcome enhancement to my system. In spite of the impression that I may have given everyone in my last posting, there will always be a soft spot in my heart for it, if only because it's the first decent operating system I've had on my system that I've had source to. I don't have to tell you people how incredibly useful it is to have source. You already know. It is very important to me to have source code to the things I run. It bothers me a great deal to run things that I don't have source to. Even the C

compiler. And the less expensive the source is, the better. This is why Dr. Tanenbaum's statements about Linux touched a raw nerve with me: Linux comes with source *and* it's free. And it's available right now. Someone, either here on this newsgroup or over on alt.os.linux, made a very valid observation: the cost of a 16 MHz 386SX system is about $140 more than a comparably equipped (in terms of RAM size, display technology, hard drive space, etc.) 8088 system. Minix is $169. In economic terms, Linux wins if you have to buy Minix. Where Minix wins (or is at least even :-) is when you can get it for free via the educational distribution clause of the license agreement. However, Minix will run even better on a 16 MHz 386SX than on an 8088. If I were a student, I'd get the 386SX unless I simply didn't have a choice. Then I'd get whichever operating system I could get for the least cost. If I could get both for free, then I'd get both. :-) Given the reasons Linus wrote Linux, I think it's hard for anyone to fault him for writing it the way he did. And he was extremely nice in making his code freely available to the rest of the world. It's not something he had to do. In my book, that makes him almost beyond reproach. Dr. Tanenbaum didn't make Minix free. His goals were different. Minix is a teaching aid above all else (unless Dr. Tanenbaum has changed his views about Minix :-). That means that he must be concerned with the most efficient way to get Minix to the student population. At the time Minix was released, Prentice-Hall was a good solution, and has been for some time. However, I must wonder whether or not this is still the case. Dr. Tanenbaum: do you still feel that free distribution of Minix via the net is not the best way to distribute Minix? Which wins? Minix or Linux? Depends on how you measure them... Kevin Brown 0, unseen,, *** EOOH *** From: kaufman@eecs.nwu.edu (Michael L. Kaufman) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 3 Feb 92 22:27:48 GMT Organization: EECS Department, Northwestern University I tried to send these two posts from work, but I think they got eaten. If you have seen them already, sorry. ---------- - Andy Tanenbaum writes an interesting article (also interesting was finding out that he actually reads this group) but I think he is missing an important point. He Wrote: >As most of you know, for me MINIX is a hobby, ... Which is also probably true of most, if not all, of the people

who are involved in Linux. We are not developing a system to take over the OS market, we are just having a good time. > What is going to happen > is that they will gradually take over from the 80x86 line. They will > run old MS-DOS programs by interpreting the 80386 in software. Well when this happens, if I still want to play with Linux, I can just run it on my 386 simulator. > MINIX was designed to be reasonably portable, and has been ported from the > Intel line to the 680x0 (Atari, Amiga, Macintosh), SPARC, and NS32016. > LINUX is tied fairly closely to the 80x86. Not the way to go. That's fine for the people who have those machines, but it wasn't a free lunch. That portibility was gained at the cost of some performance and some features on the 386. Before you decide that LINUX is not the way to go, you should think about what it is going to be used for. I am going to use it for running memory and computation intensive graphics programs on my 486. For me, speed and memory were more important then future state-of-the-artness and portability. >But in all honesty, I would >suggest that people who want a **MODERN** "free" OS look around for a >microkernel-based, portable OS, like maybe GNU or something like that. I don't know of any free microkernel-based, portable OSes. GNU is still vaporware, and likely to remain that way for the forseeable future. Do you actually have one to recomend, or are you just toying with me? ;-) ---------- In article <12615@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes: >My point is that writing a new operating system that is closely tied to any >particular piece of hardware, especially a weird one like the Intel line, >is basically wrong. An OS itself should be easily portable to new hardware >platforms. I think I see where I disagree with you now. You are looking at OS design as an end in itself. Minix is good because it is portable/Micro-Kernal/etc. Linux is not good because it is monolithic/tightly tied to Intel/etc. That is not a strange attitude for someone in the acedemic world, but it is not something you should expect to be universally shared. Linux is not being written as a teaching tool, or as an abstract exercise. It is being written to allow people to run GNU-type software _today_. The fact that it may not be in use in five years is less important then the fact that today (well, by April probably) I can run all sorts of software on it that I want to run. You keep saying that Minix is better, but if it will not run

the software that I want to run, it really isn't that good (for me) at all. > When OS/360 was written in assembler for the IBM 360 >25 years ago, they probably could be excused. When MS-DOS was written >specifically for the 8088 ten years ago, this was less than brilliant, as >IBM and Microsoft now only too painfully realize. Same point. MSoft did not come out with Dos to "explore the frontiers of os research". They did it to make a buck. And considering the fact that MS-DOS probably still outsells everyone else put together, I don't think that you say that they have failed _in their goals_. Not that MS-Dos is the best OS in terms of anything else, only that it has served their needs. Michael -- Michael Kaufman | I've seen things you people wouldn't believe. Attack ships on kaufman | fire off the shoulder of Orion. I watched C-beams glitter in @eecs.nwu.edu | the dark near the Tannhauser gate. All those moments will be | lost in time - like tears in rain. Time to die. Roy Batty 0, unseen,, *** EOOH *** From: entropy@wintermute.WPI.EDU (Lawrence C. Foard) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 6 Feb 92 09:22:40 GMT Organization: Worcester Polytechnic Institute Nntp-Posting-Host: wintermute.wpi.edu In article <1992Feb3.051258.4153@menudo.uh.edu> kevin@taronga.taronga.com (Kevin Brown) writes: >Dr. Tanenbaum claims that the microkernel architecture is the way to go. >He has a great deal more experience with operating systems than I have. >It's an understatement that it's likely that there's some substance to >his statement. :-) I tend to prefer seeing for myself rather than accepting "expert" opinion. Microkernels are nice asthetically, but there are times when practical issues must also be considered :) >w3So from my point of view, the architecture of Minix is not ideal. While >it may win in that it's a "microkernel" architecture, the division of >functionality is not entirely to my liking. As is undoubtedly plainly >obvious by now. :-) I've been told by people who have used both that Linux is significantly faster. There are certainly several factors involved (certainly using 32 bits helps alot), but the multithreading also makes for much lower overhead. >However, as a *multiuser* operating system, ~ri.e. an operating system designed >to efficiently meet the needs of multiple users simultaneously while also >performing batch operations, Minix is lacking, as far as I'm concerned. >The main

reason, of course, is the single-threaded file system (hereafter, >STFS). Now, Dr. Tanenbaum may feel that a multi-threaded file system >(hereafter, MTFS) is merely a performance hack. I think this is a very valid problem. There are two ways a single threaded FS could work and both have substantial problems. If the FS blocks while waiting for I/O it would be completely unusable for "real" work. Imagine several users accessing a database, if the FS blocks for I/O they will have to wait eventhough the data they are looking for is already in the cache. If it is designed to be non blocking then it is even more complicated than a multithreaded FS and will have more overhead. I hope it is atleast the second >However much Minix may be lacking from my standpoint, it is nevertheless >a very useful and welcome enhancement to my system. In spite of the >impression that I may have given everyone in my last posting, there will >always be a soft spot in my heart for it, if only because it's the first >decent operating system I've had on my system that I've had source to. >I don't have to tell you people how incredibly useful it is to have source. >You already know. I will agree here, Minix is infinitly better than Messy-Loss :) >Given the reasons Linus wrote Linux, I think it's hard for anyone to fault >him for writing it the way he did. And he was extremely nice in making >his code freely available to the rest of the world. It's not something he >had to do. In my book, that makes him almost beyond reproach. I think more effort has been put into making practical use of Linux possible. An educational OS is nice, but there is a world outside of colleges that is suffering from the lack of cheap and useful OS's, I've been stuck doing most consulting work in Messy Loss because customers don't want to fork out $1000 for UNIX. >Dr. Tanenbaum didn't make Minix free. His goals were different. Minix >is a teaching aid above all else (unless Dr. Tanenbaum has changed his >views about Minix :-). That means that he must be concerned with the >most efficient way to get Minix to the student population. At the time >Minix was released, Prentice-Hall was a good solution, and has been for >some time. However, I must wonder whether or not this is still the case. >Dr. Tanenbaum: do you still feel that free distribution of Minix via the >net is not the best way to distribute Minix? I would guess that Prentice-Hall would have some objections :) -- Disclaimer: Opinions are based on logic rather than biblical "fact". ------ This is your friendly | First they came for the

drug users, I said \ / neighborhood signature virus | nothing, then they came for hackers, \ / please add me to your signature! | I said nothing... STOP W.O.D. \/ 0, unseen,, *** EOOH *** From: kevin@taronga.com (Kevin Brown) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 9 Feb 92 09:02:24 GMT Organization: What's that? In article <1992Feb6.092240.16377@wpi.WPI.EDU> entropy@wintermute.WPI.EDU (Lawrence C. Foard) writes: >In article <1992Feb3.051258.4153@menudo.uh.edu> kevin@taronga.taronga.com (Kevin Brown) writes: >>Dr. Tanenbaum claims that the microkernel architecture is the way to go. >>He has a great deal more experience with operating systems than I have. >>It's an understatement that it's likely that there's some substance to >>his statement. :-) > >I tend to prefer seeing for myself rather than accepting "expert" opinion. >Microkernels are nice asthetically, but there are times when practical issues >must also be considered :) I agree. This is why I qualified my statement the way I did. :-) Having seen both monolithic and microkernel architectures running, though, I tend to agree that microkernels are generally the way to go, all other things being equal. But as you say, all things are not always equal. That's when it becomes a judgement call. Which is better? Depends on what you're trying to do. >I've been told by people who have used both that Linux is significantly >faster. There are certainly several factors involved (certainly using 32 bits >helps alot), but the multithreading also makes for much lower overhead. Yup. I think that if Minix were arranged so that it had message queueing and a true multithreaded filesystem, it might be comparable to a monolithic kernel in terms of speed. It's hard for me to say, though. I haven't played around much with multithreaded filesystems, so I don't know how hard it is to make them work efficiently. I'd think, though, that it would depend enormously on how efficient your device drivers were, and how much data copying you'd have to do (ideally, you'd pass references to the data buffers around and do your actual data transfers directly to the user's buffer). >>However, as a *multiuser* operating system, i.e. an operating system designed >>to efficiently meet the needs of multiple users simultaneously while also >>performing batch operations, Minix is lacking, as far as I'm concerned. >>The main reason, of course, is

the single-threaded file system (hereafter, >>STFS). Now, Dr. Tanenbaum may feel that a multi-threaded file system >>(hereafter, MTFS) is merely a performance hack. > >I think this is a very valid problem. There are two ways a single threaded FS >could work and both have substantial problems. If the FS blocks while waiting >for I/O it would be completely unusable for "real" work. Imagine several users >accessing a database, if the FS blocks for I/O they will have to wait >eventhough the data they are looking for is already in the cache. If it is >designed to be non blocking then it is even more complicated than a >multithreaded FS and will have more overhead. I hope it is atleast the second I haven't gone deeply into the source code of the Minix file system, but the impression I get from my perusing of it is that it blocks on disk I/O but not on terminal I/O, the idea being that disk I/O requests will almost always be satisfied relatively soon after they are made, whereas terminal I/O requests can take an indefinite amount of time to satisfy. But it seems to me that if you're going to implement the mechanism to handle I/O where the file system doesn't block waiting for it, why not use that mechanism universally??? >>However much Minix may be lacking from my standpoint, it is nevertheless >>a very useful and welcome enhancement to my system. In spite of the >>impression that I may have given everyone in my last posting, there will >>always be a soft spot in my heart for it, if only because it's the first >>decent operating system I've had on my system that I've had source to. >>I don't have to tell you people how incredibly useful it is to have source. >>You already know. > >I will agree here, Minix is infinitly better than Messy-Loss :) Which is why I try to avoid using MS-DOS whenever possible. I'll bet a lot of us Minixers do the same. :-) >>Given the reasons Linus wrote Linux, I think it's hard for anyone to fault >>him for writing it the way he did. And he was extremely nice in making >>his code freely available to the rest of the world. It's not something he >>had to do. In my book, that makes him almost beyond reproach. > >I think more effort has been put into making practical use of Linux possible. >An educational OS is nice, but there is a world outside of colleges that >is suffering from the lack of cheap and useful OS's, I've been stuck doing >most consulting work in Messy Loss because customers don't want to fork out >$1000 for UNIX. Even students can make good use of something like Linux. I have 8

megabytes of RAM on my machine, and 410 meg of harddrive space. Yet I can barely run SBProlog on my system, even though my system is considerably more macho than most. If I had demand paging on my system, this wouldn't be a problem, but the only patches I have for demand paging seem not to work very well. Once Linux becomes more stable (and gets support for Seagate ST-02 SCSI), I'll snag the sources and check it out. Since I already own Minix, I can legally transport *everything* over to it, and since both share the same filesystem layout, I can do the transporting with a minimum of hassle. >>Dr. Tanenbaum didn't make Minix free. His goals were different. Minix >>is a teaching aid above all else (unless Dr. Tanenbaum has changed his >>views about Minix :-). That means that he must be concerned with the >>most efficient way to get Minix to the student population. At the time >>Minix was released, Prentice-Hall was a good solution, and has been for >>some time. However, I must wonder whether or not this is still the case. >>Dr. Tanenbaum: do you still feel that free distribution of Minix via the >>net is not the best way to distribute Minix? > >I would guess that Prentice-Hall would have some objections :) No doubt. :-(-- Kevin Brown Disclaimer: huh? kevin@taronga.com kevin@nuchat.sccsi.com 0, unseen,, *** EOOH *** From: julien@incal.inria.fr (Julien Maisonneuve) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 3 Feb 92 17:10:14 GMT I would like to second Kevin brown in most of his remarks. I'll add a few user points : - When ast states that FS multithreading is useless, it reminds me of the many times I tried to let a job run in the background (like when reading an archive on a floppy), it is just unusable, the & shell operator could even have been left out. - Most interesting utilities are not even compilable under Minix because of the ATK compiler's incredible limits. Those were hardly understandable on a basic PC, but become absurd on a 386. Every stupid DOS compiler has a large model (more expensive, OK). I hate the 13 bit compress ! - The lack of Virtual Memory support prevents people studying this area to experiment, and prevents users to use large programs. The strange design of the MM also makes it hard to modify. The problem is that even doing exploratory work under minix is painful. If you want to get any work done (or even fun), even DOS is becoming a better alternative (with things like DJ GPP). In its basic

form, it is really no more than OS course example, a good toy, but a toy. Obtaining and applying patches is a pain, and precludes further upgrades. Too bad when not so much is missing to make it really good. Thanks for the work andy, but Linux didn't deserve your answer. For the common people, it does many things better than Minix.

Julien Maisonneuve. This is not a flame, just my experience. 0, unseen,, *** EOOH *** From: richard@aiai.ed.ac.uk (Richard Tobin) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 4 Feb 92 14:46:49 GMT Reply-To: richard@aiai.UUCP (Richard Tobin) Organization: AIAI, University of Edinburgh, Scotland In article <12615@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes: >A multithreaded file system is only a performance hack. When there is only >one job active, the normal case on a small PC, it buys you nothing I find the single-threaded file system a serious pain when using Minix. I often want to do something else while reading files from the (excruciatingly slow) floppy disk. I rather like to play rogue while waiting for large C or Lisp compilations. I look to look at files in one editor buffer while compiling in another. (The problem would be somewhat less if the file system stuck to serving files and didn't interact with terminal i/o.) Of course, in basic Minix with no virtual consoles and no chance of running emacs, this isn't much of a problem. But to most people that's a failure, not an advantage. It just isn't the case that on single-user machines there's no use for more than one active process; the idea only has any plausibility because so many people are used to poor machines with poor operating systems. As to portability, Minix only wins because of its limited ambitions. If you wanted a full-featured Unix with paging, job-control, a window system and so on, would it be quicker to start from basic Minix and add the features, or to start from Linux and fix the 386-specific bits? I don't think it's fair to criticise Linux when its aims are so different from Minix's. If you want a system for pedagogical use, Minix is the answer. But if what you want is an environment as much like (say) a Sun as possible on your home computer, it has some deficiencies. -- Richard -- Richard Tobin, AI Applications Institute, R.Tobin@ed.ac.uk Edinburgh University. 0, unseen,, *** EOOH *** From: ast@cs.vu.nl (Andy Tanenbaum) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 5 Feb 92 14:48:48 GMT Organization: Fac. Wiskunde &

Informatica, Vrije Universiteit, Amsterdam In article <6121@skye.ed.ac.uk> richard@aiai.UUCP (Richard Tobin) writes: >If you wanted a full-featured Unix with paging, job-control, a window >system and so on, would it be quicker to start from basic Minix and >add the features, or to start from Linux and fix the 386-specific >bits? Another option that seems to be totally forgotten here is buy UNIX or a clone. If you just want to USE the system, instead of hacking on its internals, you don't need source code. Coherent is only $99, and there are various true UNIX systems with more features for more money. For the true hacker, not having source code is fatal, but for people who just want a UNIX system, there are many alternatives (albeit not free). Andy Tanenbaum (ast@cs.vul.nl) 0, unseen,, *** EOOH *** From: linville@garfield.catt.ncsu.edu (John W. Linville) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Summary: Small model compiler limits Coherent as an alternative Date: 5 Feb 92 17:56:35 GMT Organization: NCSU CATT Prog In article <12696@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes: >In article <6121@skye.ed.ac.uk> richard@aiai.UUCP (Richard Tobin) writes: >>If you wanted a full-featured Unix with paging, job-control, a window >>system and so on, would it be quicker to start from basic Minix and >>add the features, or to start from Linux and fix the 386-specific >>bits? > >Another option that seems to be totally forgotten here is buy UNIX or a >clone. If you just want to USE the system, instead of hacking on its >internals, you don't need source code. Coherent is only $99, and there >are various true UNIX systems with more features for more money. For the >true hacker, not having source code is fatal, but for people who just >want a UNIX system, there are many alternatives (albeit not free). > >Andy Tanenbaum (ast@cs.vul.nl) Coherent is limited by a compiler that only supports the small memory model, making it just as difficult (perhaps more in some instances) to port 'standard' Unix programs to Coherent as it can be under Minix. Also, Coherent is not portable (or at least, to the best of my knowledge, has not been ported), so this advocacy contradicts one of your arguments against Linux. Since a true Unix system often costs as much as the machine it runs on (even more since many Unix providers un-bundle networking and development packages), buying a true Unix system is more than beyond the budget of many people. John

W. Linville 0, unseen,, *** EOOH *** From: jerry@connection.prospect.com (Jerry Shekhel) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 6 Feb 92 21:28:05 GMT Organization: Polygen Corporation, Waltham, MA linville@garfield.catt.ncsu.edu (John W. Linville) writes: > >Since a true Unix system often costs as much as the machine it runs on (even >more since many Unix providers un-bundle networking and development packages), >buying a true Unix system is more than beyond the budget of many people. > For those who may be interested, MST sells System V Release 4.0.3 for the 386/486 for $399 including development system, $499 if you need networking. X11R5 binaries may be obtained via FTP (networking is not required for X11R5). I have just such a setup, and it works great. MST's version of UNIX doesn't have too much in the way of bug fixes relative to the AT&T code, but the only thing I've really had problems with was a couple of bugs in csh. Now that I have tcsh working (built without so much as a warning!) I'll never go back :-) Micro Station Technology 1140 Kentwood Avenue Cupertine, CA 95014 Tel: 408-253-3898 Fax: 408-253-7853 I am not affiliated with MST except as a customer. > >John W. Linville > -- +------------------+---------------------+---------------------------+ | JERRY J. SHEKHEL | POLYGEN CORPORATION | When I was young, I had to walk | | Drummers do it... | Waltham, MA USA | to school and back every day -- | | ... In rhythm! | (617) 890-2175 | 20 miles, uphill both ways. | +------------------+---------------------+---------------------------------+ | ...! [princeton mit-eddie bu sunne] !polygen!jerry | | jerry@polygen.com | +---+ 0, unseen,, *** EOOH *** From: meggin@epas.utoronto.ca (David Megginson) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 5 Feb 92 20:50:55 GMT Organization: University of Toronto - EPAS Nntp-Posting-Host: epas.utoronto.ca In article <12696@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes: >Another option that seems to be totally forgotten here is buy UNIX or a >clone. If you just want to USE the system, instead of hacking on its >internals, you don't need source code. Coherent is only $99, and there >are various true UNIX systems with more features for more money. For the >true hacker, not having source

code is fatal, but for people who just >want a UNIX system, there are many alternatives (albeit not free). What Unix's _are_ available for a simple, M68000-based ST, with _or_ without source? These are the only options I know of: 1) OS 9. 2) The Beckmeyer MT C-Shell. 3) MiNT. 4) Minix. I have used all of these except for OS 9, and Minix is clearly the closest thing to Unix that I can run (though it is easier to port BSD programs to MiNT using the MiNT gcc library). I could shell out CAN $3000 for a TT, but then I may as well buy a 386 box anyway. Besides, I _like_ having the source. The extra advantage of Minix is that the user base is a lot wider than the ST market, so I can get decent system enhancements from Amiga, Mac, Sparc, XT, AT, '386 and '486 users as well as from fellow ST owners. David ## ######## David Megginson meggin@epas.utoronto.ca Centre for Medieval Studies david@doe.utoronto.ca University of Toronto 39 Queen's Park Cr. E. ## ######## 0, unseen,, *** EOOH *** From: ajt@doc.ic.ac.uk (Tony Travis) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 6 Feb 92 02:17:13 GMT Organization: Department of Computing, Imperial College, University of London, UK. ast@cs.vu.nl (Andy Tanenbaum) writes: > Another option that seems to be totally forgotten here is buy UNIX or a > clone. If you just want to USE the system, instead of hacking on its > internals, you don't need source code. Coherent is only $99, and there > are various true UNIX systems with more features for more money. For the > true hacker, not having source code is fatal, but for people who just > want a UNIX system, there are many alternatives (albeit not free). Andy, I have followed the development of Minix since the first messages were posted to this group and I am now running 1.5.10 with Bruce Evans's patches for the 386. I 'just' want a Unix on my PC and I am not interested in hacking on its internals, but I *do* want the source code! An important principle underlying the success and popularity of Unix is the philosophy of building on the work of others. This philosophy relies upon the availability of the source code in order that it can be examined, modified and re-used in new software. Many years ago, I was in the happy position of being an AT&T Seventh Edition Unix source licencee

but, even then, I saw your decision to make the source of Minix available as liberation from the shackles of AT&T copyright!! I think you may sometimes forget that your 'hobby' has had a profound effect on the availability of 'personal' Unix (ie. affordable Unix) and that the 8086 PC I ran Minix 1.2 on actually cost me considerably more than my present 386/SX clone. Clearly, Minix _cannot_ be all things to all men, but I see the progress to 386 versions in much the same way that I see 68000 or other linear address space architectures: it is a good thing for people like me who use Minix and feel constrained by the segmented architecture of the PC version for applications. NOTHING you can say would convince me that I should use Coherent ... Tony -- ---------- ---------- Dr. A.J.Travis <ajt@uk.ac.sari.rri> | Rowett Research Institute, | Greenburn Road, Bucksburn, Aberdeen, | AB2 9SB. UK. tel 0224-712751 0, unseen,, *** EOOH *** From: richard@aiai.ed.ac.uk (Richard Tobin) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 7 Feb 92 14:58:22 GMT Organization: AIAI, University of Edinburgh, Scotland In article <12696@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes: >If you just want to USE the system, instead of hacking on its >internals, you don't need source code. Unfortunately hacking on the internals is just what many of us want the system for... You'll be rid of most of us when BSD-detox or GNU comes out, which should happen in the next few months (yeah, right). -- Richard -- Richard Tobin, AI Applications Institute, R.Tobin@ed.ac.uk Edinburgh University. 0, unseen,, *** EOOH *** From: comm121@unixg.ubc.ca (Louie) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 30 Jan 92 02:55:22 GMT Organization: University of British Columbia, Vancouver, B.C., Canada Nntp-Posting-Host: chilko.ucs.ubc.ca In <12595@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes: >But in all honesty, I would >suggest that people who want a **MODERN** "free" OS look around for a >microkernel-based, portable OS, like maybe GNU or something like that. There are really no other alternatives other than Linux for people like me who want a "free" OS. Considering that the majority of people who would use a "free" OS use the 386, portability is really not all that big of a concern. If I had a Sparc I would use Solaris. As it stands, I installed Linux with gcc,

emacs 18.57, kermit and all of the GNU utilities without any trouble at all. No need to apply patches. I just followed the installation instructions. I can't get an OS like this *anywhere* for the price to do my Computer Science homework. And it seems like network support and then X-Windows will be ported to Linux well before Minix. This is something that would be really useful. In my opinion, portability of standard Unix software is important also. I know that the design using a monolithic system is not as good as the microkernel. But for the short term future (And I know I won't/can't be uprading from my 386), Linux suits me perfectly. Philip Wu pwu@unixg.ubc.ca 0, unseen,, *** EOOH *** Newsgroups: comp.os.minix From: kevin@nuchat.sccsi.com (Kevin Brown) Subject: Re: LINUX is obsolete Organization: Hierarchical Date: Thu, 30 Jan 1992 01:36:43 GMT In article <12595@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes: > >I was in the U.S. for a couple of weeks, so I haven't commented much on >LINUX (not that I would have said much had I been around), but for what >it is worth, I have a couple of comments now. > >As most of you know, for me MINIX is a hobby, something that I do in the >evening when I get bored writing books and there are no major wars, >revolutions, or senate hearings being televised live on CNN. My real >job is a professor and researcher in the area of operating systems. > >As a result of my occupation, I think I know a bit about where operating >are going in the next decade or so. Two aspects stand out: > >1. MICROKERNEL VS MONOLITHIC SYSTEM > Most older operating systems are monolithic, that is, the whole operating > system is a single a.out file that runs in 'kernel mode.' This binary > contains the process management, memory management, file system and the > rest. Examples of such systems are UNIX, MS-DOS, VMS, MVS, OS/360, > MULTICS, and many more. > > The alternative is a microkernel-based system, in which most of the OS > runs as separate processes, mostly outside the kernel. They communicate > by message passing. The kernel's job is to handle the message passing, > interrupt handling, low-level process management, and possibly the I/O. > Examples of this design are the RC4000, Amoeba, Chorus, Mach, and the > not-yet-released Windows/NT. > > While I could go into a long story here about the relative merits of the > two designs, suffice it to say that among the

people who actually design > operating systems, the debate is essentially over. Microkernels have won. > The only real argument for monolithic systems was performance, and there > is now enough evidence showing that microkernel systems can be just as > fast as monolithic systems (e.g., Rick Rashid has published papers comparing > Mach 3.0 to monolithic systems) that it is now all over but the shoutin`. Of course, there are some things that are best left to the kernel, be it micro or monolithic. Like things that require playing with the process' stack, e.g. signal handling. Like memory allocation. Things like that. The microkernel design is probably a win, all in all, over a monolithic design, but it depends on what you put in the kernel and what you leave out. > MINIX is a microkernel-based system. The file system and memory management > are separate processes, running outside the kernel. The I/O drivers are > also separate processes (in the kernel, but only because the brain-dead > nature of the Intel CPUs makes that difficult to do otherwise). Minix is a microkernel design, of sorts. The problem is that it gives special priveleges to mm and fs, when there shouldn't be any (at least for fs). It also fails to integrate most of the functionality of mm in the kernel itself, and this makes things like signal handling and memory allocation *really* ugly. If you did these things in the kernel itself, then signal handling would be as simple as setting a virtual interrupt vector and causing the signalled process to receive that interrupt (with the complication that system calls might have to be terminated. Which means that a message would have to be sent to every process that is servicing the process' system call, if any. It's considerations like these that make the monolithic kernel design appealing). The *entire* system call interface in Minix needs to be rethought. As it stands right now, the file system is not just a file system, it's also a system-call server. That functionality needs to be separated out in order to facilitate a multiple file system architecture. Message passing is probably the right way to go about making the call and waiting for it, but the message should go to a system call server, not the file system itself. In order to handle all the special caveats of the Unix API, you end up writing a monolithic "kernel" even if you're using a microkernel base. You end up with something called a "server", and an example is the BSD server that runs under Mach. And, in any case, the message-passing in Minix

needs to be completely redone. As it is, it's a kludge. I've been giving this some thought, but I haven't had time to do anything with what I've thought of so far. Suffice it to say that the proper way to do message-passing is probably with message ports (both public and private), with the various visible parts of the operating system having public message ports. Chances are, that ends up being the system call server only, though this will, of course, depend on the goals of the design. > LINUX is > a monolithic style system. This is a giant step back into the 1970s. > That is like taking an existing, working C program and rewriting it in > BASIC. To me, writing a monolithic system in 1991 is a truly poor idea. Depends on the design criteria, as you should know. If your goal is to design a Unix workalike that is relatively simple and relatively small, then a monolithic design is probably the right approach for the job, because unless you're designing for really backwards hardware, the problems of things like interrupted system calls, memory allocation within the kernel (so you don't have to statically allocate *everything* in your OS), signal handling, etc. all go away (or are at least minimized) if you use a monolithic design. If you want the ability to bring up and take down file systems, add and remove device drivers, etc., all at runtime, then a microkernel approach is the right solution. Frankly, I happen to like the idea of removable device drivers and such, so I tend to favor the microkernel approach as a general rule. >2. PORTABILITY > Once upon a time there was the 4004 CPU. When it grew up it became an > 8008. Then it underwent plastic surgery and became the 8080. It begat > the 8086, which begat the 8088, which begat the 80286, which begat the > 80386, which begat the 80486, and so on unto the N-th generation. In > the meantime, RISC chips happened, and some of them are running at over > 100 MIPS. Speeds of 200 MIPS and more are likely in the coming years. > These things are not going to suddenly vanish. What is going to happen > is that they will gradually take over from the 80x86 line. They will > run old MS-DOS programs by interpreting the 80386 in software. (I even > wrote my own IBM PC simulator in C, which you can get by FTP from > ftp.cs.vu.nl = 192.31.231.42 in dir minix/simulator.) I think it is a > gross error to design an OS for any specific architecture, since that is > not going to be around all that long. Again, look at the design criteria. If portability isn't an issue,

then why worry about it? While LINUX suffers from lack of portability, portability was obviously never much of a consideration for its author, who explicitly stated that it was written as an exercise in learning about the 386 architecture. And, in any case, while MINIX is portable in the sense that most of the code can be ported to other platforms, it *still* suffers from the limitations of the original target machine that drove the walk down the design decision tree. The message passing is a kludge because the 8088 is slow. The kernel doesn't do memory allocation (thus not allowing FS and the drivers to get away with using a malloc library or some such, and thus causing everyone to have to statically allocate everything), probably due to some other limitation of the 8088. The very idea of using "clicks" is obviously the result of the segmented architecture of the 8088. The file system size is too limited (theoretically fixed in 1.6, but now you have *two* file system formats to contend with. If having the file system as a separate process is such a big win, then why don't we have two file system servers, eh? Why simply extend the existing Minix file system instead of implementing BSD's FFS or some other high-performance file system? It's not that I'm greedy or anything... :-). > MINIX was designed to be reasonably portable, and has been ported from the > Intel line to the 680x0 (Atari, Amiga, Macintosh), SPARC, and NS32016. > LINUX is tied fairly closely to the 80x86. Not the way to go. All in all, I tend to agree. >Don`t get me wrong, I am not unhappy with LINUX. It will get all the people >who want to turn MINIX in BSD UNIX off my back. But in all honesty, I would >suggest that people who want a **MODERN** "free" OS look around for a >microkernel-based, portable OS, like maybe GNU or something like that. Yeah, right. Point me someplace where I can get a free "modern" OS and I'll gladly investigate. But the GNU OS is currently vaporware, and as far as I'm concerned it will be for a LOOOOONG time to come. Any other players? BSD 4.4 is a monolithic architecture, so by your definition it's out. Mach is free, but the BSD server isn't (AT&T code, you know), and in any case, isn't the BSD server something you'd consider to be a monolithic design??? Really. Why do you think LINUX is as popular as it is? The answer is simple, of course: because it's the *only* free Unix workalike OS in existence. BSD doesn't qualify (yet). Minix doesn't qualify. XINU isn't even in the running.

GNU's OS is vaporware, and probably will be for a long time, so *by definition* it's not in the running. Any other players? I haven't heard of any... >Andy Tanenbaum (ast@cs.vu.nl) Minix is an excellent piece of work. A good starting point for anyone who wants to learn about operating systems. But it needs rewriting to make it truly elegant and functional. As it is, there are too many kludges and hacks (e.g., the message passing). Kevin Brown 0, unseen,, *** EOOH *** From: aduncan@rhea.trl.OZ.AU (Allan Duncan) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 2 Feb 92 22:06:26 GMT Organization: Telecom Research Labs, Melbourne, Australia From article <1992Jan30.013643.3248@menudo.uh.edu>, by kevin@nuchat.sccsi.com (Kevin Brown): > The *entire* system call interface in Minix needs to be rethought. As it > stands right now, the file system is not just a file system, it's also a > system-call server. That functionality needs to be separated out in order > to facilitate a multiple file system architecture. Message passing is > probably the right way to go about making the call and waiting for it, but > the message should go to a system call server, not the file system itself. > > In order to handle all the special caveats of the Unix API, you end up writing > a monolithic "kernel" even if you're using a microkernel base. You end up > with something called a "server", and an example is the BSD server that runs > under Mach. > > And, in any case, the message-passing in Minix needs to be completely redone. > As it is, it's a kludge. I've been giving this some thought, but I haven't > had time to do anything with what I've thought of so far. Suffice it to say > that the proper way to do message-passing is probably with message ports > (both public and private), with the various visible parts of the operating > system having public message ports. Chances are, that ends up being the > system call server only, though this will, of course, depend on the goals > of the design. It gets to sound more and more like Tripos and the Amiga :-) Allan Duncan ACSnet aduncan@trl.oz (+613) 541 6708 Internet aduncan@trl.oz.au UUCP {uunet,hplabs,ukc}!munnari!trl.oz.au!aduncan Telecom Research Labs, PO Box 249, Clayton, Victoria, 3168, Australia. 0, unseen,, *** EOOH *** From: kevin@taronga.taronga.com (Kevin Brown) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 4 Feb 92 08:28:08

GMT Organization: University of Houston In article <1992Feb2.220626.25197@trl.oz.au> aduncan@rhea.trl.OZ.AU (Allan Duncan) writes: >From article <1992Jan30.013643.3248@menudo.uh.edu>, by kevin@nuchat.sccsi.com (Kevin Brown): > >> The *entire* system call interface in Minix needs to be rethought. As it >> stands right now, the file system is not just a file system, it's also a >> system-call server. That functionality needs to be separated out in order >> to facilitate a multiple file system architecture. Message passing is >> probably the right way to go about making the call and waiting for it, but >> the message should go to a system call server, not the file system itself. >> >> In order to handle all the special caveats of the Unix API, you end up writing >> a monolithic "kernel" even if you're using a microkernel base. You end up >> with something called a "server", and an example is the BSD server that runs >> under Mach. >> >> And, in any case, the message-passing in Minix needs to be completely redone. >> As it is, it's a kludge. I've been giving this some thought, but I haven't >> had time to do anything with what I've thought of so far. Suffice it to say >> that the proper way to do message-passing is probably with message ports >> (both public and private), with the various visible parts of the operating >> system having public message ports. Chances are, that ends up being the >> system call server only, though this will, of course, depend on the goals >> of the design. > >It gets to sound more and more like Tripos and the Amiga :-) There's no question that many of my ideas spring from the architecture of the Amiga's operating system. It's pretty impressive to see a message-passing, multitasking operating system that operates as fast as the Amiga's OS does on hardware that slow. They did a lot of things right. There are some ideas that, I think, are my own. Or, at least, that I've developed independently. For example, if you have a message-passing system that includes the option to transfer message memory ownership to the target process, then it naturally follows that you can globally optimize the use of your block cache by making your block cache global with respect to *all* filesystems. The filesystem code requests blocks from the block cache manager and tells the block cache manager what device driver to call and what parameters to send it when flushing the block. The block cache manager replies with a message that is the size of a block (or, if

you wish to allocate several at a time, several blocks). Since ownership is transferred as a result of passing the message, the block cache manager can allocate the memory itself, optionally flushing as many blocks as it needs in order to free up enough to send to the caller. The block cache manager is, of course, a user process. If the filesystem code is written right, you can kill the block cache manager in order to disable the block cache. The filesystem will simply do its thing unbuffered. Makes for a slow system, but at least you can do it. You can also change the behavior of the buffer cache by sending control messages to the cache manager. Can you say "tunable parameters"? :-) You could also accomplish this with some sort of shared memory, but this would require semaphore control of the allocation list. You'd also have to figure out a way to flush bits of the cache when needed (easy to do if you're a monolithic kernel, but I'm referring to a microkernel) without colliding with another process writing into the block. Semaphore control of the individual blocks as well? >Allan Duncan ACSnet aduncan@trl.oz >(+613) 541 6708 Internet aduncan@trl.oz.au > UUCP {uunet,hplabs,ukc}!munnari!trl.oz.au!aduncan >Telecom Research Labs, PO Box 249, Clayton, Victoria, 3168, Australia.

Kevin Brown 0, unseen,, *** EOOH *** From: dgraham@bmers30.bnr.ca (Douglas Graham) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 1 Feb 92 00:26:30 GMT Organization: Bell-Northern Research, Ottawa, Canada In article <12595@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes: > While I could go into a long story here about the relative merits of the > two designs, suffice it to say that among the people who actually design > operating systems, the debate is essentially over. Microkernels have won. Can you recommend any (unbiased) literature that points out the strengths and weaknesses of the two approaches? I'm sure that there is something to be said for the microkernel approach, but I wonder how closely Minix resembles the other systems that use it. Sure, Minix uses lots of tasks and messages, but there must be more to a microkernel architecture than that. I suspect that the Minix code is not split optimally into tasks. > The only real argument for monolithic systems was performance, and there > is now enough evidence showing that microkernel systems can be just as > fast as monolithic systems (e.g.,

Rick Rashid has published papers comparing > Mach 3.0 to monolithic systems) that it is now all over but the shoutin`. My main complaint with Minix is not it's performance. It is that adding features is a royal pain -- something that I presume a microkernel architecure is supposed to alleviate. > MINIX is a microkernel-based system. Is there a consensus on this? > LINUX is > a monolithic style system. This is a giant step back into the 1970s. > That is like taking an existing, working C program and rewriting it in > BASIC. To me, writing a monolithic system in 1991 is a truly poor idea. This is a fine assertion, but I've yet to see any rationale for it. Linux is only about 12000 lines of code I think. I don't see how splitting that into tasks and blasting messages around would improve it. >Don`t get me wrong, I am not unhappy with LINUX. It will get all the people >who want to turn MINIX in BSD UNIX off my back. But in all honesty, I would >suggest that people who want a **MODERN** "free" OS look around for a >microkernel-based, portable OS, like maybe GNU or something like that. Well, there are no other choices that I'm aware of at the moment. But when GNU OS comes out, I'll very likely jump ship again. I sense that you *are* somewhat unhappy about Linux (and that surprises me somewhat). I would guess that the reason so many people embraced it, is because it offers more features. Your approach to people requesting features in Minix, has generally been to tell them that they didn't really want that feature anyway. I submit that the exodus in the direction of Linux proves you wrong. Disclaimer: I had nothing to do with Linux development. I just find it an easier system to understand than Minix. -- Doug Graham dgraham@bnr.ca My opinions are my own. 0, unseen,, *** EOOH *** From: peter@ferranti.com (peter da silva) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 3 Feb 92 17:40:06 GMT Organization: Xenix Support, FICC In article <1992Feb01.002630.14861@bmerh2.bnr.ca> dgraham@bmers30.bnr.ca (Douglas Graham) writes: > Minix resembles the other systems that use it. Sure, Minix uses lots > of tasks and messages, but there must be more to a microkernel architecture > than that. I suspect that the Minix code is not split optimally into tasks. Definitely. Minix shows you how a microkernel works, but it sure doesn't show you why you would use one. A couple of years ago I

brought this up with Andy, and his response indicated that he was himself not convinced of the superiority of the microkernel design at the time. He said (as near as I can recall... this is a paraphrase) that a message passing design was inherently slower than a monolithic one... which was news to me: I had (and still have) a PC that was MUCH more responsive than any UNIX box I ever touched using a message-passing design. > > MINIX is a microkernel-based system. > Is there a consensus on this? Yes, it's not a well-factored one, and there's no API to the microkernel interface, but it's a microkernel design. -- -- Peter da Silva, Ferranti International Controls Corporation -- Sugar Land, TX 77487-5012; +1 713 274 5180 -- "Have you hugged your wolf today?" 0, unseen,, *** EOOH *** From: hedrick@klinzhai.rutgers.edu (Charles Hedrick) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 1 Feb 92 00:27:04 GMT Organization: Rutgers Univ., New Brunswick, N.J. The history of software shows that availability wins out over technical quality every time. That's Linux' major advantage. It's a small 386-based system that's fairly compatible with generic Unix, and is freely available. I dropped out of the Minix community a couple of years ago when it became clear that (1) Minix was not going to take advantage of anything beyond the 8086 anytime in the near future, and (2) the licensing -- while amazingly friendly -- still made it hard for people who were interested in producing a 386 version. Several people apparently did nice work for the 386. But all they could distribute were diffs. This made bringing up a 386 system a job that isn't practical for a new user, and in fact I wasn't sure I wanted to do it. I apologize if things have changed in the last couple of years. If it's now possible to get a 386 version in a form that's ready to run, the community has developed a way to share Minix source, and bringing up normal Unix programs has become easier in the interim, then I'm willing to reconsider Minix. I do like its design. It's possible that Linux will be overtaken by Gnu or a free BSD. However, if the Gnu OS follows the example of all other Gnu software, it will require a system with 128MB of memory and a 1GB disk to use. There will still be room for a small system. My ideal OS would be 4.4 BSD. But 4.4's release date has a history of extreme slippage. With most of their staff moving to BSDI, it's hard to believe that this situation is going to be improved. For my

own personal use, the BSDI system will probably be great. But even their very attractive pricing is likely to be too much for most of our students, and even though users can get source from them, the fact that some of it is proprietary will again mean that you can't just put altered code out for public FTP. At any rate, Linux exists, and the rest of these alternatives are vapor. 0, unseen,, *** EOOH *** From: nhc@cbnewsj.cb.att.com (n.h.chandler) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Summary: How can I get a copy of Linux? Date: 2 Feb 92 01:38:51 GMT Organization: AT&T Bell Laboratories I have been following the Minix/Linux discussion. How can I get a copy of Linux? Neville H. Chandler cbnewsj!nhc@att.com 0, unseen,, *** EOOH *** From: tytso@athena.mit.edu (Theodore Y. Ts'o) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 31 Jan 92 21:40:23 GMT Organization: Massachusetts Institute of Technology In-Reply-To: ast@cs.vu.nl's message of 29 Jan 92 12: 12:50 GMT Nntp-Posting-Host: sos.mit.edu >From: ast@cs.vu.nl (Andy Tanenbaum) >ftp.cs.vu.nl = 192.31.231.42 in dir minix/simulator.) I think it is a >gross error to design an OS for any specific architecture, since that is >not going to be around all that long. It's not your fault for believing that Linux is tied to the 80386 architecture, since many Linux supporters (including Linus himself) have made the this statement. However, the amount of 80386-specific code is probably not much more than what is in a Minix implementation, and there is certainly a lot less 80386 specific code in Linux than here is Vax-specific code in BSD 4.3. Granted, the port to other architectures hasn't been done yet. But if I were going to bring up a Unix-like system on a new architecture, I'd probably start with Linux rather than Minix, simply because I want to have some control over what I can do with the resulting system when I'm done with it. Yes, I'd have to rewrite large portions of the VM and device driver layers --- but I'd have to do that with any other OS. Maybe it would be a little bit harder than it would to port Minix to the new architecture; but this would probably be only true for the first architecture that we ported Linux to. >While I could go into a long story here about the relative merits of the >two designs, suffice it to say that among the people who actually design >operating systems, the debate is essentially over. Microkernels have won. >The only real argument for monolithic

systems was performance, and there >is now enough evidence showing that microkernel systems can be just as >fast as monolithic systems (e.g., Rick Rashid has published papers comparing >Mach 3.0 to monolithic systems) that it is now all over but the shoutin`. This is not necessarily the case; I think you're painting a much more black and white view of the universe than necessarily exists. I refer you to such papers as Brent Welsh's (welch@parc.xerox.com) "The Filsystem Belongs in the Kernel" paper, where in he argues that the filesystem is a mature enough abstraction that it should live in the kernel, not outside of it as it would in a strict microkernel design. There also several people who have been concerned about the speed of OSF/1 Mach when compared with monolithic systems; in particular, the nubmer of context switches required to handle network traffic, and networked filesystems in particular. I am aware of the benefits of a micro kernel approach. However, the fact remains that Linux is here, and GNU isn't --- and people have been working on Hurd for a lot longer than Linus has been working on Linux. Minix doesn't count because it's not free. :-) I suspect that the balance of micro kernels versus monolithic kernels depend on what you're doing. If you're interested in doing research, it is obviously much easier to rip out and replace modules in a micro kernel, and since only researchers write papers about operating systems, ipso facto micro kernels must be the right approach. However, I do know a lot of people who are not researchers, but who are rather practical kernel programmers, who have a lot of concerns over the cost of copying and the cost of context switches which are incurred in a micro kernel. By the way, I don't buy your arguments that you don't need a multi-threaded filesystem on a single user system. Once you bring up a windowing system, and have a compile going in one window, a news reader in another window, and UUCP/C News going in the background, you want good filesystem performance, even on a single-user system. Maybe to a theorist it's an unnecessary optimization and a (to use your words) "performance hack", but I'm interested in a Real operating system --- not a research toy. =-= Theodore Ts'o

bloom-beacon!mit-athena!tytso 308 High St., Medford, MA 02155
tytso@athena.mit.edu Everybody's playing the game, but nobody's rules are the same! 0, unseen,, *** EOOH *** From:

peter@ferranti.com (peter da silva) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 3 Feb 92 17:32:54 GMT Organization: Xenix Support, FICC In article <TYTSO.92Jan31164013@SOS.mit.edu> tytso@athena.mit.edu (Theodore Y. Ts'o) writes: > This is not necessarily the case; I think you're painting a much more > black and white view of the universe than necessarily exists. I refer > you to such papers as Brent Welsh's (welch@parc.xerox.com) "The > Filsystem Belongs in the Kernel" paper, where in he argues that the > filesystem is a mature enough abstraction that it should live in the > kernel, not outside of it as it would in a strict microkernel design. What does "a mature enough abstraction" mean, here? Things don't move into the kernel simply because they're now considered safe and stable enough, but because they're too inefficient when they're outside it or they lose functionality by being outside it, and there's no easy fix. The Amiga operating system certainly benefits from having a file system outside the kernel. There are dozens of file systems, many of them written by hobbyists, available. Ideas like "assigned paths" can be played with in the file system without breaking stuff. All these file systems have a common interface and so look to the application as part of the operating system, but just because something is on the other side of the API doesn't mean it is, or belongs, in the kernel. > There also several people who have been concerned about the speed of > OSF/1 Mach when compared with monolithic systems; in particular, the > nubmer of context switches required to handle network traffic, and > networked filesystems in particular. If this is because the networking was moved out of the kernel, I consider it a price well worth paying. Having networking code in the kernel is the source of many subtle bugs in networks. Just for something that bit us, what happens if you need to get to the upper level driver before you can acknowledge a packet, but the process that you need to run is hung up in the tty driver waiting for a ^Q? Something *I* would have expected to find in the kernel before now, yet isn't, is windowing systems. With a microkernel (and the associated lower *cost* of a context switch) you can get much of the advantages of a kernel window system without paying the cost in complexity. -- -- Peter da Silva, Ferranti International Controls Corporation -- Sugar Land, TX 77487-5012; +1 713 274 5180 -- "Have you hugged your

wolf today?" 0, unseen,, *** EOOH *** From: joe@jshark.rn.com Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 31 Jan 92 13:21:44 GMT Organization: a blip of entropy In article <12595@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes: > > MINIX was designed to be reasonably portable, and has been ported from the > Intel line to the 680x0 (Atari, Amiga, Macintosh), SPARC, and NS32016. > LINUX is tied fairly closely to the 80x86. Not the way to go. If you looked at the source instead of believing the author, you'd realise this is not true! He's replaced 'fubyte' by a routine which explicitly uses a segment register - but that could be easily changed. Similarly, apart from a couple of places which assume the '386 MMU, a couple of macros to hide the exact page sizes etc would make porting trivial. Using '386 TSS's makes the code simpler, but the VAX and WE32000 have similar structures. As he's already admitted, a bit of planning would have the the system neater, but merely putting '386 assembler around isn't a crime! And with all due respect: - the Book didn't make an issue of portability (apart from a few "#ifdef M8088"s) - by the time it was released, Minix had come to depend on several 8086 "features" that caused uproar from the 68000 users. >Andy Tanenbaum (ast@cs.vu.nl) joe. -- joe@jshark.rn.com 0, unseen,, *** EOOH *** From: joe@jshark.rn.com Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 2 Feb 92 23:59:12 GMT Organization: Jshark Communications In article <12595@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes: > >I was in the U.S. for a couple of weeks, so I haven't commented much on >LINUX (not that I would have said much had I been around), but for what >it is worth, I have a couple of comments now. Maybe keepng quiet would have been best. >1. MICROKERNEL VS MONOLITHIC SYSTEM > > While I could go into a long story here about the relative merits of the > two designs, suffice it to say that among the people who actually design > operating systems, the debate is essentially over. No, MS-DOS won. Sad, but there you are. 60 million: Next It would be churlish to point out that MS-DOS has loadable device drivers and that VMS is now (basically)a set of loadable service modules and drivers. "Microkernel" was the buzz-word of last year, so Minix is a microkernel. "Object-oriented" is this years, so Minix is object-oriented - right? joe. ---- joe@jshark.rn.com

uunet!nstar!jshark!joe I'm a mutated .sig virus, I got this from Henry Spencer's: "As a user, I'll take speed over features anyday" - A Tanenbaum 0, unseen,, *** EOOH *** From: entropy@wintermute.WPI.EDU (Lawrence C. Foard) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 5 Feb 92 14:56:30 GMT Organization: Worcester Polytechnic Institute Nntp-Posting-Host: wintermute.wpi.edu In article <12595@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes: >Don`t get me wrong, I am not unhappy with LINUX. It will get all the people >who want to turn MINIX in BSD UNIX off my back. But in all honesty, I would >suggest that people who want a **MODERN** "free" OS look around for a >microkernel-based, portable OS, like maybe GNU or something like that. I believe you have some valid points, although I am not sure that a microkernel is necessarily better. It might make more sense to allow some combination of the two. As part of the IPC code I'm writting for Linux I am going to include code that will allow device drivers and file systems to run as user processes. These will be significantly slower though, and I believe it would be a mistake to move everything outside the kernel (TCP/IP will be internal). Actually my main problem with OS theorists is that they have never tested there ideas! None of these ideas (with a partial exception for MACH) has ever seen the light of day. 32 bit home computers have been available for almost a decade and Linus was the first person to ever write a working OS for them that can be used without paying AT&T $100,000. A piece of software in hand is worth ten pieces of vaporware, OS theorists are quick to jump all over an OS but they are unwilling to ever provide an alternative. The general consensus that Micro kernels is the way to go means nothing when a real application has never even run on one. The release of Linux is allowing me to try some ideas I've been wanting to experment with for years, but I have never had the opportunity to work with source code for a functioning OS. -- Disclaimer: Opinions are based on logic rather than biblical "fact". ------ Hackers do it for fun. | First they came for the drug users, I said \ / "Profesionals" do it for money. | nothing, then they came for hackers, \ / Managers have others do it for them. | I said nothing... STOP W.O.D. \/ 0, unseen,, *** EOOH *** From: ast@cs.vu.nl (Andy Tanenbaum) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 5 Feb 92 23:33:23

GMT Organization: Fac. Wiskunde & Informatica, Vrije Universiteit, Amsterdam In article <1992Feb5.145630.759@wpi.WPI.EDU> entropy@wintermute.WPI.EDU (Lawrence C. Foard) writes: >Actually my main problem with OS theorists is that they have never tested >there ideas! I'm mortally insulted. I AM NOT A THEORIST. Ask anybody who was at our department meeting yesterday (in joke). Actually, these ideas have been very well tested in practice. OSF is betting its whole business on a microkernel (Mach 3.0). USL is betting its business on another one (Chorus). Both of these run lots of software, and both have been extensively compared to monolithic systems. Amoeba has been fully implemented and tested for a number of applications. QNX is a microkernel based system, and someone just told me the installed base is 200,000 systems. Microkernels are not a pipe dream. They represent proven technology. The Mach guys wrote a paper called "UNIX as an application program." It was by Golub et al., in the Summer 1990 USENIX conference. The Chorus people also have a technical report on microkernel performance, and I coauthored another paper on the subject, which I mentioned yesterday (Dec. 1991 Computing Systems). Check them out. Andy Tanenbaum (ast@cs.vu.nl) 0, unseen,, *** EOOH *** From: wolff@neuron.et.tudelft.nl (Rogier Wolff) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 8 Feb 92 09:13:39 GMT Organization: Delft University of Technology, Dept. of Electrical Engineering Nntp-Posting-Host: neuron.et.tudelft.nl ast@cs.vu.nl (Andy Tanenbaum) writes: >In article <1992Feb5.145630.759@wpi.WPI.EDU> entropy@wintermute.WPI.EDU (Lawrence C. Foard) writes: >>Actually my main problem with OS theorists is that they have never tested >>there ideas! >I'm mortally insulted. I AM NOT A THEORIST. Ask anybody who was at our >department meeting yesterday (in joke). >Actually, these ideas have been very well tested in practice. The problem is that to really do an unbiased test you would need two *identical* teams, and ask them to make two OS's, for the same destination machine, one using a microkernel architecture, and the other using the monolithic approach. This is in practice not feasable and the publications on the subject can only shout: "look: I've got a good performance using a microkernel", "we've got very good

performance using a monolithic aproach" or "it only took us X months to implement this OS" If people did benchmark their OS's they wrote the OS for one architecture, and adapted it to test the other. This adaptation will naturally degrade performance, and show that the designers were right in the first place. Anyway, anybody have an opinion about the fact that code for printf is included three times in the Minix OS when it runs (once in the kernel, MM and FS)

Roger -- If the opposite of "pro" is "con", what is the opposite of "progress"? (stolen from kadokev@iitvax ==? technews@iitmax.iit.edu) EMail: wolff@duteca.et.tudelft.nl ** Tel +31-15-783644 or +31-15-142371 0, unseen,, *** EOOH *** From: meggin@epas.utoronto.ca (David Megginson) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 8 Feb 92 15:04:31 GMT Organization: University of Toronto - EPAS Nntp-Posting-Host: epas.utoronto.ca In article <1992Feb08.091339.16121@donau.et.tudelft.nl> wolff@neuron.et.tudelft.nl (Rogier Wolff) writes: >Anyway, anybody have an opinion about the fact that code for printf >is included three times in the Minix OS when it runs (once in the >kernel, MM and FS) Back in the yore days, this might have been a problem. I remember when every program, even wordprocessors, had to be written in assembler to squeeze them down to the smallest size possible for a 64K system. One of the reasons WordPerfect is such a mess today is that it was written in assembler instead of C. Now, even the small systems which Minix runs have at least 640K, so a few wasted bytes are not so much of a problem. Why not write Linux in 80386 assembler? It would be smaller and even faster. And don't forget to code inline as much as possible, to avoid the crippling overhead of function calls. And leave out comments, because they waste disk space. David ## ######## David Megginson meggin@epas.utoronto.ca Centre for Medieval Studies david@doe.utoronto.ca University of Toronto 39 Queen's Park Cr. E. ## ######## 0, unseen,, *** EOOH *** From: cs2@doc.ic.ac.uk (Christopher Stuart) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 11 Feb 92 13:52:57 GMT Organization: Department

of Computing, Imperial College, University of London, UK. Nntp-Posting-Host: oak10.doc.ic.ac.uk Article 18297 of comp.os.minix: Path: icdoc!uknet!mcsun!uunet!cis.ohio-state.edu!rutgers!news-server.csri.toronto.edu!utgpu!news-server.ecf!epas!meggin From: meggin@epas.utoronto.ca (David Megginson) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Message-ID: <1992Feb8.150431.11030@epas.toronto.edu> Date: 8 Feb 92 15:04:31 GMT References: <1992Feb5.145630.759@wpi.WPI.EDU> <12747@star.cs.vu.nl> <1992Feb08.091339.16121@donau.et.tudelft.nl> Sender: news@epas.toronto.edu (USENET) Organization: University of Toronto - EPAS Lines: 28 Nntp-Posting-Host: epas.utoronto.ca In article <1992Feb08.091339.16121@donau.et.tudelft.nl> wolff@neuron.et.tudelft.nl (Rogier Wolff) writes: >Anyway, anybody have an opinion about the fact that code for printf >is included three times in the Minix OS when it runs (once in the >kernel, MM and FS) Back in the yore days, this might have been a problem. I remember when every program, even wordprocessors, had to be written in assembler to squeeze them down to the smallest size possible for a 64K system. One of the reasons WordPerfect is such a mess today is that it was written in assembler instead of C. Now, even the small systems which Minix runs have at least 640K, so a few wasted bytes are not so much of a problem. Why not write Linux in 80386 assembler? It would be smaller and even faster. And don't forget to code inline as much as possible, to avoid the crippling overhead of function calls. And leave out comments, because they waste disk space. David ## ######## David Megginson meggin@epas.utoronto.ca Centre for Medieval Studies david@doe.utoronto.ca University of Toronto 39 Queen's Park Cr. E. ## ######## -- /*-- ----------*/ /* Christopher Stuart: cs2@doc.ic.ac.uk */ /* Dept. Computing, Imperial College, London. */ /*-- ----------*/ 0, unseen,, *** EOOH *** From: peter@ferranti.com (peter da silva) Newsgroups: comp.os.minix Subject: Re: LINUX is

obsolete Date: 10 Feb 92 03:10:00 GMT Organization: Xenix Support, FICC In article <1992Feb08.091339.16121@donau.et.tudelft.nl> wolff@neuron.et.tudelft.nl (Rogier Wolff) writes: > The problem is that to really do an unbiased test you would need two > *identical* teams, and ask them to make two OS's [...] No, you don't. I don't think there's any question that a macrokernel is very easy to get decent performance out of. Where the microkernel design has a major advantage is in flexibility. Adding stuff to a macrokernel is fairly complex and quickly becomes pretty gross. Look at BSD or System V for examples. Adding stuff to a well designed microkernel is VERY easy. Sometimes you don't want to compare oranges and oranges. Sometimes you want to compare concentrated orange juice with fresh-squeezed. Fresh-squeezed takes longer, but it's worth it. Plus, with a microkernel you can get much better context switching between microtasks than macro processes. So you can do stuff in separate processes that would be out of the question in a macrokernel, and avoid nonsense like the myriad inconsistencies in NFS. > anyone have an opinion about why the code for printf > is included three times in the Minix OS when it runs (once in the > kernel, MM and FS) Anyone have an opinion why the code for printf is included only once in AmigaOS (even though the AmigaOS 2.04 "kernel" is actually a dozen or more separate processes)? Minix is a poor technology demonstrator for microkernels. Which is OK, since it wasn't supposed to be one. -- -- Peter da Silva, Ferranti International Controls Corporation -- Sugar Land, TX 77487-5012; +1 713 274 5180 -- "Have you hugged your wolf today?" 0, unseen,, *** EOOH *** Newsgroups: comp.os.minix From: peter@ferranti.com (peter da silva) Subject: Re: LINUX is obsolete Organization: Xenix Support, FICC Date: Thu, 6 Feb 1992 16:02:47 GMT In article <12747@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes: > QNX is a microkernel > based system, and someone just told me the installed base is 200,000 systems. Oh yes, while I'm on the subject... there are over 3 million Amigas out there, which means that there are more of them than any UNIX vendor has shipped, and probably more than all UNIX systems combined. -- -- Peter da Silva, Ferranti International Controls Corporation -- Sugar Land, TX 77487-5012; +1 713 274 5180 -- "Have you hugged your wolf today?" 0, unseen,, *** EOOH *** From: tim@maths.tcd.ie (Timothy Murphy) Newsgroups:

comp.os.minix Subject: Re: LINUX is obsolete Date: 6 Feb 92 11:14:59 GMT Organization: Dept. of Maths, Trinity College, Dublin, Ireland. Nntp-Posting-Host: salmon In <1992Feb5.145630.759@wpi.WPI.EDU> entropy@wintermute.WPI.EDU (Lawrence C. Foard) writes: >32 bit home computers have been available for almost a >decade and Linus was the first person to ever write a working OS for them >that can be used without paying AT&T $100,000. A piece of software in hand is >worth ten pieces of vaporware, OS theorists are quick to jump all over an OS >but they are unwilling to ever provide an alternative. Surely Bruce Evans' 386-Minix preceded Linux? (Diffs for PC-Minix -> 386-Minix available from archive-server@plains.nodak.edu in the directory Minix/oz) -- Timothy Murphy e-mail: tim@maths.tcd.ie tel: +353-1-2842366 s-mail: School of Mathematics, Trinity College, Dublin 2, Ireland 0, unseen,, *** EOOH *** From: ts@cup.portal.com (Tim W Smith) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 7 Feb 92 02:09:23 GMT Organization: The Portal System (TM) > Actually my main problem with OS theorists is that they have never tested > there ideas! None of these ideas (with a partial exception for MACH) has ever > seen the light of day. 32 bit home computers have been available for almost a > decade and Linus was the first person to ever write a working OS for them > that can be used without paying AT&T $100,000. A piece of software in hand is How about Netware 386 from Novell? It seems to work.

Tim Smith 0, unseen,, *** EOOH *** Newsgroups: comp.os.minix From: peter@ferranti.com (peter da silva) Subject: Re: LINUX is obsolete Organization: Xenix Support, FICC Date: Thu, 6 Feb 1992 16:00:22 GMT In article <1992Feb5.145630.759@wpi.WPI.EDU> entropy@wintermute.WPI.EDU (Lawrence C. Foard) writes: > Actually my main problem with OS theorists is that they have never tested > there ideas! I beg to differ... there are many microkernel operating systems out there for everything from an 8088 (QNX) up to large research systems. > None of these ideas (with a partial exception for MACH) has ever > seen the light of day. 32 bit home computers have been available for almost a > decade and Linus was the first person to ever write a working OS for them > that can be used without

paying AT&T $100,000. I must have been imagining AmigaOS, then. I've been using a figment of my imagination for the past 6 years. AmigaOS is a microkernel message-passing design, with better response time and performance than any other readily available PC operating system: including MINIX, OS/2, Windows, MacOS, Linux, UNIX, and *certainly* MS-DOS. The microkernel design has proven invaluable. Things like new file systems that are normally available only from the vendor are hobbyist products on the Amiga. Device drivers are simply shared libraries and tasks with specific entry points and message ports. So are file systems, the window system, and so on. It's a WONDERFUL design, and validates everything that people have been saying about microkernels. Yes, it takes more work to get them off the ground than a coroutine based macrokernel like UNIX, but the versatility pays you back many times over. I really wish Andy would do a new MINIX based on what has been learned since the first release. The factoring of responsibilities in MINIX is fairly poor, but the basic concept is good. > The general consensus that Micro kernels is the way to go means nothing when > a real application has never even run on one. I'm dreaming again. I sure throught Deluxe Paint, Sculpt 3d, Photon Paint, Manx C, Manx SDB, Perfect Sound, Videoscape 3d, and the other programs I bought for my Amiga were "real". I'll have to send the damn things back now, I guess. The availability of Linux is great. I'm delighted it exists. I'm sure that the macrokernel design is one reason it has been implemented so fast, and this is a valid reason to use macrokernels. BUT... this doesn't mean that microkernels are inherently slow, or simply research toys. -- -- Peter da Silva, Ferranti International Controls Corporation -- Sugar Land, TX 77487-5012; +1 713 274 5180 -- "Have you hugged your wolf today?" 0, unseen,, *** EOOH *** From: dsmythe@netcom.COM (Dave Smythe) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 10 Feb 92 07:08:22 GMT Organization: Netcom - Online Communication Services (408 241-9760 guest) In article <1992Feb5.145630.759@wpi.WPI.EDU> entropy@wintermute.WPI.EDU (Lawrence C. Foard) writes: >Actually my main problem with OS theorists is that they have never tested >there ideas! None of these ideas (with a partial exception for MACH) has ever >seen the light of day. David Cheriton (Prof. at Stanford, and

author of the V system) said something similar to this in a class in distributed systems. Paraphrased: "There are two kinds of researchers: those that have implemented something and those that have not. The latter will tell you that there are 142 ways of doing things and that there isn't consensus on which is best. The former will simply tell you that 141 of them don't work." He really rips on the OSI-philes as well, for a similar reason. The Internet protocols are adapted only after having been in use for a period of time, preventing things from getting standardized that will never be implementable in a reasonable fashion. OSI adherents, on the other hand, seem intent on standardizing everything possible, including "escapes" from the standard, before a reasonable reference implementation exists. Consequently, you see obsolete ideas immortalized, such as sub-byte-level data field packing, which makes good performance difficult when your computer is drinking from a 10+ Gbs fire-hose :-). Just my $.02 D --
========== Dave Smythe N6XLP dsmythe@netcom.com (also dsmythe@cs.stanford.edu) 0, unseen,, *** EOOH *** From: mitchell@mdd.comm.mot.com (Bill Mitchell) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 10 Feb 92 15:03:51 GMT Reply-To: mitchell@mdd.comm.mot.com (Bill Mitchell) Organization: Motorola, Mobile Data Division - Seattle, WA Bcc: mitchell in comp.os.minix, dsmythe@netcom.COM (Dave Smythe) said: >In article <1992Feb5.145630.759@wpi.WPI.EDU> entropy@wintermute.WPI.EDU (Lawrence C. Foard) writes: > >David Cheriton (Prof. at Stanford, and author of the V system) said something >similar to this in a class in distributed systems. Paraphrased: > > "There are two kinds of researchers: those that have implemented > something and those that have not. The latter will tell you that > there are 142 ways of doing things and that there isn't consensus > on which is best. The former will simply tell you that 141 of > them don't work." > Yeah, but what's the odds on two who have implemented something differently agreeing on which 141 don't work? -- mitchell@mdd.comm.mot.com (Bill Mitchell) 0, unseen,, *** EOOH *** From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds) Newsgroups: comp.os.minix Subject: Apologies (was Re: LINUX is

obsolete) Date: 30 Jan 92 15:38:16 GMT Organization: University of Helsinki In article <1992Jan29.231426.20469@klaava.Helsinki.FI> I wrote: >Well, with a subject like this, I'm afraid I'll have to reply. And reply I did, with complete abandon, and no thought for good taste and netiquette. Apologies to ast, and thanks to John Nall for a friendy "that's not how it's done"-letter. I over-reacted, and am now composing a (much less acerbic) personal letter to ast. Hope nobody was turned away from linux due to it being (a) possibly obsolete (I still think that's not the case, although some of the criticisms are valid) and (b) written by a hothead :-) Linus "my first, and hopefully last flamefest" Torvalds 0, unseen,, *** EOOH *** Newsgroups: comp.os.minix From: jkp@cs.HUT.FI (Jyrki Kuoppala) Subject: ast's comments on OS's [was Re: LINUX is obsolete] In-Reply-To: ast@cs.vu.nl (Andy Tanenbaum) Nntp-Posting-Host: sauna.cs.hut.fi Reply-To: jkp@cs.HUT.FI (Jyrki Kuoppala) Organization: Helsinki University of Technology, Finland Date: Fri, 31 Jan 1992 12:07:46 GMT In article <12595@star.cs.vu.nl>, ast@cs (Andy Tanenbaum) writes: >who want to turn MINIX in BSD UNIX off my back. But in all honesty, I would >suggest that people who want a **MODERN** "free" OS look around for a >microkernel-based, portable OS, like maybe GNU or something like that. I hear bsd 4.4 might also become free and appear in the near future for the 386, also someone's supposed to be working on bsd 4.4 on top of the Mach microkernel, and then there's of course GNU. Currently of course for many people Linux is the OS to use because it's here now, is free and works. >P.S. Just as a random aside, Amoeba has a UNIX emulator (running in user >space), but it is far from complete. If there are any people who would >like to work on that, please let me know. To run Amoeba you need a few 386s, >one of which needs 16M, and all of which need the WD Ethernet card. A note here, the sources I've seen seem to imply that Amoeba will not be free as in you won't be able to use it, copy it, enhance it, share it etc. without paying $$ and/or asking permission from someone. //Jyrki 0, unseen,, *** EOOH *** Newsgroups: comp.os.minix From: geoff@world.std.com (Geoff Collyer) Subject: posixiation (was Re: LINUX is obsolete) Organization: The World @ Software Tool & Die Date: Fri, 31 Jan 1992 01:13:00 GMT Andy Tanenbaum: >MINIX was designed before POSIX, and is now being

(slowly) POSIXized as >everyone who follows this newsgroup knows. May I recommend the use of the verb "posixiate" (by analogy with asphyxiate) instead of "posixize"? Similarly, I prefer "ansitise" (converse and anagram of "sanitise") to "ansify". -- Geoff Collyer world.std.com!geoff, uunet.uu.net!geoff 0, unseen,, *** EOOH *** From: pmacdona@sanjuan (Peter MacDonald) Newsgroups: comp.os.minix Subject: re: Linux is obsolete Date: 1 Feb 92 02:10:06 GMT Organization: University of Victoria, Victoria, BC, CANADA Nntp-Posting-Host: sanjuan.uvic.ca Since I think I posted one of the earliest messages in all this discussion of Minix vs Linux, I feel compelled to comment on my reasons for switching from Minix to Linux. In order of importance they are: 1) Linux is free 2) Linux is evolving at a satisfactory clip (because new features are accepted into the distribution by Linus). The first requires some explanation, because if I have already purchased Minix, what posssible concern could price have for me? Simple. If the OS is free, many more people will use/support/enhance it. This is also the same reasoning I used when I bought my 386 instead of a sparc (which I could have got for just 30% more). Since PCs are cheap and generally available, more people will buy/use them and thus good, cheap/free software will be abundant. The second should be pretty obvious to anyone who has been using Minix for for any period of time. AST generally does not accept enhancements to Minix. This is not meant as a challenge, but merely a statement of fact. AST has good and legitimate reasons for this, and I do not dispute them. But Minix has some limitations which I just could no longer live with, and due to this policy, the prospect of seeing them resolved in reasonable time was unsatisfactory. These limitations include: no 386 support no virtual consoles no soft links no select call no ptys no demand paging/swapping/shared-text/shared-libs... (efficient mm) chmem (inflexible mm) no X-Windows (advocated for the same reasons as Linux and the 386). no TCP/IP no GNU/SysV integration (portability) Some of these could be fixed by patches (and if you have done this yourself, I don't have to tell you how satisfactory that is), but at least the last 5 items were/are beyond any reasonable expectation. Finally, my comment (crack?) about Minix's segmented kernel, or micro-kernel architecture was more an expression of my

frustration/ bewilderment at attempting to use the Minix PTY patches as a guide of how to do it under Linux. That particular instance was one where message passing greatly complicated the implementation of a feature. I do have an opinion about Monlithic vs Message Passing, but won't express it now, and did not mean to expresss it then. My goals are totally short term (maximum functionality in the minimum amount of time/cost/hassle), and so my views on this are irrelevant, and should not be misconstrued. If you are non-plussed by the lack of the above features, then you should consider Minix, as long as you don't mind paying of course :) 0, unseen,, *** EOOH *** From: ts@cup.portal.com (Tim W Smith) Newsgroups: comp.os.minix Subject: re: Linux is obsolete Date: 7 Feb 92 01:52:51 GMT Organization: The Portal System (TM) Someone says: > If the OS is free, many more people will use/support/enhance it. > This is also the same reasoning I used when I bought my 386 instead > of a sparc (which I could have got for just 30% more). Since > PCs are cheap and generally available, more people will buy/use > them and thus good, cheap/free software will be abundant. Good cheap/free software will also become redundant. Who cares if the cheap machine has fifteen versions of everything available and the more expensive machine only has one or two, if the one or two are good? The only real exception to this seems to be games. A lot of games seem to only become available on one system.

Tim Smith 0, unseen,, *** EOOH *** From: jmaynard@oac.hsc.uth.tmc.edu (Jay Maynard) Newsgroups: comp.os.minix Subject: Re: Linux is obsolete Date: 7 Feb 92 11:59:55 GMT Organization: UT Health Science Center Houston Nntp-Posting-Host: oac.hsc.uth.tmc.edu In article <54087@cup.portal.com> ts@cup.portal.com (Tim W Smith) writes: >Good cheap/free software will also become redundant. Who cares if the >cheap machine has fifteen versions of everything available and the >more expensive machine only has one or two, if the one or two are >good? I do, if the one or two versions available are implemented in a fashion I have some kind of problem with. -- Jay Maynard, EMT-P, K5ZC, PP-ASEL | Never ascribe to malice that which can jmaynard@oac.hsc.uth.tmc.edu | adequately be explained by a .sig virus. "IMHO, USENET includes printing out articles and sticking them on my fridge with little

magnets." -- Charles Geyer 0, unseen,, *** EOOH *** From: olaf@oski.toppoint.de (Olaf Schlueter) Newsgroups: comp.os.minix Subject: Re: Linux is obsolete Date: 7 Feb 92 11:41:44 GMT Organization: Toppoint Mailbox e.V. Just a few comments to the discussion of Linux vs Minix, which evolved partly to a discussion of monolithic vs micro-kernel. I think there will be no aggreement between the two parties advocating either concept, if they forget, that Linux and Minix have been designed for different applications. If you want a cheap, powerful and enhancable Unix system running on a single machine, with the possibility to adapt standard Unix software without pain, then Linux is for you. If you are interested in modern operating system concepts, and want to learn how a microkernel based system works, then Minix is the better choice. It is not an argument against microkernel system, that for the time being monolithic implemenations of Unix on PCs have a better performance. This means only, that Unix is maybe better implemented as a monolithic OS, at least as long as it runs on a single machine. From the users point of view, the internal design of the OS doesn't matter at all. Until it comes to networks. On the monolithic approach, a file server will become a user process based on some hardware facility like ethernet. Programs which want to use this facility will have to use special libraries which offer the calls for communication with this server. In a microkernel system it is possible to incorporate the server into the OS without the need for new "system" calls. From the users point of view this has the advantage, that nothing changes, he just gets better performance (in terms of more disk space for example). From the implementors point of view, the microkernel system is faster adaptable to changes in hardware design. It has been critized, that AST rejects any improvements to Minix. As he is interested in the educational value of Minix, I understand his argument, that he wants to keep the code simple, and don't want to overload it with features. As an educational tool, Minix is written as a microkernel system, although it is running on hardware platforms, who will probably better perform with a monolithic OS. But the area of network applications is growing and modern OS like Amoeba or Plan 9 cannot be written as monolithic systems. So Minix has been written with the intention to give students a practical example of a microkernel OS, to let them play with tasks and messages. It was

not the idea to give a lot of people a cheap, powerful OS for a tenth of the price of SYSV or BSD implementations. Resumee: Linux is not better than Minix, or the other way round. They are different for good reasons. -- Olaf Schlueter, Sandkuhle 4-6, | olaf@oski.toppoint.de, 2300 Kiel 1, Germany, Toppoint Mailbox e.V. | olaf@tpki.toppoint.de "When MSDOS was written specifically for the 8088 ..., this was less then brilliant. Writing an OS only for the 386 in 91 gets you the second 'F'..." AST 0, unseen,, *** EOOH *** From: cwr@pnet01.cts.com (Will Rose) Newsgroups: comp.os.minix Subject: Re: LINUX is obsolete Date: 1 Feb 92 12:16:12 GMT Organization: People-Net [pnet01], El Cajon CA I've used Minix quite a bit on a PC XT, from version 1.2 onwards, and a couple of points seem worth making. Firstly that I ordered version 1.1 from Prentice Hall, and am devoutly thankful that they delayed my order until 1.2 was available. The first version of something as complicated as an OS is only for the dedicated, and that goes for Linux too I should think. Secondly Minix has evolved to a reliable OS on its original PC platform, but is still getting there on eg. the Mac; these things do take time. Thirdly even (standard) PC 1.5 Minix won't run a lot of current Unix software. Partly this is a matter of the hardware being too limited, and partly a matter of Minix being too limited in eg: the tty driver. (And even this tty driver took a lot of sorting out in the early days). Fourthly, I bought my XT four years ago - the motherboard was $110, and memory (falling in price) was $7.00 per 256KB chip. Last autumn I bought my wife an XT to replace her CP/M word-processor - the m/b was $50, and memory was $1.50 a chip. This week I replaced a dead 286 board for a friend - the drop-in 16MHz 386SX was $140, and memory was $40 for 9 x 1MB... If I actually wanted an OS to use today, I think I'd go with Linux; but if I wanted to learn about OS's, I think I'd use Minix. It looks as if they both do what they were designed to do. Will cwr@pnet01.cts.com 0, unseen,, *** EOOH *** From: meggin@epas.utoronto.ca (David Megginson) Newsgroups: comp.os.minix Subject: Mach/Minix/Linux/Gnu etc. Date: 1 Feb 92 17:11:03 GMT Organization: University of Toronto - EPAS Nntp-Posting-Host: epas.utoronto.ca Well, this has been a fun discussion. I am absolutely convinced by Prof. Tannenbaum that a micro-kernel _is_ the way to go, but the more I look at the Minix source, the less I believe

that it is a micro-kernel. I would probably not bother porting Linux to the M68000, but I want more services than Minix can offer. What about a micro-kernel which is message/syscall compatible with MACH? It doesn't actually have to do everything that MACH does, like virtual memory paging -- it just has to _look_ like MACH from the outside, to fool programs like the future Gnu Unix-emulator, BSD, etc. This would extend the useful lives of our M68000- or 80286-based machines for a little longer. In the meantime, I will probably stay with Minix for my ST rather than switching back to MiNT -- after all, Minix at least looks like Unix, while MiNT looks like TOS trying to look like Unix (it has to, to be TOS compatible). David ## ######## David Megginson meggin@epas.utoronto.ca Centre for Medieval Studies david@doe.utoronto.ca University of Toronto 39 Queen's Park Cr. E. ## ######## 0, unseen,, *** EOOH *** From: gkm@sys6626.bison.mb.ca (greg moeller) Newsgroups: comp.os.minix Subject: Minix VS Linux Date: 2 Feb 92 04:38:14 GMT Organization: system 6626 BBS, Winnipeg MB Just wanted to throw in one point about getting Minix because it can run on just an 8088. If I were a student with an 8088 and had a choice of buying Minix for $169, or spending about $140 for a 16 Mhz 386sx motherboard and get Linux for free, I think I'd get the new motherboard. Greg P.S. all in all, I think it depends on what you want to do with the OpSys that should determine what you get. --- (greg moeller) a user of sys6626, running waffle 1.64 E-mail: gkm@sys6626.bison.mb.ca system 6626: 63 point west drive, winnipeg manitoba canada R3T 5G8 0, unseen,, *** EOOH *** From: ortmann@plains.NoDak.edu (Daniel Ortmann) Newsgroups: comp.os.minix Subject: Re: Minix VS Linux Date: 2 Feb 92 19:42:13 GMT Organization: North Dakota State University, Fargo, ND What type of kernal is gnu going to have? 0, unseen,, *** EOOH *** Newsgroups: comp.os.minix From: meggin@epas.utoronto.ca (David Megginson) Subject: Re: Minix VS Linux Organization: University of Toronto - EPAS Date: Sun, 2 Feb 1992 20:23:11 GMT Nntp-Posting-Host: epas.utoronto.ca In article <14883@plains.NoDak.edu> ortmann@plains.NoDak.edu (Daniel Ortmann) writes: >What type of

kernal is gnu going to have? Gnu will have a Unix emulator running on top of the Mach kernel. The Unix emulator is currently in the early stages. There is an information sheet on the Gnu project at prep.ai.mit.edu. David 0, unseen,, *** EOOH *** From: odiug@cip-s08.informatik.rwth-aachen.de (Guido Muesch) Newsgroups: comp.os.minix Subject: Re: Minix VS Linux Date: 4 Feb 92 13:16:26 GMT Organization: RBI - RWTH Aachen Nntp-Posting-Host: cip-s08 gkm@sys6626.bison.mb.ca (greg moeller) writes: >Just wanted to throw in one point about getting Minix because it can run >on just an 8088. The 8088 is obsolate. And my opinion is that Minix should only support the i386 of all intel architectures in the future. It is to much overhead to support the old 16-bit processors. Minix could be cleaned up easier if the old 16-bit stuff will be removed. >If I were a student with an 8088 and had a choice of buying Minix for >$169, or spending about $140 for a 16 Mhz 386sx motherboard and get Linux >for free, I think I'd get the new motherboard. Thats also my reason for suggesting NOT to support the old 16-bit architectures. People who are willing to pay $169 for Minix-PC can also spend another $140 for a new Motherboard. (If not they still have Linux 8-)) Cheers Guido >Greg >P.S. all in all, I think it depends on what you want to do with the OpSys > that should determine what you get. >--- (greg moeller) a user of sys6626, running waffle 1.64 >E-mail: gkm@sys6626.bison.mb.ca >system 6626: 63 point west drive, winnipeg manitoba canada R3T 5G8 -- /\ bitnet: odiug%messua.informatik.rwth-aachen.de@unido.bitnet / \ email: odiug@messua.informatik.rwth-aachen.de __/ \

_____ \/ snail: Guido Muesch, Ruetscher Str.165/1413, 5100 Aachen 0, unseen,, *** EOOH *** From: vpaul@nmsu.edu (Vincent J. Paul) Newsgroups: comp.os.minix Subject: Re: Minix VS Linux Summary: rebuttal flame-from-hell Date: 4 Feb 92 18:55:47 GMT Organization: New Mexico State University In article <odiug.697209386@cip-s08>, odiug@cip-s08.informatik.rwth-aachen.de (Guido Muesch) writes: > > > The 8088 is obsolate. And my opinion is that Minix should only support > the i386 of all intel architectures in the future. It is to much overhead > to support the old 16-bit processors. Minix could be cleaned up easier if > the old 16-bit

stuff will be removed. > > >If I were a student with an 8088 and had a choice of buying Minix for > >$169, or spending about $140 for a 16 Mhz 386sx motherboard and get Linux > >for free, I think I'd get the new motherboard. > > Thats my reason for suggesting NOT to support the old 16-bit architectures. > People who are willing to pay $169 for Minix-PC can also spend another $140 > for a new Motherboard. (If not they still have Linux 8-)) > You missed the point completely, Guido. I'll be generous, though, and assume it was due to language diffulties. In smaller words, if you already own an 8088 box (and a lot of people still do) then buying a new box for $800 is MUCH more expensive than buying a $169 OS that is compatible. Saying "[spend] $140 on a 386 motherboard and get Linux for free..." is like saying to Ethiopians "Just buy a plane ticket to somewhere where there is a food surplus." AST made a few good points which have been glossed over: with limited budgets (like mine), and a fixed set of hardware, Minix is much more affordable than Linux. IF (and only if) that fixed set of hardware jsut HAPPENS to be 386- based, THEN Linux is a great bargain. Your attitude is much like that attributed to Marie Antoinette\--"Let them eat cake!" Atari and Mac owners have the wrong hardware. Let them get the proper one! (Note: not the proper "ones"--portability, bud, is a GOOD thing.) And as for that statement, "suggesting NOT to support the old 16-bit architectures," are you offering to pay for the upgrades? Somehow I doubt it. Just for your information (and the world at large) I bought Minix 1-1/2 years ago, and my XT-clone 5 years ago for $400. Since then, I've been going to school, not getting rich. I can hardly afford eating regularly, much less keeping up with Intel's CPU-chip fashions. If you want a 32-bit Intel-dedicated OS, go get Linux, and get the hell off this newsgroup. Don't go toasting anyone for providing, as Andy Tanenbaum REALLY has, a public service. Even if he doesn't cater to you. There, I've had my day. Let's call an end to this controversy, shall we? If you prefer Linux, go hang out with the Linux crowd over on (I think) alt.linux. Rag on Minix all you can stand there. Hanging around in comp.os.minix reeks of low to lowest class. And if anyone was going to convert, they already would have by now, don't you think? (Really. DON'T you?) Vince Paul/vpaul@nmsu.edu ---------- ---------- Vincent J. Paul | vpaul@nmsu.edu New

Mexico State University | OPR016%BITNET.NMSUVM1 "I take no resposibility for anyone believing anything I say. Even this." 0, unseen,, *** EOOH *** From: asg@sage.cc.purdue.edu (The Grand Master) Newsgroups: comp.os.minix Subject: Re: Minix VS Linux Date: 6 Feb 92 02:10:58 GMT Organization: Purdue University Computing Center In article <1992Feb4.185547.9585@nmsu.edu> vpaul@nmsu.edu (Vincent J. Paul) writes: }In article <odiug.697209386@cip-s08>, odiug@cip-s08.informatik.rwth-aachen.de (Guido Muesch) writes: }> }> }> }> >If I were a student with an 8088 and had a choice of buying Minix for }> >$169, or spending about $140 for a 16 Mhz 386sx motherboard and get Linux }> >for free, I think I'd get the new motherboard. }> }> Thats my reason for suggesting NOT to support the old 16-bit architectures. }> People who are willing to pay $169 for Minix-PC can also spend another $140 }> for a new Motherboard. (If not they still have Linux 8-)) }> }You missed the point completely, Guido. I'll be generous, though, and }assume it was due to language diffulties. In smaller words, if you This is totally uncalled for - and I think you owe Guido an apology. Geez - and we (Americans) wonder why we have such a bad reputation abroad. }already own an 8088 box (and a lot of people still do) then buying a }new box for $800 is MUCH more expensive than buying a $169 OS that is }compatible. Maybe you are the one who is not understanding. What he is saying is that you do not NEED to spend $800 on a new system, just $140 for a 386sx motherboard. }Saying "[spend] $140 on a 386 motherboard and get Linux }for free..." is like saying to Ethiopians "Just buy a plane ticket to }somewhere where there is a food surplus." Not even close. IF SOMEONE CAN AFFORD TO BUY MINIX, THEN THEY CAN OBVIOUSLY AFFORD TO PUT THAT $169 DOLLARS TOWARDS A 386sx MOTHERBOARD! And *SAVE* $29 in the process if you find a 386sx motherboard for $140. So your analogy does not hold - considering that a starving Ethiopian is not likely to have the money to buy a plane ticket to anywhere. }AST made a few good points }which have been glossed over: with limited budgets (like mine), and }a fixed set of hardware, Minix is much more affordable than Linux. }IF (and only if) that fixed set of hardware jsut HAPPENS to be 386- }based, THEN Linux is a great bargain. Your attitude is much like that

}attributed to Marie Antoinette\--"Let them eat cake!" Atari and Mac }owners have the wrong hardware. Let them get the proper one! (Note: }not the proper "ones"--portability, bud, is a GOOD thing.) Portability is both good and bad - the unfortunate thing about portability is that it does not allow you to take advantage of many hardware-specific features. } Just for your information (and the world at large) I bought }Minix 1-1/2 years ago, and my XT-clone 5 years ago for $400. Since }then, I've been going to school, not getting rich. I can hardly }afford eating regularly, much less keeping up with Intel's CPU-chip }fashions. If you want a 32-bit Intel-dedicated OS, go get Linux, }and get the hell off this newsgroup. Don't go toasting anyone for }providing, as Andy Tanenbaum REALLY has, a public service. Even }if he doesn't cater to you. Mr. Tanenbaum, though undoubtedly a gifted professor, and undoubtedly a gifted programmer, is not providing a "PUBLIC SERVICE". MINIX was written for his students, and AST had the fortune that MINIX caught on, and has now sold many copies. AST makes money off the deal I am sure - and if he does not, then I cannot understand why he does not free the code of copyright, so it can be distributed. And you might remember that the person who started this thread was not Guido, nor anyone else in the "Linux camp". } And if anyone was going to convert, they already }would have by now, don't you think? (Really. DON'T you?) } It really has nothing to do with conversion. Don't get me wrong - I think MINIX is a brilliant piece of work, and it is certainly well thought of by some (likely large) segment of the OS-knowledgable population. But AST attacked Linux for no reason whatsoever. Linus has created a brilliant piece of work as well - one that incorperates many of the features that people want which are not readily (or at all in some cases) available for MINIX. The important thing about Linux is that it is here, and it works, and its free. There is NO other OS available (portable or not) that fits the bill that Linux does. AST's comments were absolutely uncalled for - and his feelings for those of us out here who would like some common useful features (like > 64k segments, or job control, or TCP/IP) are readily appearant from his comment about (note: slightly paraphrased) ".....keeping all those people who want to turn MINIX into BSD...." off his back. Clearly he is not in the "Public Service" field. -- -How long must we fight? How long Courtesy of Bruce Varney until we can live in

peace. asg@sage.cc.purdue.edu -Until the madmen are dead my son, Or until they realize that they cannot count on us to do nothing 0, unseen,, *** EOOH *** From: stolk@fwi.uva.nl (Bram) Newsgroups: comp.os.minix Subject: 'folks with enough money' (was: LINUX-MINIX) Date: 3 Feb 92 11:43:17 GMT Organization: FWI, University of Amsterdam Nntp-Posting-Host: irwin.fwi.uva.nl Hi, In one of the previous articles, ast wrote: > Just for the record, as of about 1 year ago, there were two >versions, one for the PC (360K diskettes) and one for the 286/386 (1.2M). >The PC version was outselling the 286/386 version by 2 to 1. I don't have >figures, but my guess is that the fraction of the 60 million existing PCs that >are 386/486 machines as opposed to 8088/286/680x0 etc is small. Among students >it is even smaller. Making software free, but only for folks with enough money >to buy first class hardware is an interesting concept. Hmm... can I do a little math here? define: configuration 1: 8088 PC, 2 floppydrives, monitor. configuration 2: 386 AT, 2 floppydrives, monitor. let A be the price of config. 1 at the time of MINIX' conception. Let B be the price of config. 2 at the time of LINUX' conception. Question: What are the chances of 'A >= B' being true? My answer: I would say, better than average. The point I am trying to make is: nowadays, students buy 386s. Not because students are rich, but because 386s from Taiwan come VERY cheap. (Many of my fellow students have 386s) So 'the folks with enough money' bit is kinda naive. IBM's 8088s used to sell at the price of small cars, in the old days. I would favor LINUX, even if it were only for the lack of proper object files in MINIX. Bram Stolk 0, unseen,, *** EOOH *** From: cwr@pnet01.cts.com (Will Rose) Newsgroups: comp.os.minix Subject: Re: Minix VS Linux Date: 3 Feb 92 14:16:11 GMT Organization: People-Net [pnet01], El Cajon CA kevin@taronga.taronga.com (Kevin Brown) writes: >It has been brought to my attention that my last posting was exceedingly >harsh. Having reread it, I'm inclined to agree. Didn't seem so to me - or is everyone else getting much more tactful in the 'kinder, gentler' 1990s? >Despite that, Minix is quite usable in many ways as a personal operating >system, i.e. one where there is usually only one person logged into the >system. .... lines omitted >However, as a *multiuser* operating system, i.e. an operating system designed >to efficiently meet

the needs of multiple users simultaneously while also >performing batch operations, Minix is lacking, as far as I'm concerned. >The main reason, of course, is the single-threaded file system (hereafter, >STFS). In fact, Minix has noticeable problems even as a *single-user* system; such a user is likely to be developing code in an 'edit - compile - test' cycle, and on a multitasking machine compilation can conveniently take place in the background using multiple processes. Given the small amount of memory on the early Minix machines, it probably wasn't practical to edit in the foreground during a compile; but a low-end XT such as mine can now put the ram-disk in expanded memory and have 500KB free after the OS is loaded. However, editing during a compile is still totally impractical; the standard 1.5 scheduler can't cope, and keyboard response time is both long (seconds) and variable. The fix is simple, use Kai-Uwe Bloem's (spelling ?) patch. KuB's scheduler algorithm isn't the last word in sophistication, and the code change is a small one, yet it is extremely effective. I don't understand why the original scheduler lasted so long. Broadly speaking, when porting programs to Minix, or writing them from scratch, the main problems are not details of kernel implementation but things that are simply missing, such as non-blocking reads or SIGHUP. If a program is to be used on the majority of Minix systems, then it must be written for the lowest common denominator ie: the standard current release, and this has significant limitations. Only after these are overcome does the lack of memory space, or lack of filesystem performance, become important. (It is quite possible that many of these problems will be fixed in 1.6, but that still lies in the future. By the time it arrives, the goalposts will have moved again...) >Someone, either here on this newsgroup or over on alt.os.linux, made a >very valid observation: the cost of a 16 MHz 386SX system is about $140 >more than a comparably equipped (in terms of RAM size, display technology, >hard drive space, etc.) 8088 system. Minix is $169. In economic terms, >Linux wins if you have to buy Minix. This may have been drawn from a comment of mine, concerning the recent replacement of a dead 286 motherboard with a 386SX; it was part of an example of just how fast hardware prices are falling. The *board* price was $140, ie: $90 more than an XT board. Memory is about the same price for both boards, but obviously the system price for the 386 will tend to be higher with the use of faster,

larger disk drives, more memory, and so on. In the last year PH has started marketing Minix as a small general-purpose/ training Unix, but if Minix 2.0 isn't released for another year, I think it will become strictly an adjunct to a successful textbook. It won't run enough of the then-available PD code to be useful for anything else, since such code will mostly require gcc and megabytes of memory, implying at least a 386. (I'm assuming that the cheapest hardware will still be Intel/ISA bus). Linux seems well-placed to pick up on this wave; whether it will also make inroads into academe via easy FTP availability (a version of 'OS Design and Implementation' without Minix is coincidentally in hand) remains to be seen. Will Rose cwr@pnet01.cts.com 0, unseen,, *** EOOH *** From: kevin@taronga.taronga.com Newsgroups: comp.os.minix Subject: Re: Minix VS Linux Date: 4 Feb 92 07:56:04 GMT Organization: University of Houston In article <1992Feb03.141611.6995@crash.cts.com> cwr@pnet01.cts.com (Will Rose) writes: > >kevin@taronga.taronga.com (Kevin Brown) writes: > >>It has been brought to my attention that my last posting was exceedingly >>harsh. Having reread it, I'm inclined to agree. > >Didn't seem so to me - or is everyone else getting much more tactful in >the 'kinder, gentler' 1990s? Guess it depends on whether or not you agree with what I had to say. ;-) Some people read other people's postings with an implied "IMHO". Others don't. Those that don't will probably think of that posting as exceedingly harsh. Since I try to be as explicit as possible, I would say I didn't put enough "IMHO"s in. >>Despite that, Minix is quite usable in many ways as a personal operating >>system, i.e. one where there is usually only one person logged into the >>system. >... lines omitted >>However, as a *multiuser* operating system, i.e. an operating system designed >>to efficiently meet the needs of multiple users simultaneously while also >>performing batch operations, Minix is lacking, as far as I'm concerned. >>The main reason, of course, is the single-threaded file system (hereafter, >>STFS). > >In fact, Minix has noticeable problems even as a *single-user* system; such >a user is likely to be developing code in an 'edit - compile - test' cycle, >and on a multitasking machine compilation can conveniently take place in >the background using multiple processes. Given the small amount of memory >on the early Minix machines, it probably wasn't practical to edit in the >foreground during a compile;

but a low-end XT such as mine can now put the >ram-disk in expanded memory and have 500KB free after the OS is loaded. >However, editing during a compile is still totally impractical; the standard >1.5 scheduler can't cope, and keyboard response time is both long (seconds) >and variable. The fix is simple, use Kai-Uwe Bloem's (spelling ?) patch. >KuB's scheduler algorithm isn't the last word in sophistication, and the code >change is a small one, yet it is extremely effective. I don't understand >why the original scheduler lasted so long. I agree wholeheartedly. In fact, Bruce Evans sent me a very simple patch that gives an I/O bound process like an editor effectively infinite priority: it causes the process that's being readied (in the ready() routine in proc.c) to be added to the beginning of the process queue instead of the end. This fixes the problem for sure, but it causes CPU bound processes to suffer rather badly in the face of I/O bound processes. >Broadly speaking, when porting programs to Minix, or writing them from >scratch, the main problems are not details of kernel implementation but things >that are simply missing, such as non-blocking reads or SIGHUP. I've had the SIGHUP problem solved on my system for some time, but I've got weird problems with the signal handling as relates to process groups and signalling a parent process in a process group before signalling the child. Causes very strange things to happen (like the FS to restart! You get the message "executing in protected mode" as a result of killing a parent process before killing the child process). This is the primary reason I don't have UUCP going on my system already. I don't trust it enough to handle a UUCP connection going away (dropping carrier) in the middle of a transfer. >If a program >is to be used on the majority of Minix systems, then it must be written for >the lowest common denominator ie: the standard current release, and this has >significant limitations. Only after these are overcome does the lack of >memory space, or lack of filesystem performance, become important. (It >is quite possible that many of these problems will be fixed in 1.6, but >that still lies in the future. By the time it arrives, the goalposts will >have moved again...) What he said. :-) >>Someone, either here on this newsgroup or over on alt.os.linux, made a >>very valid observation: the cost of a 16 MHz 386SX system is about $140 >>more than a comparably equipped (in terms of RAM size, display technology, >>hard drive space, etc.) 8088 system. Minix

is $169. In economic terms, >>Linux wins if you have to buy Minix. > >This may have been drawn from a comment of mine, concerning the recent >replacement of a dead 286 motherboard with a 386SX; it was part of an >example of just how fast hardware prices are falling. The *board* price >was $140, ie: $90 more than an XT board. Memory is about the same price >for both boards, but obviously the system price for the 386 will tend to be >higher with the use of faster, larger disk drives, more memory, and so on. Right. The only time the architecture bites you is when your applications are larger as a result of using a 32-bit memory model, though. So while you tend to use faster and larger disk drives, more memory, etc. for the 386, that's because you *can* more than anything else. The architecture supports it. I figure the larger and faster hard drives come as a result of having more memory to stuff programs into. :-) If the price difference is only $90, then the 386SX *definitely* wins. For the $80 I save in buying a 386SX and getting Linux, I can get another 50 meg of hard drive space, or perhaps even more, depending on where on the price/capacity curve I am. >In the last year PH has started marketing Minix as a small general-purpose/ >training Unix, but if Minix 2.0 isn't released for another year, I think it >will become strictly an adjunct to a successful textbook. It won't run >enough of the then-available PD code to be useful for anything else, since >such code will mostly require gcc and megabytes of memory, implying at least >a 386. (I'm assuming that the cheapest hardware will still be Intel/ISA bus). I don't know. While I agree that many of the applications will be too large, there's something to be said for small, efficient programs, and something to be said against large, baroque programs. Having lots of features isn't necessarily a good thing, despite what the marketers might tell you. >Linux seems well-placed to pick up on this wave; whether it will also make >inroads into academe via easy FTP availability (a version of 'OS Design and >Implementation' without Minix is coincidentally in hand) remains to be seen. Maybe we can get Linus to write a book on operating systems. :-) :-) >Will Rose >cwr@pnet01.cts.com Kevin Brown 0, unseen,, *** EOOH *** From: peter@ferranti.com (peter da silva) Newsgroups: comp.os.minix Subject: What good does this war do? (Re: LINUX is obsolete) Date: 3 Feb 92 16:37:24 GMT Organization: Xenix Support, FICC Will you quit flaming each other?

I mean, linux is designed to provide a reasonably high performance environment on a hardware platform crippled by years of backwards-compatible kludges. Minix is designed as a teaching tool. Neither is that good at doing the other's job, and why should they? The fact that Minix runs out of steam quickly (and it does) isn't a problem in its chosen mileau. It's sure better than the TOY operating system. The fact that Linux isn't transportable beyond the 386/AT platform isn't a problem when there are millions of them out there (and quite cheap: you can get a 386/SX for well under $1000). A monolithic kernel is easy enough to build that it's worth doing it if it gets a system out the door early. Think of it as a performance hack for programmer time. The API is portable. You can replace the kernel with a microkernel design (and MINIX isn't the be-all and end-all of microkernel designs either: even for low end PCs... look at AmigaOS) without disturbing the applications. That's the whole point of a portable API in the first place. Microkernels are definitely a better design for many tasks. I takes more work to make them efficient, so a simpler design that doesn't take advantage of the microkernel in any real way is worth doing for pedagogical reasons. Think of it as a performance hack for student time. The design is still good and when you can get an API to the microkernel interface you can get VERY impressive performance (thousands of context switches per second on an 8 MHz 68000). -- -- Peter da Silva, Ferranti International Controls Corporation -- Sugar Land, TX 77487-5012; +1 713 274 5180 -- "Have you hugged your wolf today?" 0, unseen,, *** EOOH *** From: ast@cs.vu.nl (Andy Tanenbaum) Newsgroups: comp.os.minix Subject: Unhappy campers Date: 3 Feb 92 22:46:40 GMT Organization: Fac. Wiskunde & Informatica, Vrije Universiteit, Amsterdam I've been getting a bit of mail lately from unhappy campers. (Actually 10 messages from the 43,000 readers may seem like a lot, but it is not really.) There seem to be three sticking points: 1. Monolithic kernels are just as good as microkernels 2. Portability isn't so important 3. Software ought to be free If people want to have a serious discussion of microkernels vs. monolithic kernels, fine. We can do that in comp.os.research. But please don't sound off if you have no idea of what you are talking about. I have helped design and implement 3 operating systems, one monolithic and two micro, and have studied many others in detail. Many of the arguments offered are

nonstarters (e.g., microkernels are no good because you can't do paging in user space-- except that Mach DOES do paging in user space). If you don't know much about microkernels vs. monolithic kernels, there is some useful information in a paper I coauthored with Fred Douglis, Frans Kaashoek and John Ousterhout in the Dec. 1991 issue of COMPUTING SYSTEMS, the USENIX journal). If you don't have that journal, you can FTP the paper from ftp.cs.vu.nl (192.31.231.42) in directory amoeba/papers as comp_sys.tex.Z (compressed TeX source) or comp_sys.ps.Z (compressed PostScript). The paper gives actual performance measurements and supports Rick Rashid's conclusion that microkernel based systems are just as efficient as monolithic kernels. As to portability, there is hardly any serious discussion possible any more. UNIX has been ported to everything from PCs to Crays. Writing a portable OS is not much harder than a nonportable one, and all systems should be written with portability in mind these days. Surely Linus' OS professor pointed this out. Making OS code portable is not something I invented in 1987. While most people can talk rationally about kernel design and portability, the issue of free-ness is 100% emotional. You wouldn't believe how much [expletive deleted] I have gotten lately about MINIX not being free. MINIX costs $169, but the license allows making two backup copies, so the effective price can be under $60. Furthermore, professors may make UNLIMITED copies for their students. Coherent is $99. FSF charges >$100 for the tape its "free" software comes on if you don't have Internet access, and I have never heard anyone complain. 4.4 BSD is $800. I don't really believe money is the issue. Besides, probably most of the people reading this group already have it. A point which I don't think everyone appreciates is that making something available by FTP is not necessarily the way to provide the widest distribution. The Internet is still a highly elite group. Most computer users are NOT on it. It is my understanding from PH that the country where MINIX is most widely used is Germany, not the U.S., mostly because one of the (commercial) German computer magazines has been actively pushing it. MINIX is also widely used in Eastern Europe, Japan, Israel, South America, etc. Most of these people would never have gotten it if there hadn't been a company selling it. Getting back to what "free" means, what about free source code? Coherent is

binary only, but MINIX has source code, just as LINUX does. You can change it any way you want, and post the changes here. People have been doing that for 5 years without problems. I have been giving free updates for years, too. I think the real issue is something else. I've been repeatedly offered virtual memory, paging, symbolic links, window systems, and all manner of features. I have usually declined because I am still trying to keep the system simple enough for students to understand. You can put all this stuff in your version, but I won't put it in mine. I think it is this point which irks the people who say "MINIX is not free," not the $60. An interesting question is whether Linus is willing to let LINUX become "free" of his control. May people modify it (ruin it?) and sell it? Remember the hundreds of messages with subject "Re: Your software sold for money" when it was discovered the MINIX Centre in England was selling diskettes with news postings, more or less at cost? Suppose Fred van Kempen returns from the dead and wants to take over, creating Fred's LINUX and Linus' LINUX, both useful but different. Is that ok? The test comes when a sizable group of people want to evolve LINUX in a way Linus does not want. Until that actually happens the point is moot, however. If you like Linus' philosophy rather than mine, by all means, follow him, but please don't claim that you're doing this because LINUX is "free." Just say that you want a system with lots of bells and whistles. Fine. Your choice. I have no argument with that. Just tell the truth. As an aside, for those folks who don't read news headers, Linus is in Finland and I am in The Netherlands. Are we reaching a situation where another critical industry, free software, that had been totally dominated by the U.S. is being taken over by the foreign competition? Will we soon see President Bush coming to Europe with Richard Stallman and Rick Rashid in tow, demanding that Europe import more American free software? Andy Tanenbaum (ast@cs.vu.nl) 0, unseen,, *** EOOH *** From: meulenbr@ce.philips.nl (Frans Meulenbroeks) Newsgroups: comp.os.minix Subject: Re: Unhappy campers Date: 4 Feb 92 17:21:31 GMT Organization: Philips Consumer Electronics, Eindhoven First of all I want to say, that I like to support the idea of Andy to keep minix small. If you want something large go for GNU. I'm also against featurism. However, undoubtly my wanted set of features differs from Andy's set. Therefore what I would like to see, is something which

allows it to add features easily. If for instance device drivers are loadable, or easier to add as they are now, then the core and some drivers can be the standard PH minix. If people want to add something, then it is nice to have a mechanism with which you can do so, without having to meddle with things like fs/table.c kernel/table.c include/minix/config.h (for NR_TASKS) etc. ast@cs.vu.nl (Andy Tanenbaum) writes: >Suppose Fred van Kempen returns from the dead and wants to take over, creating Just a remark for the readers. As far as I know Fred is not physically dead. If english is not your native language (just like mine) you might conclude otherwise from ast's words. Fred is only electronically dead. -- Frans Meulenbroeks (meulenbr@prl.philips.nl) Philips Research Laboratories 0, unseen,, *** EOOH *** From: jonathan@ukmug.uk.mugnet.org (Jonathan Allen) Newsgroups: comp.os.minix Subject: Re: Unhappy campers Date: 6 Feb 92 11:43:05 GMT Organization: MUGNET UK Backbone (UKMUG) In article <1992Feb4.172131.18145@philce.ce.philips.nl>, meulenbr@ce.philips.nl (Frans Meulenbroeks) wrote: > ast@cs.vu.nl (Andy Tanenbaum) writes: > >>Suppose Fred van Kempen returns from the dead and wants to take over, creating > > Just a remark for the readers. As far as I know Fred is not physically dead. > If english is not your native language (just like mine) you might > conclude otherwise from ast's words. Fred is only electronically dead. Well, just you wait and see :-) !!! 0, unseen,, *** EOOH *** From: meggin@epas.utoronto.ca (David Megginson) Newsgroups: comp.os.minix Subject: Re: Unhappy campers Date: 4 Feb 92 20:17:22 GMT Organization: University of Toronto - EPAS Nntp-Posting-Host: epas.utoronto.ca In article <12667@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes: >As an aside, for those folks who don't read news headers, Linus is in Finland >and I am in The Netherlands. Are we reaching a situation where another >critical industry, free software, that had been totally dominated by the U.S. >is being taken over by the foreign competition? Will we soon see >President Bush coming to Europe with Richard Stallman and Rick Rashid >in tow, demanding that Europe import more American free software? Actually, the U.S. is more likely to have you arrested for selling restricted software to unfriendly powers (not that anyone in the C.I.S. can afford software this winter -- they're probably burning their PCs for heat). I remember that

they jailed some poor computer vendor in the U.K. a couple of years ago. Seriously, you must have run into this problem with crypt and Minix distribution in and out of the U.S. David ## ######## David Megginson meggin@epas.utoronto.ca Centre for Medieval Studies david@doe.utoronto.ca University of Toronto 39 Queen's Park Cr. E. 0, unseen,, *** EOOH *** Newsgroups: comp.os.minix From: jkp@cs.HUT.FI (Jyrki Kuoppala) Subject: Re: Unhappy campers In-Reply-To: meggin@epas.utoronto.ca (David Megginson) Nntp-Posting-Host: sauna.cs.hut.fi Reply-To: jkp@cs.HUT.FI (Jyrki Kuoppala) Organization: Helsinki University of Technology, Finland Date: Wed, 5 Feb 1992 15:47:26 GMT In article <1992Feb4.201722.20620@epas.toronto.edu>, meggin@epas (David Megginson) writes: >Actually, the U.S. is more likely to have you arrested for selling >restricted software to unfriendly powers (not that anyone in the >C.I.S. can afford software this winter -- they're probably burning >their PCs for heat). I remember that they jailed some poor computer >vendor in the U.K. a couple of years ago. Around here a few years ago they had a trial against a couple of guys who sold old VAXes to the Soviet Union. They were charged with treason because they were breaking some export rules. They weren't convicted, though, I think it was something about the export rules not being a law in Finland. For some reason the U.S. troups didn't come to Finland to kill a few thousand people and take these guys to USA for a trial like they did with another person in another country. //Jyrki 0, unseen,, *** EOOH *** From: ast@cs.vu.nl (Andy Tanenbaum) Newsgroups: comp.os.minix Subject: Re: Unhappy campers Date: 6 Feb 92 11:03:51 GMT Organization: Fac. Wiskunde & Informatica, Vrije Universiteit, Amsterdam In article <1992Feb5.154726.983@nntp.hut.fi> jkp@cs.HUT.FI (Jyrki Kuoppala) writes: > >Around here a few years ago they had a trial against a couple of guys >who sold old VAXes to the Soviet Union. In their book, Cyberpunk, Katie Hafner and John Markoff report that a group of hackers tried to sell MINIX to the KGB claiming it was the VMS source code. The KGB didn't buy. Andy Tanenbaum (ast@cs.vu.nl) 0, unseen,, *** EOOH *** From: james@fiskville.mc.utexas.edu (James Hammett) Newsgroups: comp.os.minix Subject: Re: Unhappy campers Date: 8 Feb 92 02:18:28

GMT Reply-To: james@fiskville.mc.utexas.edu (James Hammett) Organization: The University of Texas at Austin From what little I understand, the reason Minix needs to use a second class encryptor for its passwd file is because of the export restrictions. You can't export or transmit the algorhthym (sp). James 0, unseen,, *** EOOH *** From: fnf@fishpond.uucp (Fred Fish) Newsgroups: comp.os.minix Subject: Re: Unhappy campers Date: 4 Feb 92 20:57:40 GMT Organization: Amiga Library Distribution Services In article <12667@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes: >While most people can talk rationally about kernel design and portability, >the issue of free-ness is 100% emotional. You wouldn't believe how much >[expletive deleted] I have gotten lately about MINIX not being free. MINIX >costs $169, but the license allows making two backup copies, so the effective >price can be under $60. Furthermore, professors may make UNLIMITED copies >for their students. Coherent is $99. FSF charges >$100 for the tape its "free" >software comes on if you don't have Internet access, and I have never heard >anyone complain. 4.4 BSD is $800. I don't really believe money is the issue. >Besides, probably most of the people reading this group already have it. The distribution cost is not the problem. As you've noted, nobody complains about the FSF's distribution fee being too high. The problem, as I see it, is that there is only one legal source for for the software for people that simply want a working release. And from watching the minix group since minix first became available, my impression is that nobody enjoys dealing with PH for a whole host of reasons. >I think the real issue is something else. I've been repeatedly offered virtual >memory, paging, symbolic links, window systems, and all manner of features. I >have usually declined because I am still trying to keep the system simple >enough for students to understand. You can put all this stuff in your version, >but I won't put it in mine. I think it is this point which irks the people who >say "MINIX is not free," not the $60. If PH was not granted a monopoly on distribution, it would have been possible for all of the interested minix hackers to organize and set up a group that was dedicated to producing enhanced-minix. This aim of this group could have been to produce a single, supported version of minix with all of the commonly requested enhancements. This would have allowed minix to evolve in much the

same way that gcc has evolved over the last few years. Sure there are variant versions of gcc, but most of the really good enhancements, bug fixes, etc are eventually folded back into a master source base that future distributions derive from. Thus you would have been left in peace to continue your tight control over the educational version of minix, and everyone else that wanted more than an educational tool could put their energies into enhanced-minx. The primary reason I've never gotten into using minix, after the initial excitement of hearing about it's availability way back when, is that I have no interest in trying to apply random patches from all over the place, sort out the problems, and eventually end up with a system that does what I want it to, but which I can't pass on to anyone else. >The >test comes when a sizable group of people want to evolve LINUX in a way Linus >does not want. Until that actually happens the point is moot, however. Where is the sizeable group of people that want to evolve gcc in a way that rms/FSF does not approve of? Where is the sizeable group of people that want to evolve emacs in a way that rms/FSF doesn't approve of? I'd say that if the primary maintainers of a large piece of useful, freely redistributable, software are at all responsive to incorporating useful enhancements and acting as the central repository and clearing house for the software, then these splinter groups simply do not have sufficient motivation to form. Having a single source for the software, and having the primary maintainer(s) be unresponsive to the desires of a large group of users, is the catalyst that causes these sorts of pressures; not the freedom of the software. -Fred -- |\/ o\ Fred Fish, 1835 E. Belmont Drive, Tempe, AZ 85284, USA |/__/ 1-602-491-0048 {asuvax,mcdphx,cygint,amix}!fishpond!fnf 0, unseen,, *** EOOH *** From: vladimir@Eng.Sun.COM (Vladimir Ivanovic) Newsgroups: comp.os.minix Subject: Re: Unhappy campers Date: 5 Feb 92 04:34:18 GMT Organization: Sun Microsystems, Inc. NNTP-Posting-Host: ronnie In-reply-to: fnf@fishpond.uucp's message of 4 Feb 92 20:57:40 GMT >>>>> On 4 Feb 92 20:57:40 GMT, fnf@fishpond.uucp (Fred Fish) said: fnf> If PH was not granted a monopoly on distribution, it would have been fnf> possible for all of the interested minix hackers to organize and set fnf> up a group that was dedicated to producing enhanced-minix. This aim fnf> of this group could have been to produce a single, supported version fnf> of minix with all of the

commonly requested enhancements. This would fnf> have allowed minix to evolve in much the same way that gcc has evolved fnf> over the last few years. Sure there are variant versions of gcc, but fnf> most of the really good enhancements, bug fixes, etc are eventually fnf> folded back into a master source base that future distributions derive fnf> from. Thus you would have been left in peace to continue your tight fnf> control over the educational version of minix, and everyone else that fnf> wanted more than an educational tool could put their energies into fnf> enhanced-minx. I don't get it. What's preventing people from doing this? The quoted paragraph doesn't give any reasons for its assertions. -- Vladimir -- Vladimir G. Ivanovic
Sun Microsystems, Inc (415) 336-2315 MTV12-33
vladimir@Eng.Sun.COM 2550 Garcia Ave.
{decwrl,hplabs,ucbvax}!sun!Eng!vladimir Mountain View, CA 94043-1100 Disclaimer: I speak for myself. 0, unseen,, *** EOOH *** From: mitchell@mdd.comm.mot.com (Bill Mitchell) Newsgroups: comp.os.minix Subject: Re: Unhappy campers Date: 5 Feb 92 15:09:12 GMT Reply-To: mitchell@mdd.comm.mot.com (Bill Mitchell) Organization: Motorola, Mobile Data Division - Seattle, WA Bcc: mitchell in comp.os.minix, vladimir@Eng.Sun.COM (Vladimir Ivanovic) said: >>>>>> On 4 Feb 92 20:57:40 GMT, fnf@fishpond.uucp (Fred Fish) said: > >fnf> If PH was not granted a monopoly on distribution, it would have been >fnf> possible for all of the interested minix hackers to organize and set >fnf> up a group that was dedicated to producing enhanced-minix. This aim >fnf> of this group could have been to produce a single, supported version >fnf> of minix with all of the commonly requested enhancements. This would >fnf> have allowed minix to evolve in much the same way that gcc has evolved >fnf> over the last few years. Sure there are variant versions of gcc, but >fnf> most of the really good enhancements, bug fixes, etc are eventually >fnf> folded back into a master source base that future distributions derive >fnf> from. Thus you would have been left in peace to continue your tight >fnf> control over the educational version of minix, and everyone else that >fnf> wanted more than an educational tool could put their energies into >fnf> enhanced-minx. > >I don't get it. What's preventing people from doing this? The quoted >paragraph doesn't give any reasons for its assertions. > As I understand it PH

takes the position that the copyright must be observed, and that distribution of modified copies of their copyrighted code is in violation of the copyright. I think that they are required to defend the copyright against such violations if they want it to remain legally valid. Diffs against the baseline are allowed. But diffs have to be applied in strict order. This would be difficult to administer in a tight-knit organization with well defined distribution channels. In net.anarchy it's all but unworkable. The fact that the Minix-386 variant of PC Minix 1.5 has managed to remain stable as a set of diffs suprises me. I recall that the NLMUG Minix variant was distributed as modified baseline code instead of as diffs, and that PH copyright lawyers put a stop to that. Or do I have it wrong? -- mitchell@mdd.comm.mot.com (Bill Mitchell) 0, unseen,, *** EOOH *** From: fnf@fishpond.uucp (Fred Fish) Newsgroups: comp.os.minix Subject: Re: Unhappy campers Date: 5 Feb 92 15:33:33 GMT Organization: Amiga Library Distribution Services In article <VLADIMIR.92Feb4203418@ronnie.Eng.Sun.COM> vladimir@Eng.Sun.COM (Vladimir Ivanovic) writes: >>>>>> On 4 Feb 92 20:57:40 GMT, fnf@fishpond.uucp (Fred Fish) said: >fnf> If PH was not granted a monopoly on distribution, it would have been >fnf> possible for all of the interested minix hackers to organize and set > >I don't get it. What's preventing people from doing this? The quoted >paragraph doesn't give any reasons for its assertions. My understanding is that PH is the only one that is allowed to distribute a complete minix distribution. All others are restricted to the "2 copy limit" or by the educational provision for classroom use. There is apparently no place where I can send a check for $50 - $150 and receive a complete, ready to install, copy of an enhanced minix that has most of the common enhancements that are floating around as diff kits, already installed and tested. When I run into a problem, I can't post a message saying that eminix version 15.4 does mumble-foo when it should do mumble-bar, and expect other people to be easily able to reproduce the problem. Except for people that are running strictly vanilla minix as received from PH, I doubt that there are any two minix sites anywhere in the world, that are not directly in touch with each other, that are running the same set of binaries. This is bug heaven, and a maintenance nightmare. If I'm wrong, and there is such a place to get a

full 32 bit minix with VM, hardware supported memory protection, and other assorted enhancements such as GNU gcc, g++, emacs, etc, with full sources and binaries for the entire system, preferably for an Amiga 3000, then please tell me. Directions that go "well first order minix X.X from PH, and then apply patch kit xxx from ftp site yyy and patch kit zzz from ftp site kkk, and write to John Doe for his patch kit www, and pick up patch kit ddd from the minix usenet archives" will be cheerfully ignored. -Fred -- |\/ o\ Fred Fish, 1835 E. Belmont Drive, Tempe, AZ 85284, USA |/__/ 1-602-491-0048 {asuvax,mcdphx,cygint,amix}!fishpond!fnf 0, unseen,, *** EOOH *** From: al@escom.com (Al Donaldson) Newsgroups: comp.os.minix Subject: Re: Unhappy campers Date: 6 Feb 92 14:53:21 GMT Reply-To: al@escom.COM (Al Donaldson) Organization: ESCOM Corp., Oakton VA (USA) In article <207@fishpond.uucp> fnf@fishpond.uucp (Fred Fish) writes: >My understanding is that PH is the only one that is allowed to distribute >a complete minix distribution. Fred, To the best of my knowledge, PH has no such restriction. In fact, as of a year ago, PH had an active program to license MINIX to anyone who wants to use it for whatever purpose. This specifically includes building variants of the MINIX system such as "enhanced" MINIX. >There is apparently no place where I can send a check for $50 - $150 and >receive a complete, ready to install, copy of an enhanced minix that has >most of the common enhancements that are floating around as diff kits, >already installed and tested. Although I didn't explore this specifically, I saw nothing that would preclude someone or some company from signing a license agreement with PH, building an "enhanced" version of MINIX, and then selling as many as the market will bear. The only rule is that this licensee (Company X) would have to behave like a business: report each sale and collect a license fee. The license fees depend upon the quantity Company X believes can be sold, and require some amount of prepayment. License fees appear to be fairly reasonable, for example, perhaps US $50 at quantity 1000. But for this license fee, Company X must duplicate its own diskettes and produce its own release notes. And answer its own phones, and handle customer gripes, and strike its own licensing agreements with the individuals who wrote the various add-on packages that make up eminix. And so on.. >If I'm wrong, and

there is such a place to get a full 32 bit minix with >VM, hardware supported memory protection, and other assorted enhancements >such as GNU gcc, g++, emacs, etc, with full sources and binaries for the >entire system, preferably for an Amiga 3000, then please tell me. I don't know of anyone who sells this, but that's not PH's fault.. Al 0, unseen,, *** EOOH *** From: ast@cs.vu.nl (Andy Tanenbaum) Newsgroups: comp.os.minix Subject: Re: Unhappy campers Date: 5 Feb 92 23:23:26 GMT Organization: Fac. Wiskunde & Informatica, Vrije Universiteit, Amsterdam In article <205@fishpond.uucp> fnf@fishpond.uucp (Fred Fish) writes: >If PH was not granted a monopoly on distribution, it would have been possible >for all of the interested minix hackers to organize and set up a group that >was dedicated to producing enhanced-minix. This aim of this group could have >been to produce a single, supported version of minix with all of the commonly >requested enhancements. This would have allowed minix to evolve in much the >same way that gcc has evolved over the last few years. This IS possible. If a group of people wants to do this, that is fine. I think co-ordinating 1000 prima donnas living all over the world will be as easy as herding cats, but there is no legal problem. When a new release is ready, just make a diff listing against 1.5 and post it or make it FTPable. While this will require some work on the part of the users to install it, it isn't that much work. Besides, I have shell scripts to make the diffs and install them. This is what Fred van Kempen was doing. What he did wrong was insist on the right to publish the new version, rather than diffs against the PH baseline. That cuts PH out of the loop, which, not surprisingly, they weren't wild about. If people still want to do this, go ahead. Of course, I am not necessarily going to put any of these changes in my version, so there is some work keeping the official and enhanced ones in sync, but I am willing to co-operate to minimize work. I did this for a long time with Bruce Evans and Frans Meulenbroeks. If Linus wants to keep control of the official version, and a group of eager beavers want to go off in a different direction, the same problem arises. I don't think the copyright issue is really the problem. The problem is co-ordinating things. Projects like GNU, MINIX, or LINUX only hold together if one person is in charge. During the 1970s, when structured programming was introduced, Harlan Mills pointed out that the programming team should

be organized like a surgical team--one surgeon and his or her assistants, not like a hog butchering team--give everybody an axe and let them chop away. Anyone who says you can have a lot of widely dispersed people hack away on a complicated piece of code and avoid total anarchy has never managed a software project. >Where is the sizeable group of people that want to evolve gcc in a way that >rms/FSF does not approve of? A compiler is not something people have much emotional attachment to. If the language to be compiled is a given (e.g., an ANSI standard), there isn't much room for people to invent new features. An operating system has unlimited opportunity for people to implement their own favorite features. Andy Tanenbaum (ast@cs.vu.nl) 0, unseen,, *** EOOH *** From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds) Newsgroups: comp.os.minix Subject: Re: Unhappy campers Date: 6 Feb 92 10:33:31 GMT Organization: University of Helsinki In article <12746@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes: > >If Linus wants to keep control of the official version, and a group of eager >beavers want to go off in a different direction, the same problem arises. This is the second time I've seen this "accusation" from ast, who feels pretty good about commenting on a kernel he probably haven't even seen. Or at least he hasn't asked me, or even read alt.os.linux about this. Just so that nobody takes his guess for the full thruth, here's my standing on "keeping control", in 2 words (three?): I won't. The only control I've effectively been keeping on linux is that I know it better than anybody else, and I've made my changes available to ftp-sites etc. Those have become effectively official releases, and I don't expect this to change for some time: not because I feel I have some moral right to it, but because I haven't heard too many complaints, and it will be a couple of months before I expect to find people who have the same "feel" for what happens in the kernel. (Well, maybe people are getting there: tytso certainly made some heavy changes even to 0.10, and others have hacked it as well) In fact I have sent out feelers about some "linux-kernel" mailing list which would make the decisions about releases, as I expect I cannot fully support all the features that will /have/ to be added: SCSI etc, that I don't have the hardware for. The response has been non-existant: people don't seem to be that eager to change yet. (well, one person felt I should ask around for donations so

that I could support it - and if anybody has interesting hardware lying around, I'd be happy to accept it :) The only thing the copyright forbids (and I feel this is eminently reasonable) is that other people start making money off it, and don't make source available etc... This may not be a question of logic, but I'd feel very bad if someone could just sell my work for money, when I made it available expressly so that people could play around with a personal project. I think most people see my point. That aside, if Fred van Kempen wanted to make a super-linux, he's quite wellcome. He won't be able to make much money on it (distribution fee only), and I don't think it's that good an idea to split linux up, but I wouldn't want to stop him even if the copyright let me. >I don't think the copyright issue is really the problem. The problem is >co-ordinating things. Projects like GNU, MINIX, or LINUX only hold together >if one person is in charge. Yes, coordination is a big problem, and I don't think linux will move away from me as "head surgeon" for some time, partly because most people understand about these problems. But copyright /is/ an issue: if people feel I do a bad job, they can do it themselves. Likewise with gcc. The minix copyright, however, means that if someone feels he could make a better minix, he either has to make patches (which aren't that great whatever you say about them) or start off from scratch (and be attacked because you have other ideals). Patches aren't much fun to distribute: I haven't made cdiffs for a single version of linux yet (I expect this to change: soon the patches will be so much smaller than the kernel that making both patches and a complete version available is a good idea - note that I'd still make the whole version available too). Patches upon patches are simply impractical, especially for people that may do changes themselves. >>Where is the sizeable group of people that want to evolve gcc in a way that >>rms/FSF does not approve of? >A compiler is not something people have much emotional attachment to. If >the language to be compiled is a given (e.g., an ANSI standard), there isn't >much room for people to invent new features. An operating system has unlimited >opportunity for people to implement their own favorite features. Well, there's GNU emacs... Don't tell us people haven't got emotional attachment to editors :) Linus 0, unseen,, *** EOOH *** From: laverman@cs.rug.nl (Bert Laverman) Newsgroups: comp.os.minix Subject: Re: Unhappy campers Date: 6 Feb 92 09:41:07

GMT Organization: Dept. of Computing Science, Groningen University Originator: laverman@rug16 Nntp-Posting-Host: rug16 Andy Tanenbaum writes: > >Where is the sizeable group of people that want to evolve gcc in a way that > >rms/FSF does not approve of? > A compiler is not something people have much emotional attachment to. If > the language to be compiled is a given (e.g., an ANSI standard), there isn't > much room for people to invent new features. An operating system has unlimited > opportunity for people to implement their own favorite features. Try posting an announcement in one of the gnu groups about porting FSF software to the Macintosh... ;-) People will be emotional about anything they make, and more so if others try to do unexpected things with it. Software is no exception. If I remeber well, the only real attempt at global coordination of Minix projects has been Glen Overby's Projects directory at plains. Problem is that usually people don't want to announce all of their projects, because they are not sure about whether or when they will finish it. Fred was ready, and he told me he even set up big plans for only sending out his stuff to people who could prove they owned an original. PH was sceptical about his ability to check this, and their ability to check Fred. They were even less enthausiastic about his using the name "Minix" in the name of the final product. sigh. I've seen some of his code. Too bad he only works on PCs. Some of the ideas and comments floating around in the course of the current discussion were fixed by Fred; centralized system call server, dynamically added servers who can then serve new system calls. He ran TCP/IP as an added server. Sure, Fred has an unusual personality, and sometimes an attitude that smells of simply liking to run into a brick wall (not literally, please), but he is one hell of a programmer, and when he was still unemployed he had just the dynamism needed to get a large Minix rework done. I'ld allmost feel sorry for the fact that he has a job now. ;-) Perhaps we can use all this shouting and selfreflecting caused by the Linux wars to some good end. Glen Overby had the right idea in setting up a central site for collecting project info and ideas. What actually is needed is someone to volunteer maintaining such a database. Just saying "dump it here" will only transform anarchy to organized anarchy. Also portability of patches is currently far from ideal. As the recent sighup/init/getty activity shows, we have several packages, some of which are PC only, while others are

ST only, and again others have simply never been tested on a wider range of machines. We have for the FS: - symbolic links, multi-threading For mm: - ?? kernel: - virtual consoles (standard on ST, addon on PC), sighup, the kub scheduler tools: - new init, boot packages, shoelace lib: - ?? commands: - zillions of programs I would like to suggest to those that built a patch set, to try and collect as much as possible info on portability, interference with other major patch sets, reliance on other major patch sets, and include this info with their patch sets. It would greatly help if an enthausiastic ST owner, having found a nice feature for his system, would not be (unpleasantly) surprised by the fact that the patch is only for PCs. Or vice versa of course. Greetings, Bert --- #include <std/disclaimer> Bert Laverman, Dept. of Computing Science, Groningen University Friendly mail to: laverman@cs.rug.nl The rest to: /dev/null -- #include <std/disclaimer> Bert Laverman, Dept. of Computing Science, Groningen University Friendly mail to: laverman@cs.rug.nl The rest to: /dev/null 0, unseen,, *** EOOH *** From: kentd@FtCollins.NCR.com (Kent Dalton) Newsgroups: comp.os.minix Subject: Re: Unhappy campers Date: 5 Feb 92 09:23:56 GMT Organization: NCR Microelectronics, Ft. Collins, CO In-reply-to: ast@cs.vu.nl's message of 3 Feb 92 22:46:40 GMT >>>>> On 3 Feb 92 22:46:40 GMT, ast@cs.vu.nl (Andrew "Dice" Tanenbaum) said: Andy> As an aside, for those folks who don't read news headers, Linus is Andy> in Finland and I am in The Netherlands. Are we reaching a Andy> situation where another critical industry, free software, that had Andy> been totally dominated by the U.S. is being taken over by the Andy> foreign competition? I prefer to think of it as indicative of the fact that the U.S. is still the pioneer when it comes to new ideas and technologies. The rest of the world just follows along, trying to refine and capitalize upon our new developments. :^) Andy> Will we soon see President Bush coming to Europe with Richard Andy> Stallman and Rick Rashid in tow, demanding that Europe import more Andy> American free software? Be glad software isn't as important as oil. ;^) --
/**********/ /* Kent Dalton * EMail: Kent.Dalton@FtCollinsCO.NCR.COM */ /* NCR Microelectronics

* CIS: 72320,3306 */ /* 2001 Danfield Ct. MS470A * */ /* Fort Collins, Colorado 80525 * (303) 223-5100 X-319 */ /**********/ I've got an IDEA!! Why don't I STARE at you so HARD, you forget your SOCIAL SECURITY NUMBER!! 0, unseen,, *** EOOH *** From: nao@sala.sony.co.jp (HAMADA Naoki) Newsgroups: comp.os.minix Subject: Re: Unhappy campers Date: 6 Feb 92 12:08:51 GMT Reply-To: nao@sala.sony.co.jp Organization: Science Art Laboratory, Tokyo, Japan. In-Reply-To: ast@cs.vu.nl's message of 3 Feb 92 22:46:40 GMT Nntp-Posting-Host: skyline Greetings! In article <12667@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes: (most stuffs omitted) > If you don't have that journal, you can FTP the paper from > ftp.cs.vu.nl (192.31.231.42) in directory amoeba/papers as comp_sys.tex.Z > (compressed TeX source) or comp_sys.ps.Z (compressed PostScript). I am so much interested in this paper, but I cannot get it. The ftp server of ftp.cs.vu.nl rejects my connection because the IP address of machines in my site (and most sites in Japan), from machines outside of Japan, cannot be resolved by ptr. Is there anyone who has fetched it and placed it in annonymous ftp? Archie servers seem to have no idea. Thanks in advance. -HAMADA Naoki Naoki is my first name. nao@sala.sony.co.jp 0, unseen,, *** EOOH *** From: awoodhull@hamp.hampshire.edu Newsgroups: comp.os.minix Subject: Minix for 8088? Yes! Date: 4 Feb 92 15:10:46 GMT Organization: Hampshire College Subject: Minix for 8088? Yes! Some of the correspondence about Minix vs. Linux has argued that the 8088 compatibility of Minix is no longer important, because 80386 machines are not that much more expensive now. That may be true when you talk about new machines, but students are also in the market for used machines. I got a system quite competent to run Minix, a PC clone with two 360K floppy drives, for $75 recently at a yard sale. You aren't going to find second hand '386s for a while yet. Like ast I am also a professor. I expect to be teaching an OS course using Minix in Nicaragua next year. There, as in many other places outside the over-developed world, availability of entry-level or used computer equipment for student use is the best that most students can hope for.

For the purpose for which it was created, continued compatibility of Minix with low-end hardware is definitely worth maintaining, and I hope ast will not change his target. Linux also sounds like a nice experiment, for a different purpose. I may decide to play with it, too, after I get Minix working on my new '486. I don't think I will use Linux in my teaching, however. Albert S. Woodhull School of Natural Science, Hampshire College, Amherst, MA 01002 awoodhull@hamp.hampshire.edu, woodhull@dawn.hampshire.edu 0, unseen,, *** EOOH *** From: dmiller@acg.uucp (David Miller) Newsgroups: comp.os.minix Subject: Linux is Obsolete and follow up postings Date: 3 Feb 92 01:03:46 GMT Distribution: all Organization: AppliedComputerGroup As an observer interested in operating system design, I couldn't resist this thread. Please realize that I am not really experienced with minux or linux: I have been into unix for many years. First, a few observations: Minix was written to be an educational tool for ASTs' classes, not a commercial operating system. It was never a design parameter to have it run freely available source code for unix systems. I think it was also a statement of how operating systems should be designed, with a micro kernel and seperate processes covering as much of the required functionality as possible. Linux was written mostly as a learning exercise on Linus part - how to program the 386 family. Designing the ultimate operating system was not an objective. Providing a usable, free platform that would run all sorts of widely available free software was a consideration, and one that appears to have been well met. Criticism from anyone that either of these systems isn't what *they* would like it to be is misplaced. After all, anybody that has a computer that will run either system is free to do what Linus and Andrew did: write your own! I, for one, applaud Linus for his considerable effort in developing Linux and his decision to make it free to everybody. I applaud AST for his effort to make minix affordable - I have real trouble relating to complaints that minix isn't free. If you can afford the time to explore minix, and a basic computer system, $150 is not much more - and you do get a book to go with it. Next, a few questions for the professor: Is minix supposed to be a "real operating system" or an educational tool ? As an educational tool it is an excellent work. As a real operating system it presents some terribly rough edges (why no malloc() ?, just for starters) My feeling from

reading The Book and listening to postings here is that you wanted a tool to teach your classes, and a lot of others wanted to play with an affordable operating system. These others have been trying to bolt on enough features to make it a "real operating system", with less than outstanding success. Why split fundemental os functions, such as memory management, into user processes? As all good *nix gurus know, the means to success is to divide and conquer, with the goal being to *simplify* the problem into managable, well defined components. If splitting basic parts of the operating system into user space processes complicates the function by introducing additional mechanisms (message passing, complicated signals), have we met the objective of simplifying the design and implementation? I agree that *nix has suffered a bad case of feature-itis - especially sysVr4. Perhaps the features that people want for either functionality or compatibility could be offered by run-time loadable modules/libraries that offer these features. The micro-kernel would still be a base-level resource manager that also routes function requests to the appropriate module/library. The modules could be threads or user processes. (I think - os hackers please correct me :-)) Just my $.04 worth - please feel free to post or email responses. I have no formal progressive training in computer science, so I am really asking these questions in ignorance. I suspect a lot of others on the net have similar questions in their own minds, but I've been wrong before. -- -- David David Miller Phone: (207) 873-3317 Applied Computer Group Fax: (207) 872-5018 5 Kimball Street UUCP: uunet!acg!dmiller 0, unseen,, *** EOOH *** From: rdc30@nmrdc1.nmrdc.nnmc.navy.mil (LCDR Michael E. Dobson) Newsgroups: comp.os.minix Subject: Re: Fred returning from the dead <was Unhappy campers> Date: 6 Feb 92 13:18:22 GMT Organization: Naval Medical Research & Development Command In article <1992Feb4.172131.18145@philce.ce.philips.nl> meulenbr@ce.philips.nl (Frans Meulenbroeks) writes: > >ast@cs.vu.nl (Andy Tanenbaum) writes: > >>Suppose Fred van Kempen returns from the dead and wants to take over, creating > >Just a remark for the readers. As far as I know Fred is not physically dead. >If english is not your native language (just like mine) you might >conclude otherwise from ast's words. Fred is only electronically dead. > He's not electronicly dead any longer either. I and several others involved with

MUGNET have received electronic communications from him, that is unless our hosts are haunted by Fred's electronic spirit. -- Mike Dobson, Sys Admin for | Internet: rdc30@nmrdc1.nmrdc.nnmc.navy.mil nmrdc1.nmrdc.nnmc.navy.mil | UUCP: ...uunet!mimsy!nmrdc1!rdc30 AT&T 3B2/600G Sys V R 3.2.2 | BITNET: dobson@usuhsb or nrd0mxd@vmnmdsc WIN/TCP for 3B2 | MCI-Mail: 377-2719 or 0003772719@mcimail.com 0, unseen,, *** EOOH *** From: michael@gandalf.informatik.rwth-aachen.de (Michael Haardt) Newsgroups: comp.os.minix Subject: 1.6.17 summary and why I think AST is right. Date: 6 Feb 92 20:07:25 GMT Reply-To: u31b3hs@messua.informatik.rwth-aachen.de (Michael Haardt) Organization: Gandalf - a 386-20 machine Nntp-Posting-Host: kaa I will first give a summary of what you can expect from MINIX in *near* future, and then explain why I think AST is right. Some time ago, I asked for details about the next MINIX release (1.6.17). I got some response, but only from people running 1.6.16. The following informations are not official and may be wrong, but they are all I know at the moment. Correct me if something is wrong: - The 1.6.17 patches will be relative to 1.5 as shipped by PH. - The header files are clean. - The two types of filesystems can be used together. - The signal handling is rewritten for POSIX. The old bug is removed. - The ANSI compiler (available from Transmediar, I guess) comes with compiler binaries and new libraries. - There don't seem to be support for the Amoeba network protocol. - times(2) returns a correct value. termios(2) is implemented, but it's more a hack. I don't know if "implemented" means in the kernel, or the current emulation. - There is no documentation about the new filesystem. There is a new fsck and a new mkfs, don't know about de. - With the ANSI compiler, there is better floating point support. - The scheduler is improved, but not as good as written by Kai-Uwe Bloem. I asked these things to get facts for the decision if I should upgrade to MINIX 1.6.17 or to Linux after the examens are over. Well, the decision is made: I will upgrade to Linux at the end of the month and remove MINIX from my winchester, when Linux runs all the software I need and which currently runs under MINIX 1.5 with heavy patches. I guess this may take up to two months. These are the main reasons for my decision: - There is no

"current" MINIX release, which can be used as basis for patches and nobody knows, when 1.6.17 will appear. - The library contains several bugs and from what I have heard, there is no work done at them. There will not be a new compiler, and the 16 bit users still have to use buggy ACK. - 1.6.17 should offer more POSIX, but a complete termios is still missing. - I doubt that there is still much development for 16 bit users. I think I will stop maintaining the MINIX software list in a few months. Anyone out there, who would like to continue it? Until Linux runs *perfect* on my machine, each update of Origami will still run on 16-bit MINIX. I will announce when the last of these versions appears. In my opinion, AST is right in his decision about MINIX. I read the flame war and can't resist to say that I like MINIX the way it is, now where there is Linux. MINIX has some advantages: - You can start playing with it without a winchester, you can even compile programs. I did this a few years ago. - It is so small, you don't need to know much to get a small system which runs ok. - There is the book. Ok, only for version 1.3, but most of it is still valid. - MINIX is an example of a non-monolithic kernel. Call it a microkernel or a hack to overcome braindamaged hardware: It demonstrates a concept, with its pros and cons -- a documented concept. In my eyes, it is a nice system for first steps in UNIX and systems programming. I learned most of what I know about UNIX with MINIX, in all areas, from programming in C under UNIX to system administration (and security holes:) MINIX grew with me: 1.5.xx upgrades, virtual consoles, mail & news, text processing, crosscompiling etc. Now it is too small for me. I don't need a teaching system anymore, I would like to get a more complicated and featureful UNIX, and there is one: Linux. Back in the old days, v7 was state of the art. There was MINIX which offered most of it. In one or two years, POSIX is what you are used to see. Hopefully, there will be MINIX, offering most of it, with a new book, for people who want to run a small system to play and experiment with. Stop flaming, MINIX and Linux are two different systems with different purposes. One is a teaching tool (and a good one I think), the other is real UNIX for real hackers. Michael 0, unseen,, *** EOOH *** From: dingbat@diku.dk (Niels Skov Olsen) Newsgroups: comp.os.minix Subject: Re: 1.6.17 summary and why I think AST is right. Date: 10 Feb 92 17:33:39 GMT Organization: Department of

Computer Science, U of Copenhagen
michael@gandalf.informatik.rwth-aachen.de (Michael Haardt) writes:
>Stop flaming, MINIX and Linux are two different systems with different >purposes. One is a teaching tool (and a good one I think), the other is >real UNIX for real hackers. Hear, hear! And now Linux articles in alt.os.linux (or comp.os.misc if your site don't receive alt.*) and Minix articles here. eoff (end of flame fest :-) Niels

---

Kapitel 9. Historische Dokumente

8.8. GNU Free Documentation License | Kapitel 10. Impressum

Kapitel 10. Impressum

Kapitel 9. Historische Dokumente Stichwortverzeichnis

Frank Ronneburg Luckauer Strasse 2 10969 Berlin E-Mail: fr@debiananwenderhandbuch.de

Haftungshinweis: Trotz sorgfältiger inhaltlicher Kontrolle übernehmen wir keine Haftung für die Inhalte externer Links. Für den Inhalt der verlinkten Seiten sind ausschließlich deren Betreiber verantwortlich.

Kapitel 10. Impressum

Kapitel 9. Historische Dokumente Stichwortverzeichnis

Stichwortverzeichnis

Kapitel 10. Impressum

Symbole

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Personen

Q

R

S

T

Informationen zu Freier Software, Freie Software / Open Source

Installation von Paketen überwachen, Status-Anzeige

Jedes Zeichen zählt..., tar - Komprimieren der Archive

Kommandovervollständigung, APT und Verwandte

Lange Texte ansehen, tar - Archivieren von Dateien

Nicht-Entwickler Upload, Upload von Paketen

PDF-Dateien bearbeiten mit Vim, vi für Fortgeschrittene

Plattenplatzbelegung in Farbe, cfdisk und mount - Einbinden eines Dateisystems

Platzbedarf von Task-Paketen, tasksel

Rekonfigurierbare Pakete ermitteln, configure-debian

Setzen von Umgebungsvariablen, Internationalisierung und Lokalisierung

Suchen ohne apt-cache, apt-cache

Suchen von nicht installierten Dateien, dlocate

tar zum Plaudern bringen..., tar - Entpacken von Dateien

Testen eines neuen Kernels mit GRUB, Kommandozeile

Verschiedene Dateien gleichzeitig bearbeiten, Programmstart

Versteckte Dateien, ls - Auflisten von Dateien und Verzeichnissen

Virtuelle Pakete ermitteln, Das Debian Paketformat - .deb

Webbasierte Tag-Suche, Debian Package Tags (debtags)

Zeilenumbruch entfernen, Verschiedenes

touch, Einige Beispiele

Toy Story, Die Geschichte von Debian, Kodenamen

Treiber, GRUB

Tripwire, Integrität des Dateisystems

tunefs, Festplattenpartitionen

Tux, Und dieser Pinguin?

tzconfig, base-config

U

V

W

X

Y

Z

Stichwortverzeichnis

Kapitel 10. Impressum

Impressum

Unterstützen Sie die Arbeit an diesem Buch:

spenden

Debian Links

Debian Projekt

Debian WN

Debian Forum

Debian-news