

Die Debian GNU/Linux-FAQ

Die Autoren finden Sie in Autoren der Debian-FAQ

Version 8.1, 28 August 2016

Zusammenfassung

Dieses Dokument beantwortet häufig gestellte Fragen über Debian GNU/Linux.

Copyright-Hinweis

Copyright © 1996-2016 Software in the Public Interest, portions copyright © 2004, 2005, 2006 Kamaraju Kusumanchi

Permission is granted to make and distribute verbatim copies of this document provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this document under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this document into another language, under the above conditions for modified versions, except that this permission notice may be included in translations approved by the Free Software Foundation instead of in the original English.

Inhaltsverzeichnis

1	Definitionen und Überblick	1
1.1	Was ist diese FAQ?	1
1.2	Was ist Debian GNU/Linux?	1
1.3	OK, jetzt weiß ich, was Debian ist ... aber was ist Linux?!	2
1.4	Befasst sich Debian nur mit GNU/Linux?	2
1.5	Was ist der Unterschied zwischen Debian GNU/Linux und anderen Linux-Distributionen? Warum sollte ich Debian wählen statt einer anderen Distribution?	2
1.6	Wie fügt sich das Debian-Projekt in das GNU-Projekt der Free Software Foundation ein und lässt es sich damit vergleichen?	3
1.7	Wie spricht man Debian aus und was bedeutet das?	3
2	Debian GNU/Linux bekommen und installieren	5
2.1	Was ist die aktuelle Version von Debian?	5
2.2	Gibt es Paketaktualisierungen in »Stable«?	5
2.3	Woher/wie kann ich mir die Debian-Installations-Images beschaffen?	6
2.4	Wie kann ich Debian von CD-ROM installieren?	6
2.5	Warum enthält eine offiziell als Stable veröffentlichte CD-ROM symbolische Links für »Frozen« und »Unstable«? Ich dachte, diese CD enthält »Stable«!	6
2.6	Kann ich Debian direkt über das Internet bekommen und installieren?	6
2.7	Gibt es alternative Strategien, um den Debian Installer zu booten?	6
3	Eine Debian-Distribution auswählen	7
3.1	Welche Debian-Distribution (Stable/Testing/Unstable) ist geeignet für mich?	7
3.1.1	Sie haben mir empfohlen, Stable zu installieren. Allerdings wird unter Stable bestimmte Hardware nicht erkannt oder funktioniert nicht richtig. Was soll ich tun?	7
3.1.2	Gibt es unterschiedliche Versionen eines Pakets in unterschiedlichen Distributionen?	8
3.1.3	Die stabile Distribution enthält wirklich eine ganze Menge ziemlich veralteter Pakete, wie an Kde, Gnome, Xorg oder sogar dem Kernel deutlich erkennbar ist. Warum ist das so?	8
3.1.4	Wenn ich zu einer anderen Distribution wechseln wollte, ginge das?	8
3.1.5	Sollte man Stable, Testing oder Unstable installieren?	8
3.1.6	Hier ist die Rede davon, dass Testing »kaputt« gehen kann. Was soll das heißen?	9
3.1.7	Wie kann es sein, dass Testing monatelang beschädigt ist? Würden die in Unstable eingepflegten Fehlerkorrekturen nicht direkt in Testing einfließen?	9
3.1.8	Vom Standpunkt eines Systemverwalters her gesehen: Welche Distribution verlangt mehr Aufmerksamkeit?	10
3.1.9	Was passiert, wenn eine neue Version von Debian veröffentlicht wird?	10

3.1.10	Ich habe einen Desktop-Rechner/Rechner-Cluster mit Debian. Wie kann ich herausfinden, welche Distribution darauf läuft?	10
3.1.11	Augenblicklich verwende ich Stable. Kann ich zu Testing oder Unstable wechseln? Und wenn, wie?	11
3.1.12	Im Augenblick verwende ich Testing (stretch). Was wird passieren, wenn eine neue Veröffentlichung freigegeben wird? Werde ich weiterhin Testing haben oder wird auf meinem Rechner die neue Stable-Distribution laufen?	11
3.1.13	Ich bin noch immer verwirrt. Was sagten Sie, soll ich installieren?	11
3.2	Aber was ist mit Knoppix, Linux Mint Debian Edition, Ubuntu und all den anderen Distributionen?	12
3.2.1	Ich weiß jetzt, dass Knoppix/Linux Mint Debian Edition/Ubuntu/ . . . auf Debian basieren. Bedeutet das auch, dass ich die »apt«-Paketwerkzeuge verwenden kann, wenn ich eine dieser Distributionen auf meiner Festplatte installiere?	12
3.2.2	Ich habe Knoppix/Linux Mint Debian Edition/Ubuntu/ . . . auf meiner Festplatte installiert. Jetzt habe ich ein Problem. Was soll ich tun?	12
3.2.3	Ich benutze Knoppix/LMDE/Ubuntu/ . . . und möchte nun Debian verwenden. Wie steige ich um?	12
4	Kompatibilitätsfragen	13
4.1	Auf welchen Hardware-Architekturen/Systemen läuft Debian GNU/Linux?	13
4.2	Welche Kernel beinhalten die Distributionen von Debian GNU/Linux?	14
4.3	Wie kompatibel ist Debian mit anderen Linux-Distributionen?	14
4.4	In welchem Maße ist der Quellcode von Debian mit anderen Unix-Systemen kompatibel?	14
4.5	Kann ich Debian-Pakete (».deb«-Dateien) auf meinem Red Hat-/Slackware-/ . . . Linux-System verwenden? Kann ich Red Hat-Pakete (».rpm«-Dateien) auf meinem Debian GNU/Linux-System einsetzen?	15
4.6	Wie soll ich ein nicht-Debian-Programm installieren?	15
4.7	Warum kann ich keine Programme kompilieren, die libtermcap benötigen?	16
4.8	Warum kann ich AccelX nicht installieren?	16
5	Für Debian-Systeme verfügbare Software	17
5.1	Welche Art von Anwendungen und Entwicklungs-Software ist für Debian GNU/Linux verfügbar?	17
5.2	Wer schreibt die ganzen Programme?	17
5.3	Wie kann ich eine aktuelle Liste der Programme, die für Debian paketiert wurden, bekommen?	17
5.4	Wie kann ich eine Entwicklungsumgebung installieren, um Pakete zu bauen?	18
5.5	Was fehlt bei Debian GNU/Linux?	18
5.6	Wieso bekomme ich die Nachricht »ld: cannot find -lfoo«, wenn ich Programme kompiliere? Warum gibt es keine libfoo.so-Dateien in den Debian-Bibliothekspaketen?	18
5.7	(Wie) Unterstützt Debian Java?	18
5.8	Wie kann ich prüfen, ob ich ein Debian-System benutze und welche Version es ist?	19
5.9	Wie unterstützt Debian andere Sprachen als Englisch?	19
5.10	Wo sind ezmlm/djbdns/qmail?	19
5.11	Wo ist der Player für Flash (SWF)?	20
5.12	Wo ist Google Earth?	20
5.13	Wo sind die VoIP-Programme?	20
5.14	Ich habe eine Karte für drahtloses Netzwerk, die aber nicht mit Linux funktioniert. Was soll ich tun?	20

6 Die Debian FTP-Archive	21
6.1 Wie viele Debian-Distributionen gibt es?	21
6.2 Was haben all diese Namen wie Etch, Lenny usw. zu bedeuten?	21
6.2.1 Welche Codenamen wurden in der Vergangenheit verwendet?	21
6.2.2 Woher stammen diese Codenamen?	21
6.3 Was ist mit »Sid«?	22
6.4 Was enthält das stable-Verzeichnis?	22
6.5 Was enthält das testing-Verzeichnis?	23
6.5.1 Wie erhält Testing den »frozen«-Status?	23
6.6 Was enthält das unstable-Verzeichnis?	23
6.7 Was haben all die Verzeichnisse in den Debian-FTP-Archiven zu bedeuten?	23
6.8 Was haben die ganzen Verzeichnisse in <code>dists/stable/main</code> zu bedeuten?	24
6.9 Wo befindet sich der Quellcode?	24
6.10 Was befindet sich in dem <code>pool</code> -Verzeichnis?	24
6.11 Was ist »Incoming«?	25
6.12 Wie erstelle ich mein eigenes, apt-taugliches Paketdepot?	25
7 Grundlagen des Debian-Paketverwaltungssystems	27
7.1 Was ist ein Debian-Paket?	27
7.2 Wie ist ein binäres Debian-Pakets aufgebaut?	28
7.3 Warum sind Debian-Paketdateinamen so lang?	28
7.4 Was ist eine Debian-control-Datei?	28
7.5 Was ist ein Debian »conffile«?	29
7.6 Was sind Debians »preinst«, »postinst«, »prerm«- und »postrm«-Skripte?	29
7.7 Was ist ein <i>Essential</i> -, <i>Required</i> -, <i>Important</i> -, <i>Standard</i> -, <i>Optional</i> - oder <i>Extra</i> -Paket?	30
7.8 Was ist ein virtuelles Paket?	30
7.9 Was ist damit gemeint, dass ein Paket eine <i>Depends</i> -, <i>Recommends</i> -, <i>Suggests</i> -, <i>Conflicts</i> -, <i>Replaces</i> - oder <i>Provides</i> -Abhängigkeit zu einem anderen Paket hat?	31
7.10 Was bedeutet Pre-Depends (Vor-Abhängigkeit)?	31
7.11 Was bedeutet <i>unknown</i> , <i>install</i> , <i>remove</i> <i>purge</i> und <i>hold</i> im Paket-Status?	31
7.12 Wie setze ich ein Paket auf »hold« (zurückhalten)?	32
7.13 Wie installiere ich ein Quellpaket?	32
7.14 Wie baue ich Binärpakete aus einem Quellpaket?	33
7.15 Wie kann ich selbst Debian-Pakete erstellen?	33
8 Die Debian-Paketverwaltungswerkzeuge	35
8.1 Welche Programme bietet Debian zur Paketverwaltung an?	35
8.1.1 <code>dpkg</code>	35
8.1.2 <code>APT</code>	36
8.1.3 <code>aptitude</code>	37
8.1.4 <code>synaptic</code>	38
8.1.5 <code>tasksel</code>	38
8.1.6 Andere Paketverwaltungswerkzeuge	38

8.2	Angeblich ist es mit Debian möglich, ein derzeit laufendes Programm zu aktualisieren. Wie funktioniert das?	38
8.3	Wie kann ich feststellen, welche Pakete bereits auf einem Debian-System installiert sind?	39
8.4	Wie kann ich alle Dateien auflisten, die in einem installierten Paket enthalten sind?	39
8.5	Wie kann ich herausfinden, welches Paket eine bestimmte Datei angelegt hat?	39
8.6	Warum wird »foo-data« nicht entfernt, wenn ich »foo« deinstalliere? Wie stelle ich sicher, dass alte unbenutzte Bibliothekspakete restlos entfernt werden?	40
9	Wie man sein Debian-System auf aktuellem Stand hält	41
9.1	Wie kann ich mein Debian-System aktuell halten?	41
9.1.1	aptitude	41
9.1.2	apt-get und apt-cdrom	42
9.1.3	mirror	42
9.2	Muss ich in den Einzelbenutzermodus wechseln, um ein Paket zu aktualisieren?	43
9.3	Muss ich all diese .deb-Archivdateien auf meinem Rechner aufbewahren?	43
9.4	Wie kann ich protokollieren, welche Pakete ich meinem System hinzugefügt habe? Ich möchte gerne wissen, wann welche Pakete aktualisiert oder entfernt wurden!	43
9.5	Kann ich das System automatisch aktualisieren lassen?	43
9.6	Ich habe mehrere Rechner; wie kann ich erreichen, dass ich die Updates nur einmal herunterladen muss?	44
10	Debian und der Kernel	45
10.1	Kann ich einen Kernel ohne gewisse Debian-spezifische Optimierungen installieren und kompilieren?	45
10.2	Welche Werkzeuge zum Erzeugen eines angepassten Kernels stellt Debian zur Verfügung?	45
10.3	Welche Hilfsmittel stellt Debian für den Umgang mit Modulen bereit?	45
10.4	Kann ich sicher ein altes Kernel-Paket deinstallieren und wenn ja, wie?	45
10.5	Wo finde ich mehr Informationen über Linux-Pakete für Debian?	46
11	Anpassen Ihres Debian GNU/Linux-Systems	47
11.1	Wie kann ich sicherstellen, dass alle Programme dieselbe Papiergröße verwenden?	47
11.2	Wie lassen sich ohne Abstriche bei der Sicherheit Peripheriegeräte zur Verfügung stellen?	47
11.3	Wie kann ich eine Konsolen-Schriftart auf Debian-Art beim Systemstart laden?	47
11.4	Wie kann ich die Standardeinstellung eines X11-Programms konfigurieren?	47
11.5	Jede Distribution scheint eine andere Boot-Methode zu haben. Beschreiben Sie jene von Debian.	48
11.6	Welche Möglichkeiten außer rc.local gibt es zum Anpassen des Bootprozesses?	48
11.7	Wie geht das Paketverwaltungssystem mit Paketen um, die Konfigurationsdateien für andere Pakete enthalten?	49
11.8	Wie kann ich eine durch ein Paket installierte Datei außer Kraft setzen, so dass stattdessen eine andere Version verwendet werden kann?	49
11.9	Wie kann ich meine lokal erstellten Pakete in die Liste der verfügbaren Pakete, die dem Paketmanagementsystem bekannt sind, aufnehmen?	50
11.10	Einige Benutzer mögen mawk, andere gawk; einige mögen vim, andere elvis; einige trn, wieder andere tin; wie unterstützt Debian die Vielfalt?	50
12	Unterstützung für Debian GNU/Linux erhalten	53
12.1	Welche andere Dokumentation gibt es auf einem und für ein Debian-System?	53
12.2	Wo und wie kann man sich online über Debian austauschen?	54
12.2.1	Mailinglisten	54

12.2.2	Web-Foren	54
12.2.3	Wiki	55
12.2.4	Betreuer	55
12.2.5	Usenet-Nachrichtengruppen (Newsgroups)	55
12.3	Gibt es einen schnellen Weg, um nach Informationen über Debian GNU/Linux zu suchen?	55
12.4	Gibt es Protokolle von bekannten Fehlern?	55
12.5	Wie melde ich einen Fehler in Debian?	55
13	Zum Debian-Projekt beitragen	57
13.1	Wie kann ich Debian-Mitglied/Debian-Entwickler werden?	57
13.2	Wie kann ich Ressourcen zum Debian-Projekt beisteuern?	57
13.3	Wie kann ich das Debian-Projekt finanziell unterstützen?	57
13.3.1	Software in the Public Interest	58
13.3.2	Weitere Organisationen	58
14	Debian GNU/Linux in einem kommerziellen Produkt verwenden	59
14.1	Darf ich Debian-CDs erstellen und verkaufen?	59
14.2	Kann Debian mit unfreier Software (non-free) paketierr werden?	59
14.3	Ich erstelle eine spezielle Linux-Distribution für einen speziellen Anwendungsbereich. Kann ich Debian GNU/Linux als Linux-Grundsystem verwenden und meine eigenen Programme hinzufügen?	59
14.4	Kann ich mein kommerzielles Programm in einem Debian-»Paket« ablegen, damit es auf jedem Debian-System mühelos installiert werden kann?	60
15	Zu erwartende Änderungen in der nächsten Hauptversion von Debian	61
15.1	Härten des Systems	61
15.2	Erweiterte Unterstützung für nicht Englisch sprechende Benutzer	61
15.3	Verbesserungen im Debian Installer	62
15.4	Weitere (Rechner-)Architekturen	62
15.5	Weitere Kernel	62
16	Allgemeine Informationen über die FAQ	63
16.1	Autoren	63
16.2	Feedback	63
16.3	Verfügbarkeit	63
16.4	Dokumentenformat	64

Kapitel 1

Definitionen und Überblick

1.1 Was ist diese FAQ?

Dieses Dokument enthält häufig gestellte Fragen (sowie deren Antworten!) über die Debian-Distribution (Debian GNU/Linux und andere) und das Debian-Projekt. Wenn möglich, wird auf andere Dokumentation verwiesen; wir vermeiden es, größere Abschnitte externer Dokumentation in diesem Dokument wiederzugegeben. Sie werden bemerken, dass einige Antworten etwas Wissen über Unix-ähnliche Betriebssysteme voraussetzen. Es wird versucht, so wenig Vorwissen wie möglich vorauszusetzen, Antworten auf allgemeine Anfängerfragen sind einfach gehalten.

Wenn Sie das, was Sie suchen, in dieser FAQ nicht finden, werfen Sie einen Blick in 'Welche andere Dokumentation gibt es auf einem und für ein Debian-System?' auf Seite 53. Wenn selbst das nicht hilft, wird auf 'Feedback' auf Seite 63 verwiesen.

1.2 Was ist Debian GNU/Linux?

Debian GNU/Linux ist eine bestimmte *Distribution* des Linux-Betriebssystems und zahlreicher Pakete, die darunter laufen.

Debian GNU/Linux ist:

- **voll ausgestattet:** Debian enthält zur Zeit mehr als 42551 Software-Pakete. Die Anwender können wählen, welche Pakete installiert werden sollen; Debian enthält ein Werkzeug für diesen Zweck. Eine Liste der derzeit in Debian verfügbaren Pakete und deren Beschreibungen finden Sie auf jedem Debian-Spiegelserver (<http://www.debian.org/distrib/ftplist>).
- **frei verwendbar und verteilbar:** Um sich an der Entwicklung und Verteilung zu beteiligen, sind weder Zahlungen noch die Mitgliedschaft in einer Vereinigung erforderlich. Alle Pakete, die formal Teil von Debian GNU/Linux sind, können frei weitergegeben werden, normalerweise unter den Bedingungen der GNU General Public License.
Die FTP-Archive von Debian enthalten auch ungefähr 696 Software-Pakete (in den `non-free`- und `contrib`-Bereichen), die unter den speziellen Bedingungen des jeweiligen Pakets weitergegeben werden können.
- **dynamisch:** Mit ungefähr 1033 Freiwilligen, die ständig neuen und verbesserten Code beisteuern, entwickelt sich Debian rasant weiter. Die FTP-Archive werden zweimal täglich aktualisiert.

Die meisten Linux-Benutzer verwenden eine bestimmte Linux-*Distribution* wie etwa Debian GNU/Linux. Im Prinzip könnten Anwender sich den Linux-Kernel jedoch auch selbst aus dem Internet oder andernorts besorgen und selbst kompilieren. Anschließend könnten sie sich den Quelltext vieler Anwendungen auf dem selben Weg besorgen, die Programme kompilieren und diese dann auf ihren Systemen installieren. Dieses Vorgehen ist für komplexe Anwendungen zeitaufwändig und fehleranfällig. Daher entscheiden sich Anwender oft dafür, das Betriebssystem und die Programmpakete von einem der Linux-Distributoren zu beziehen. Was die einzelnen Linux-Distributoren voneinander unterscheidet, sind die Software, Protokolle und ihre Gepflogenheiten im Umgang mit Paketierung, Installation und der Verfolgung der auf den Anwendersystemen installierten Programmpakete, kombiniert mit Installations- und Wartungswerkzeugen, Dokumentation und anderen Dienstleistungen.

Debian GNU/Linux ist das Ergebnis der Anstrengungen Freiwilliger, ein durch die Zusammenstellung von Anwendungsprogrammen komplettiertes, qualitativ hochwertiges, freies und mit Unix kompatibles Betriebssystem zu schaffen. Die Idee eines freien Unix-ähnlichen Betriebssystems stammt ursprünglich vom GNU-Projekt und viele der Anwendungen, die Debian GNU/Linux ausmachen, wurden vom GNU-Projekt entwickelt.

In ihrem Verständnis von »frei« stimmen das Debian- und das GNU-Projekt überein (siehe Debian-Richtlinien für Freie Software (http://www.debian.org/social_contract#guidelines)). Wenn wir von freier Software sprechen, meinen wir damit die Freiheit und nicht den Preis. Freie Software bedeutet, dass Sie die Freiheit haben, Kopien freier Software zu vertreiben, dass Sie den Quelltext bei Bedarf bekommen können, dass Sie die Software verändern oder Teile davon in neuen freien Programmen verwenden dürfen; und dass Sie wissen, dass Ihnen all dies erlaubt ist.

Das Debian-Projekt wurde 1993 von Ian Murdock gegründet, anfangs unter der Schirmherrschaft des GNU-Projekts der Free Software Foundation. Heutzutage betrachten es die Debian-Entwickler als direkten Nachkommen des GNU-Projekts.

Obwohl Debian GNU/Linux selbst freie Software ist, verwenden es andere Linux-Distributoren entsprechend ihrer Ziele als Grundlage. Indem ein zuverlässiges, voll ausgestattetes Basissystem zur Verfügung gestellt wird, bietet Debian Anwendern von Linux erhöhte Kompatibilität und den Herstellern von Linux-Distributionen die Möglichkeit, unnötigen Aufwand zu vermeiden und sich auf die Dinge zu konzentrieren, die ihre Distribution besonders machen. Für weitere Informationen dazu lesen Sie bitte 'Ich erstelle eine spezielle Linux-Distribution für einen speziellen Anwendungsbereich. Kann ich Debian GNU/Linux als Linux-Grundsystem verwenden und meine eigenen Programme hinzufügen?' auf Seite 59.

1.3 OK, jetzt weiß ich, was Debian ist ... aber was ist Linux?!

Kurz gesagt stellt Linux den Kernel eines Unix-ähnlichen Betriebssystems dar. Es wurde ursprünglich für 386er und darauf folgende Prozessoren in Personal Computern entwickelt; heutzutage läuft Linux auch auf einem Dutzend anderer Systeme. Linux wird von Linus Torvalds und vielen weiteren Informatikern auf der ganzen Welt geschrieben.

Abgesehen von seinem Kernel hat ein »Linux«-System üblicherweise:

- eine Verzeichnisstruktur gemäß dem Linux Filesystem Hierarchy Standard <http://www.pathname.com/fhs/>.
- eine große Auswahl an Unix-Werkzeugen, von denen viele vom GNU-Projekt und der Free Software Foundation entwickelt wurden.

Die Kombination aus Linux-Kernel, der Verzeichnisstruktur sowie den GNU-, FSF- und sonstigen Werkzeugen zielt darauf ab, dem POSIX (IEEE 1003.1)-Standard zu entsprechen; siehe 'In welchem Maße ist der Quellcode von Debian mit anderen Unix-Systemen kompatibel?' auf Seite 14.

Für weitere Informationen über Linux lesen Sie What is Linux (<http://www.linux.org/info/>) von Linux Online (<http://www.linux.org/>).

1.4 Befasst sich Debian nur mit GNU/Linux?

Derzeit ist Debian nur für Linux erhältlich, aber mit Debian GNU/Hurd und Debian auf BSD-Kerneln haben wir begonnen, auch nicht-Linux-basierte Betriebssysteme als Entwicklungs-, Server- und Desktop-Plattform anzubieten. Allerdings sind diese nicht-Linux-Portierungen derzeit noch nicht offiziell verfügbar.

Das älteste Portierungs-Bestreben ist Debian GNU/Hurd.

Hurd ist eine Ansammlung von Servern, die auf dem GNU Mach-Microkernel aufsetzen. Zusammen stellen sie die Basis des GNU-Betriebssystems dar.

Unter <http://www.gnu.org/software/hurd/> finden Sie weitere Informationen über GNU/Hurd im Allgemeinen, unter <http://www.debian.org/ports/hurd/> weitere Informationen über Debian GNU/Hurd.

Ein weiteres Bestreben ist die Portierung auf einen BSD-Kernel; dabei wird mit dem FreeBSD-Kernel gearbeitet.

Weitere Informationen zu diesen nicht-Linux-Portierungen finden Sie unter <http://www.debian.org/ports/#nonlinux>.

1.5 Was ist der Unterschied zwischen Debian GNU/Linux und anderen Linux-Distributionen? Warum sollte ich Debian wählen statt einer anderen Distribution?

Folgende Schlüsselmerkmale unterscheiden Debian von anderen Linux-Distributionen:

Freiheit: Wie im Debian-Gesellschaftsvertrag (http://www.debian.org/social_contract) festgelegt, wird Debian zu 100% frei bleiben. Debian verfolgt konsequent das Konzept freier Software. Bei der Entscheidung, ob eine Arbeit als »frei« bezeichnet werden kann, wendet Debian seine Richtlinien für Freie Software (http://www.debian.org/social_contract#guidelines) an.

Das Paketverwaltungssystem von Debian: Das ganze System oder jede einzelne seiner Komponenten kann im laufenden Betrieb aktualisiert werden, ohne neu formatieren zu müssen, ohne angepasste Konfigurationsdateien zu verlieren und (in den meisten Fällen) ohne Neustart des Systems. Die meisten heutzutage erhältlichen Linux-Distributionen bringen ein Paketverwaltungssystem mit; das Paketverwaltungssystem von Debian ist einzigartig und besonders robust (siehe 'Grundlagen des Debian-Paketverwaltungssystems' auf Seite 27).

Offene Entwicklung: Im Gegensatz zu anderen Linux-Distributionen, welche von Einzelpersonen, kleinen, geschlossenen Gruppen oder kommerziellen Herstellern entwickelt werden, ist Debian eine bedeutende, gemeinschaftlich entwickelte Linux-Distribution. Die daran Beteiligten realisieren ein freies Betriebssystem im gleichen Geiste wie Linux und andere freie Software.

Mehr als 1033 freiwillige Paketbetreuer arbeiten an über 42551 Paketen und verbessern Debian GNU/Linux. Die Debian-Entwickler tragen zum Projekt meistens nicht bei, indem sie neue Programme schreiben, sondern durch das Paketieren bestehender Software entsprechend den Projektstandards, indem sie Fehler an die Programmautoren melden und indem sie die Anwender unterstützen. Unter 'Wie kann ich Debian-Mitglied/Debian-Entwickler werden?' auf Seite 57 finden Sie auch Informationen dazu, wie Sie selbst zu Debian beitragen können.

Das universelle Betriebssystem: Debian enthält mehr als 42551 Pakete (<http://packages.debian.org/stable/>) und läuft auf 10 Architekturen (<http://www.debian.org/ports/>). Das ist bedeutend mehr als bei allen anderen GNU/Linux-Distributionen. 'Welche Art von Anwendungen und Entwicklungs-Software ist für Debian GNU/Linux verfügbar?' auf Seite 17 bietet einen Überblick über die verfügbare Software und 'Auf welchen Hardware-Architekturen/Systemen läuft Debian GNU/Linux?' auf Seite 13 eine Beschreibung der unterstützten Hardware-Plattformen.

Die Fehlerdatenbank: Weil rund um die Welt an Debian gearbeitet wird, bedurfte es ausgeklügelter Werkzeuge und Mittel zur schnellen Kommunikation von Fehlern und deren Behebung, um die Entwicklung des Systems zu beschleunigen. Die Anwender sind dazu aufgefordert, Fehlerberichte auf eine formale Art und Weise zu übermitteln, so dass sie schnell über WWW-Archive oder E-Mail verfügbar sind. Sie finden in dieser FAQ unter 'Gibt es Protokolle von bekannten Fehlern?' auf Seite 55 auch weitere Informationen zur Handhabung gemeldeter Fehler.

Die Debian-Richtlinien: Debian hat eine ausführliche Spezifikation seiner Qualitätsstandards, die Debian-Richtlinien. Dieses Dokument definiert die Qualitäten und Standards, die wir für Debian-Pakete fordern.

Weitere Informationen dazu entnehmen Sie bitte unserer Website über die Gründe für die Wahl von Debian (http://www.debian.org/intro/why_debian).

1.6 Wie fügt sich das Debian-Projekt in das GNU-Projekt der Free Software Foundation ein und lässt es sich damit vergleichen?

Das Debian-System basiert auf den Idealen freier Software, die zuerst von der Free Software Foundation (<http://www.gnu.org/>) und vor allem von Richard Stallman (<http://www.stallman.org/>) vertreten wurden. Die mächtigen System-Entwicklungswerkzeuge der FSF und die Werkzeuge und Anwendungen sind ebenfalls ein fester Bestandteil des Debian-Systems.

Das Debian-Projekt ist unabhängig von der FSF, aber wir kommunizieren regelmäßig und arbeiten bei verschiedenen Projekten zusammen. Die FSF hat ausdrücklich verlangt, dass wir unser System »Debian GNU/Linux« nennen, und wir kommen dieser Aufforderung gerne nach.

Das langfristige Ziel der FSF ist es, ein neues Betriebssystem namens GNU zu entwickeln, das auf Hurd (<http://www.gnu.org/software/hurd/>) basiert. Debian arbeitet mit der FSF zusammen an diesem System, Debian GNU/Hurd (<http://www.debian.org/ports/hurd/>) genannt.

1.7 Wie spricht man Debian aus und was bedeutet das?

Der Projektname wird im Englischen Deb'-ii-en ausgesprochen, mit einem kurzen »e« in Deb und mit Betonung auf der ersten Silbe. Dieser Name ist eine Kombination aus Debra und Ian Murdock, der das Projekt gegründet hat. (Wörterbücher scheinen verschiedene Möglichkeiten der Aussprache von Ian (!) anzubieten, aber Ian bevorzugt ii'-en.)

Kapitel 2

Debian GNU/Linux bekommen und installieren

Das offizielle Dokument, welches die Installation beschreibt, ist die Debian GNU/Linux Installationsanleitung (<http://www.debian.org/releases/stable/installmanual>). Im Folgenden geben wir einige ergänzende Hinweise, wie Sie Debian GNU/Linux bekommen und installieren können.

2.1 Was ist die aktuelle Version von Debian?

Zurzeit gibt es drei Debian GNU/Linux-Versionen:

Release 8, die derzeitige »Stable«-Distribution, Codename jessie Hierbei handelt es sich um stabile und ausführlich getestete Software, die nur geändert wird, wenn bedeutende Verbesserungen hinsichtlich Sicherheit und Benutzung zu integrieren sind.

die »Testing«-Distribution, Codename stretch Hier werden die Pakete abgelegt, die in der nächsten Stable-Distribution veröffentlicht werden sollen; sie sind schon etwas in Unstable getestet worden, aber es kann sein, dass sie für die Veröffentlichung noch nicht ganz bereit sind. Diese Distribution wird öfter aktualisiert als Stable, jedoch nicht so oft wie Unstable.

die »Unstable«-Distribution, Codename Sid Dies ist die in Entwicklung befindliche Version, sie wird laufend aktualisiert. Sie können sich die Pakete des Unstable-Archivs von jedem Debian-FTP-Server herunterladen und sie jederzeit benutzen, um Ihr System zu aktualisieren, jedoch sollten Sie nicht erwarten, dass Ihr System genauso benutzbar und stabil bleibt wie vorher – darum wird sie »Unstable« (instabil) genannt!

Weitere Informationen dazu entnehmen Sie bitte 'Wie viele Debian-Distributionen gibt es?' auf Seite 21.

2.2 Gibt es Paketaktualisierungen in »Stable«?

Zur stabilen Veröffentlichung werden keine neuen Funktionalitäten hinzugefügt. Sobald eine Debian-Version herausgegeben und als stabil markiert wurde, gibt es dafür nur noch Sicherheitsaktualisierungen. Das heißt, dass nur Pakete aktualisiert werden, für die nach der Veröffentlichung eine Sicherheitslücke gefunden wurde. Alle Sicherheitsaktualisierungen stehen unter security.debian.org (<ftp://security.debian.org>) bereit.

Sicherheitsaktualisierungen haben nur einen Zweck: die Korrekturen von Sicherheitslücken. Sie dienen nicht dazu, ohne das normale Durchlaufen des Prozesses für Zwischenveröffentlichungen (Point releases) zusätzliche Änderungen in die stabile Veröffentlichung einzuschleusen. Folglich werden Korrekturen für Sicherheitsproblemen nicht die Software aktualisieren (im Sinne zusätzlicher oder neuer Funktionalitäten). Das Debian-Sicherheitsteam wird stattdessen die notwendigen Korrekturen auf die Version der Software in »Stable« zurückportieren.

Für weitere Informationen zur Sicherheitsunterstützung lesen Sie bitte die Sicherheits-FAQ (<http://www.debian.org/security/faq>) oder die Anleitung zum Absichern von Debian (<http://www.debian.org/doc/manuals/securing-debian-howto/>).

2.3 Woher/wie kann ich mir die Debian-Installations-Images beschaffen?

Sie können sich die Dateien für die Installations-Images von der Debian-Website (<https://www.debian.org/distrib/>) herunterladen.

Weitere Informationen über CD- oder DVD-Images finden Sie unter Debian GNU/Linux auf CDs (<http://www.debian.org/CD>).

2.4 Wie kann ich Debian von CD-ROM installieren?

Debian von CD zu installieren ist ziemlich einfach: konfigurieren Sie Ihr System so, dass es von CD bootet, legen Sie die CD ein und starten Sie den Rechner neu. Ihr System wird jetzt den Debian Installer starten. Näheres finden Sie in der Debian GNU/Linux Installationsanleitung (<http://www.debian.org/releases/stable/installmanual>).

2.5 Warum enthält eine offiziell als Stable veröffentlichte CD-ROM symbolische Links für »Frozen« und »Unstable«? Ich dachte, diese CD enthält »Stable«!

Offizielle Debian-CD-Images enthalten in der Tat symbolische Links wie

```
/dists/frozen -> jessie/  
/dists/stable -> jessie/  
/dists/testing -> jessie/  
/dists/unstable -> jessie/
```

damit sie funktionieren, wenn Ihre sources.list einen Eintrag wie

```
deb cdrom:[<Name laut CD-Bezeichnung>] / unstable main [...]
```

enthält.

Die Tatsache, dass diese symbolischen Links vorhanden sind, bedeutet *nicht*, dass das Image »Unstable« oder »Testing« oder etwas anderes enthält. Lesen Sie die CD-Bezeichnung in `.disk/info`, um herauszufinden, welche Debian-Version es enthält. Diese Information ist auch in der Datei `/README.txt` auf der CD zu finden.

Lesen Sie <http://www.debian.org/releases/>, um mehr über die aktuellen »Stable«- und »Testing«-Veröffentlichungen zu erfahren.

2.6 Kann ich Debian direkt über das Internet bekommen und installieren?

Ja. Sie können das Debian-Installationssystem von Dateien starten, die Sie sich von unserem FTP-Server oder dessen Spiegeln heruntergeladen haben.

Sie können sich eine kleine CD-Image-Datei herunterladen, damit eine startfähige CD erstellen, das Basissystem davon installieren und den Rest dann über das Netzwerk herunterladen und installieren. Näheres hierzu finden Sie unter <http://www.debian.org/CD/netinst/>.

2.7 Gibt es alternative Strategien, um den Debian Installer zu booten?

Ja. Neben dem Booten von CD oder DVD können Sie die Installation von Debian GNU/Linux auch von USB-Stick, direkt von Festplatte oder mittels TFTP-Netzwerk-Boot starten. Um eine Vielzahl von Computern zu installieren, sind sogar vollautomatische Installationen möglich. Wohlgemerkt: nicht alle Methoden werden von allen Computerarchitekturen unterstützt. Sobald der Installer gebootet wurde, kann der Rest des Systems über das Netzwerk heruntergeladen oder von einem lokalen Medium installiert werden. Weitere Informationen finden Sie in der Debian GNU/Linux Installationsanleitung (<http://www.debian.org/releases/stable/installmanual>).

Kapitel 3

Eine Debian-Distribution auswählen

Es gibt viele verschiedene Debian-Distributionen. Die richtige auszuwählen, ist eine wichtige Entscheidung. Dieser Abschnitt umfasst einige Informationen, die Benutzern helfen sollen zu entscheiden, welche Distribution am besten zu ihrem System passt. Darüber hinaus gibt er Antworten auf mögliche Fragen, die während des Entscheidungsprozesses auftauchen können. Es geht also nicht so sehr um das »Warum Sie Debian wählen sollten«, sondern um »welche Distribution von Debian«.

Um mehr Informationen über die verfügbaren Distributionen zu erhalten, lesen Sie 'Wie viele Debian-Distributionen gibt es?' auf Seite 21.

3.1 Welche Debian-Distribution (Stable/Testing/Unstable) ist geeignet für mich?

Die Antwort darauf ist ein wenig kompliziert. Sie hängt sehr davon ab, was Sie mit dem System zu tun beabsichtigen. Eine Lösung wäre, einen Bekannten zu fragen, der selber Debian verwendet. Aber das bedeutet natürlich nicht, dass Sie nicht selbst eine unabhängige Entscheidung treffen könnten. Tatsächlich sollten Sie genau dazu in der Lage sein, wenn Sie dieses Kapitel bis zum Ende durchgelesen haben.

- Wenn Ihnen Sicherheit und Stabilität am wichtigsten sind, dann installieren Sie Stable. Punkt. Dieser Weg wird am häufigsten eingeschlagen.
- Wenn Sie erstmals Debian benutzen, und es auf einem Desktop-Rechner installieren wollen, beginnen Sie mit Stable. Ein Teil der enthaltenen Software ist zwar betagt, andererseits werden Sie kaum einen Software-Fehler antreffen. Sobald Sie es sich dann zutrauen, können Sie mit wenig Aufwand zum etwas moderneren Unstable (oder Testing) wechseln.
- Wenn Sie ein bereits mit Linux vertrauter Desktop-Benutzer sind und Macken, Systemabstürze sowie mehr oder weniger umfassende Fehlfunktionen hinnehmen können, dann benutzen Sie Unstable. Es bietet topmoderne Software und Fehler werden üblicherweise rasch behoben.
- Wenn Sie einen Server betreiben, insbesondere einen, an den hohe Anforderungen hinsichtlich der Stabilität gestellt werden oder der über das Internet erreichbar ist, sollten Sie Stable installieren. Das ist bei weitem die beste und sicherste Wahl hierfür.

Die folgenden Fragen behandeln Einzelheiten dieser drei Möglichkeiten. Wenn Sie nach dem Lesen dieser FAQ immer noch keine Entscheidung treffen können, bleiben Sie bei Stable.

3.1.1 Sie haben mir empfohlen, Stable zu installieren. Allerdings wird unter Stable bestimmte Hardware nicht erkannt oder funktioniert nicht richtig. Was soll ich tun?

Finden sie im Internet mithilfe einer Suchmaschine heraus, ob jemand anderes Ihre Hardware zum Laufen gebracht hat. Die meiste Hardware sollte gut unter Stable funktionieren. Wenn Sie aber bestimmte, auf höchste Leistung getrimmte oder brandneue Komponenten verwenden, kann es sein, dass diese unter Stable nicht funktionieren. Sie sollten in diesem Fall vielleicht zu Testing oder Unstable wechseln.

Für Laptops ist <http://www.linux-on-laptops.com/> eine sehr gute Anlaufstelle, um herauszufinden, ob jemand Ihr Gerät unter Linux zum Laufen bekommen hat. Die Website bezieht sich nicht direkt auf Debian, ist aber dennoch eine gute Fundgrube für Informationen. Eine ähnliche Informationsquelle für Desktop-Rechner ist uns leider nicht bekannt.

Eine andere Möglichkeit ist, auf der Mailingliste `debian-user-german` nachzufragen, indem Sie eine E-Mail an `debian-user-german@lists.debian.org` schicken. Sie müssen sich dafür nicht anmelden. Das Archiv ist unter <http://lists.debian.org/debian-user-german/> einzusehen. Dort wird auch beschrieben, wie sie sich anmelden können. Wir möchten nachdrücklich empfehlen, dass Sie Ihre Fragen über die Mailingliste schicken, anstatt sie im irc (<http://www.debian.org/support>) zu stellen. Die Nachrichten auf der Mailingliste werden archiviert. Auf diese Weise kann die Lösung für Ihr Problem auch anderen Benutzern helfen, die die gleichen Schwierigkeiten haben.

3.1.2 Gibt es unterschiedliche Versionen eines Pakets in unterschiedlichen Distributionen?

Ja. Unstable hat die aktuellsten (jüngsten) Versionen. Allerdings sind die Pakete in Unstable nicht umfassend getestet und haben möglicherweise Fehler.

Demgegenüber enthält Stable ältere Paketversionen, die viele Tests durchlaufen haben, so dass Sie kaum Fehler antreffen werden.

Die Pakete in Testing stellen den Mittelweg zwischen diesen beiden Extremen dar.

3.1.3 Die stabile Distribution enthält wirklich eine ganze Menge ziemlich veralteter Pakete, wie an Kde, Gnome, Xorg oder sogar dem Kernel deutlich erkennbar ist. Warum ist das so?

Da könnten Sie recht haben. Das Alter der Pakete in Stable hängt davon ab, wann die letzte Veröffentlichung stattgefunden hat. Angesichts dessen, dass üblicherweise mehr als ein Jahr zwischen den Debian-Veröffentlichungen liegt, kann es sein, dass Sie den Eindruck gewinnen, Stable enthalte alte Pakete. Allerdings wurden diese durch und durch getestet. Man kann guten Gewissens sagen, dass diese Pakete keine schwerwiegenden Fehler, Sicherheitslücken und dergleichen enthalten. Pakete in Stable harmonisieren problemlos mit anderen Stable-Paketen. Diese Eigenschaften sind sehr wichtig für produktive Server, die 24 Stunden am Tag und sieben Tage die Woche funktionieren müssen.

Andererseits können Pakete in Testing und Unstable versteckte Fehler, Sicherheitslücken und ähnliches aufweisen. Schlimmer noch, manche Pakete in Testing und Unstable funktionieren möglicherweise nicht wie beabsichtigt. Für gewöhnlich ziehen Personen, die an einem einzelnen Desktop-Rechner arbeiten, die jüngsten und modernsten Paket-Sets vor. Unstable ist die richtige Wahl für diese Leute.

Wie Sie sehen, handelt es sich bei Stabilität und Aktualität um entgegengesetzte Enden eines Spektrums. Wenn Stabilität erforderlich ist, dann installieren Sie die Stable-Distribution. Wenn Sie mit den neusten Paketen arbeiten möchten, dann installieren Sie Unstable.

3.1.4 Wenn ich zu einer anderen Distribution wechseln wollte, ginge das?

Ja, aber das ist eine Einbahnstraße. Sie können von Stable → Testing → Unstable wechseln, aber in umgekehrter Richtung ist das nicht »möglich«. Ein Wechsel zu Unstable sollte daher wohlüberlegt sein.

Tatsächlich verhält es sich so: Wenn Sie Experte sind und willens, etwas Zeit darauf zu verwenden, vorsichtig sind und wissen, was Sie tun, dann ist es eventuell möglich, von Unstable zu Testing und von Testing dann zu Stable zu gehen. Die Installations-Skripte wurden allerdings nicht für dergleichen entworfen. Daher kann es passieren, dass während des Vorgangs Ihre Konfigurationsdateien verloren gehen und ...

3.1.5 Sollte man Stable, Testing oder Unstable installieren?

Nein. Das ist eine ziemlich subjektive Angelegenheit. Es gibt darauf keine absolut richtige Antwort, da dies davon abhängt, welche Software Sie benötigen, ob Sie gewillt sind, sich im Falle von Problemen mit der Fehlerbehebung zu beschäftigen, sowie von Ihrer Erfahrung bei der Systemadministration. Hier einige Tipps:

- Stable ist »robust wie ein Fels«. Sie profitieren davon, dass Debian Sicherheitsaspekte kontinuierlich im Blick hat und die Distribution eine in sich konsistente Zusammenstellung von Softwarepaketen darstellt. Unter Umständen wird aber sehr neue Hardware nicht unterstützt.
- Testing hat aktuellere Software als Stable, und es gibt seltener gravierende Probleme als in Unstable. Sollten allerdings Mängel auftreten, dauert es oft eine ganze Weile, bis die Dinge wieder im Lot sind. Manchmal kann das Tage dauern, hin und wieder auch Monate. Und Testing hat keine zuverlässige Sicherheitsunterstützung.
- In Unstable ändert sich sehr viel. Die Pakete können somit jederzeit fehlerhaft oder nicht verfügbar sein. Allerdings werden Probleme in vielen Fällen binnen Tagen behoben. Und Unstable enthält immer die neuesten Pakete.

Bei der Entscheidung zwischen Testing und Unstable sollten Sie bedenken, dass Ihnen Testing das eine oder andere Problem ersparen kann. Einer der Autoren dieser Dokumentation erlebte so etwas während dem Wechsel von gcc3 auf gcc4: Er versuchte, das Paket `labplot` auf einem mit Unstable betriebenen Rechner zu installieren. Das gelang nicht, da der gcc-Übergang bei Unstable nicht komplett vollzogen war. In Testing hingegen waren die neuen Versionen noch nicht angelangt; es war in sich konsistent und erlaubte die Installation des von gcc abhängigen Paketes.

3.1.6 Hier ist die Rede davon, dass Testing »kaputt« gehen kann. Was soll das heißen?

Es kann vorkommen, dass sich ein Paket nicht mit den Paketmanagement-Werkzeugen installieren lässt. Manchmal sind Pakete nicht mehr verfügbar, da sie wegen Fehlern oder verletzten Abhängigkeiten (vorübergehend) entfernt wurden. Oder ein Paket lässt sich zwar installieren, verhält sich aber nicht ordnungsgemäß.

Wenn solche Dinge passieren, sagt man von einer Distribution, dass sie (zumindest in Bezug auf dieses eine Paket) »kaputt« ist.

3.1.7 Wie kann es sein, dass Testing monatelang beschädigt ist? Würden die in Unstable eingepflegten Fehlerkorrekturen nicht direkt in Testing einfließen?

Die Fehlerkorrekturen und Verbesserungen aus der Unstable-Distribution gelangen nur allmählich nach Testing. Zuvor müssen alle veröffentlichungskritischen Fehler behoben werden und innerhalb einer gewissen Sperrfrist (üblicherweise fünf Tage) dürfen keine neuen mehr gemeldet worden sein. Wenn also ein veröffentlichungskritischer Fehler für das Paket vorliegt, wird es auch nach Ablauf der Frist nicht nach Testing gelangen.

Die Idee ist folgende: wenn das Paket irgendwelche Fehler hat, dann würden diese von Benutzern entdeckt werden, die Unstable verwenden und würden bereinigt, bevor das Paket nach Testing kommt. So bleibt Testing für die meiste Zeit in einem benutzbaren Zustand. Insgesamt ein brillantes Konzept. Allerdings sind die Dinge nicht immer so einfach. Bedenken Sie folgende Situation:

- Stellen Sie sich vor, Sie wären am Paket XYZ interessiert.
- Nehmen wir an, dass am 10. Juni die Version XYZ-3.6 in Testing ist und Version XYZ-3.7 in Unstable.
- Nach fünf Tagen wandert XYZ-3.7 von Unstable nach Testing.
- Also haben am 15. Juni sowohl Testing als auch Unstable XYZ-3.7 in ihren Repositories.
- Sagen wir, ein Benutzer der Testing-Distribution bemerkt, dass ein neues XYZ-Paket verfügbar ist und macht ein Update von XYZ-3.6 auf XYZ-3.7.
- Nun entdeckt am 25. Juni jemand, der Testing oder Unstable benutzt, einen veröffentlichungskritischen Fehler in XYZ-3.7 und meldet diesen an das BTS.
- Der Betreuer von XYZ behebt den Fehler und lädt ein neues Paket nach Unstable hoch, sagen wir am 30. Juni. Wir nehmen hierbei an, dass der Betreuer fünf Tage braucht, um den Fehler zu beheben und die neue Version hochzuladen. Die Zahl von fünf Tagen sollte hier nicht falsch verstanden werden. Es kann länger dauern oder auch nicht, in Abhängigkeit von der Schwere des vorliegenden veröffentlichungskritischen Fehlers.
- Die neue Version ist nun in Unstable. Laut Plan soll XYZ-3.8 daraufhin Testing am 5. Juli erreichen.
- Aber schon am 3. Juli entdeckt wieder jemand einen veröffentlichungskritischen Fehler in XYZ-3.8.
- Nehmen wir an, der Betreuer löst das neue Problem und lädt die neue Version von XYZ nach fünf Tagen hoch.
- Also ist am 8. Juli XYZ-3.7 in Testing und XYZ-3.9 in Unstable.
- Die neue Version XYZ-3.9 soll nach dem neuen Zeitplan nun am 13. Juli in Testing ankommen.
- Nachdem Sie ja Testing verwenden und XYZ-3.7 fehlerhaft ist, könnten sie wahrscheinlich XYZ erst wieder nach dem 13. Juli benutzen. Im Endeffekt hätten Sie dann rund einen Monat lang ein beschädigtes Paket XYZ gehabt.

Die Angelegenheit kann noch weitaus komplizierter werden, wenn etwa XYZ von vier anderen Paketen abhängt. Das kann dann zu einer monatelang nicht benutzbaren Testing-Distribution führen. Das soeben vorgestellte Szenario ist zwar konstruiert, kann aber im echten Leben tatsächlich so vorkommen. Allerdings geschieht das nur selten.

3.1.8 Vom Standpunkt eines Systemverwalters her gesehen: Welche Distribution verlangt mehr Aufmerksamkeit?

Einer der Gründe, warum viele Leute Debian anderen Linux-Distributionen vorziehen, ist der, dass es nur sehr wenig Verwaltungsaufwand erfordert. Diese Leute wollen ein System, das einfach nur funktioniert. Im allgemeinen kann man sagen, dass Stable nur sehr wenig Wartung bedarf, während Testing und Unstable nach fortwährender Pflege durch den Systemverwalter verlangen. Wenn Sie Stable verwenden, müssen Sie lediglich darauf achten, mit den Sicherheits-Updates Schritt zu halten. Wenn Sie Testing oder Unstable verwenden, ist es klug, ein Auge auf neu entdeckte Fehler, Fehlerkorrekturen und neue Fähigkeiten in den bereits installierten Paketen zu haben.

3.1.9 Was passiert, wenn eine neue Version von Debian veröffentlicht wird?

Diese Frage wird Ihnen nicht helfen, eine Debian-Distribution auszuwählen, aber früher oder später stellt sich sich ihnen doch.

Derzeit ist jessie die stabile Distribution. Die nächste stabile Veröffentlichung wird stretch heißen. Lassen Sie uns nun den speziellen Fall betrachten, dass stretch als neue stabile Distribution freigegeben wird.

- oldstable = wheezy; stable = jessie; testing = stretch; unstable = sid
- Unstable wird immer Sid genannt, unabhängig von den Veröffentlichungen.
- Pakete wandern fortwährend von Sid nach Testing (beispielsweise stretch). Aber Pakete in Stable (derzeit also jessie) bleiben immer die selben, außer im Falle von Sicherheits-Updates.
- Nach einer bestimmten Zeit wird Testing eingefroren, wird aber weiterhin als Testing bezeichnet. Von nun an gelangen keine Pakete von Unstable nach Testing mehr, es sei denn, um veröffentlichungskritische Fehler (RC bugs) zu beheben.
- Nachdem Testing eingefroren wurde, müssen alle neu anstehenden Fehlerkorrekturen händisch durch die Mitglieder des Veröffentlichungs-Teams überprüft werden. Das soll sicherstellen, dass sich keine ernsthaften Fehler mehr in die eingefrorene Testing-Veröffentlichung einschleichen können.
- Die veröffentlichungskritischen Fehler im eingefrorenen Testing werden entweder auf Null reduziert oder markiert als »für diese Veröffentlichung ignoriert« bzw. als »für die nächste Zwischenveröffentlichung aufgeschoben«.
- Das eingefrorene Testing wird ohne veröffentlichungskritische Fehler als neue Stable-Veröffentlichung freigegeben. In unserem Beispiel würde diese neue Veröffentlichung stretch genannt werden.
- Nun ist oldstable = jessie, stable = stretch. Zu diesem Zeitpunkt sind der Inhalt von Stable und eingefrorenem Testing identisch.
- Ein neues Testing wird aus dem alten Testing abgeleitet.
- Pakete kommen ab jetzt wieder aus Sid nach Testing und die Debian-Gemeinschaft beginnt, auf die nächste Stable-Veröffentlichung hinzuarbeiten.

3.1.10 Ich habe einen Desktop-Rechner/Rechner-Cluster mit Debian. Wie kann ich herausfinden, welche Distribution darauf läuft?

Das lässt sich meistens leicht klären. Werfen Sie einen Blick in die Datei `/etc/apt/sources.list`. Dort finden Sie einen Eintrag wie diesen:

```
deb http://ftp.de.debian.org/debian/ unstable main contrib
```

Das dritte Feld (hier im Beispiel »unstable«) gibt die Debian-Distribution an, die derzeit auf dem System läuft.

Sie können auch `lsb_release` verwenden (verfügbar in dem Paket `lsb-release`). Wenn Sie dies auf einem System mit Unstable ausführen, erhalten Sie Folgendes:

```
$ lsb_release -a
LSB Version:    core-2.0-noarch:core-3.0-noarch:core-3.1-noarch:core-2.0-ia32:core-3.0-ia32:core-3.1-ia32
Distributor ID: Debian
Description:    Debian GNU/Linux unstable (sid)
Release:        unstable
Codename:       sid
```

Allerdings ist es nicht immer so einfach. Manche Systeme haben eine `sources.list`-Datei mit mehreren Einträgen, die sich auf unterschiedliche Distributionen beziehen. Das kann der Fall sein, wenn der Systemverwalter verschiedene Pakete aus unterschiedlichen Distributionen verwendet. Dies wird häufig als `apt-pinning` bezeichnet. Solche Systeme verwenden wahrscheinlich einen Mix aus unterschiedlichen Distributionen.

3.1.11 Augenblicklich verwende ich Stable. Kann ich zu Testing oder Unstable wechseln? Und wenn, wie?

Wenn Sie derzeit Stable einsetzen, dann wird in der Datei `/etc/apt/sources.list` das dritte Feld entweder »jessie« oder »stable« enthalten. Diesen Eintrag müssen Sie passend zu der Distribution ändern, die Sie verwenden möchten. Wenn das Testing ist, dann ändern Sie das dritte Feld in `/etc/apt/sources.list` in »testing«. Wenn Sie Unstable verwenden wollen, dann tragen Sie in das dritte Feld »unstable« ein.

Gegenwärtig wird Testing stretch genannt. Wenn Sie das dritte Feld von `/etc/apt/sources.list` in »stretch« ändern, dann werden Sie künftig Testing verwenden. Aber auch wenn stretch zu Stable wird, werden Sie weiterhin stretch haben.

Unstable wird immer Sid genannt. Wenn Sie also das dritte Feld von `/etc/apt/sources.list` in »sid« ändern, werden Sie stets Unstable nutzen.

Gegenwärtig bietet Debian Sicherheits-Updates für Testing, aber nicht für Unstable an, da Korrekturen für Unstable direkt im Hauptarchiv vorgenommen werden. Wenn Sie also Unstable benutzen, sollten Sie sicherstellen, dass Sie die Zeilen in `/etc/apt/sources.list` entfernen, die sich auf Sicherheits-Updates beziehen.

Wenn es für die Veröffentlichung, zu der Sie hochrüsten möchten, Veröffentlichungshinweise gibt (möglicherweise auch obwohl die offizielle Veröffentlichung noch bevorsteht), wäre es klug, diese zu lesen. Sie enthalten eventuell Informationen darüber, wie Sie das Upgrade vornehmen sollten.

Dennoch können Sie, nachdem Sie die obigen Veränderungen vorgenommen haben, `aptitudeupdate` laufen lassen und dann die Pakete installieren, die Sie haben möchten. Beachten Sie, dass das Installieren eines Pakets aus einer anderen Distribution möglicherweise automatisch gleich Ihr halbes System einem Upgrade unterzieht. Wenn Sie nur einzelne Pakete installieren, haben Sie zum Schluss ein System, auf denen eine Mischung verschiedener Distributionen läuft.

In bestimmten Situationen ist es daher wahrscheinlich das Beste, ein vollständiges Upgrade auf die neue Distribution durchzuführen, indem Sie `apt full-upgrade`, `aptitude safe-upgrade` oder `aptitude full-upgrade` benutzen. Lesen Sie die Handbuchseiten von `apt` und `aptitude` für weiterführende Informationen.

3.1.12 Im Augenblick verwende ich Testing (stretch). Was wird passieren, wenn eine neue Veröffentlichung freigegeben wird? Werde ich weiterhin Testing haben oder wird auf meinem Rechner die neue Stable-Distribution laufen?

Das hängt von den Einträgen in der Datei `/etc/apt/sources.list` ab. Wenn Sie gegenwärtig Testing verwenden, sollten diese Einträge in etwa so aussehen:

```
deb http://ftp.de.debian.org/debian/ testing main
```

oder

```
deb http://ftp.de.debian.org/debian/ stretch main
```

Wenn der Eintrag im dritten Feld von `/etc/apt/sources.list` »testing« lautet, dann werden Sie auch nach der Veröffentlichung weiter Testing verwenden. Nachdem stretch veröffentlicht wurde, werden Sie also eine neue Debian-Distribution mit einem anderen Codenamen haben. Am Anfang werden die Veränderungen kaum auffallen. Das wird sich aber ändern, sobald neue Pakete von Unstable nach Testing kommen.

Steht jedoch »stretch« im dritten Feld, werden Sie künftig Stable haben (weil stretch dann die neue stabile Veröffentlichung sein wird).

3.1.13 Ich bin noch immer verwirrt. Was sagten Sie, soll ich installieren?

Wenn Sie sich unsicher sind, dann wäre wohl die Stable-Distribution die beste Wahl.

3.2 Aber was ist mit Knoppix, Linux Mint Debian Edition, Ubuntu und all den anderen Distributionen?

All diese Distributionen sind nicht Debian. Sie *basieren auf Debian*. Obwohl es viele Ähnlichkeiten und Gemeinsamkeiten bei ihnen gibt, gibt es aber auch einige entscheidende Unterschiede.

All diese Distributionen haben ihre eigenen Vorzüge und sind auf eine bestimmte Gruppe von Benutzern zugeschnitten. Mehr Informationen hierzu finden Sie unter Auf Debian basierende Software-Distributionen (<http://www.debian.org/misc/children-distros>) auf Debians Website.

3.2.1 Ich weiß jetzt, dass Knoppix/Linux Mint Debian Edition/Ubuntu/ ... auf Debian basieren. Bedeutet das auch, dass ich die »apt«-Paketwerkzeuge verwenden kann, wenn ich eine dieser Distributionen auf meiner Festplatte installiere?

Diese Distributionen basieren auf Debian, sind jedoch nicht Debian. Sie können aber dennoch die apt-Paketwerkzeuge zusammen mit einer `/etc/apt/sources.list`-Datei verwenden, die auf die Repositories dieser Distributionen verweist. Aber dann verwenden Sie nicht Debian, sondern eine andere Distribution. Das ist nicht das gleiche.

Wenn Sie eine bestimmte Distribution nutzen, sollten Sie auch genau diese benutzen, und nicht versuchen, zusätzlich Pakete aus anderen Distributionen einzumischen. Recht häufig zieht es Probleme nach sich, wenn versucht wird, Debian-Pakete anderer Distributionen zu installieren. Die Tatsache, dass alle dasselbe Format und den gleichen Namen (`.deb`) verwenden, macht die Pakete mitnichten automatisch kompatibel zueinander.

Beispielsweise handelt es sich bei Knoppix um eine Linux-Distribution, die entworfen wurde, um als Live-CD gebootet zu werden, wohingegen Debian für die Installation auf der Festplatte gedacht ist. Knoppix ist hervorragend geeignet, wenn Sie ermitteln wollen, ob eine bestimmte Hardware funktioniert oder wenn Sie herausfinden möchten, wie sich ein GNU/Linux-System »anfühlt«. Knoppix ist ein gutes Vorfühersystem, während Debian entworfen wurde, um 24 Stunden am Tag und 7 Tage die Woche zu laufen. Darüber hinaus ist die Anzahl der verfügbaren Pakete und die Zahl der unterstützten Prozessorarchitekturen bei Debian weitaus größer als bei Knoppix.

Wenn Sie Debian wollen, ist es das Beste, es mit den vom Debian-Projekt bereitgestellten Mitteln zu installieren. Auch wenn es möglich ist, Debian über andere Distributionen wie etwa Knoppix zu installieren, erfordert das etwas Sachverstand. Aufgrunddessen, dass Sie diese FAQ lesen, nehmen wir an, dass Debian und Knoppix neu für Sie sind. In diesem Fall ersparen Sie sich besser den späteren Ärger und installieren Sie Debian sauber von Grund auf neu.

3.2.2 Ich habe Knoppix/Linux Mint Debian Edition/Ubuntu/... auf meiner Festplatte installiert. Jetzt habe ich ein Problem. Was soll ich tun?

Wir raten Ihnen, nicht die Debian-Foren zu benutzen (weder die Mailinglisten noch den IRC), da die Leute, die Ihnen helfen wollen, glauben könnten, dass Sie ein Debian-System benutzen. Die »Lösungen«, die Sie so erhalten, passen möglicherweise nicht zu Ihrem System. Es könnte sogar sein, dass dadurch Ihre Probleme nur noch größer werden.

Benutzen Sie statt dessen vorrangig die Foren der entsprechenden Distribution. Wenn Sie dort keine Hilfe finden, oder die Lösungen Ihr Problem nicht beheben, können Sie immer noch in Debian-Foren nachhaken. Behalten Sie jedoch das zuvor gesagte im Hinterkopf.

3.2.3 Ich benutze Knoppix/LMDE/Ubuntu/... und möchte nun Debian verwenden. Wie steige ich um?

Betrachten Sie den Wechsel von einer auf Debian basierenden Distribution zu Debian als Wechsel von einem Betriebssystem zu einem anderen. Sie sollten Sicherungskopien Ihrer Daten anlegen und das Betriebssystem von Grund auf neu installieren. Sie sollten nicht versuchen, ein »Upgrade« auf ein Debian mit Hilfe der Paketwerkzeuge zu machen, da Sie am Ende möglicherweise mit einem unbrauchbaren System dastehen könnten.

Wenn sich alle Ihre Benutzerdaten (beispielsweise `/home`) auf einer gesonderten Partition befinden, ist der Umstieg auf Debian ziemlich einfach. Sie müssen das Installationssystem lediglich anweisen, diese Partition bei der Neuinstallation einzubinden (aber nicht zu formatieren). Sicherungskopien Ihrer Daten ebenso wie der Konfigurationsdateien (etwa `/etc/` und vielleicht auch `/var/`) anzulegen, ist dennoch ratsam.

Kapitel 4

Kompatibilitätsfragen

4.1 Auf welchen Hardware-Architekturen/Systemen läuft Debian GNU/Linux?

Debian GNU/Linux enthält den kompletten Quelltext aller mitgelieferten Programme, daher sollte es auf allen Systemen laufen, die vom Linux-Kernel unterstützt werden; Details finden Sie in der Linux FAQ (<http://en.tldp.org/FAQ/Linux-FAQ/intro.html#DOES-LINUX-RUN-ON-MY-COMPUTER>).

Die aktuelle Debian GNU/Linux-Veröffentlichung 8, enthält eine komplette Binärdistribution für die folgenden Architekturen:

- *amd64*: für alle Systeme basierend auf AMD 64-Bit-CPU's mit AMD64-Erweiterung und Intel-CPU's mit EM64T-Erweiterung und einem normalen 64-Bit-Userspace.
- *arm64*: unterstützt die aktuellsten 64-Bit ARM-Geräte.
- *armel*: für Little-Endian ARM-Maschinen.
- *armhf*: eine Alternative zu *armel*, für ARMv7-Maschinen mit Hardware-Fließkommaeinheit.
- *i386*: für PCs, die auf Intel- oder kompatiblen Prozessoren basieren, einschließlich 386, 486, Pentium, Pentium Pro, Pentium II (sowohl Klamath als auch Celeron) und Pentium III von Intel, und den meisten kompatiblen Prozessoren von AMD, Cyrix und anderen.
- *ia64*: für Intel-IA-64-Computer (»Itanium«).
- *mips*: für Big-Endian-MIPS-Systeme von SGI, Indy und Indigo2; *mipsel*: für Little-Endian-MIPS-Maschinen, Digital DECstations.
- *powerpc*: für einige IBM-/Motorola-PowerPC-Maschinen, einschließlich den Apple Macintosh PowerMac-Modellen sowie den CHRP-, und PReP- Open-Architecture-Maschinen.
- *ppc64el*: 64-bit Little-Endian PowerPC-Portierung, unterstützt verschiedene aktuelle PowerPC-/POWER-Prozessoren.
- *s390x*: 64-Bit-Portierung für IBM System z-Maschinen; hat die s390-Maschinen ersetzt.

Weitere Debian-Binärdistributionen befinden sich derzeit in der Entwicklung: *hurd-i386* (für GNU-Hurd-Kernel auf i386 32-Bit-PCs), *mipsel64* (für 64-Bit MIPS im Little-Endian-Modus), *powerpcspe* (Portierung für die "Signal Processing Engine"-Hardware), *sparc64* (für 64-Bit SPARC-Prozessoren), *sh* (für Hitachi SuperH-Prozessoren) und *x32* (für amd64/x86_64-CPU's mit 32-Bit-Zeigern).

Die Unterstützung für die *m68k*-Architektur wurde in der Etch-Veröffentlichung (Debian 4.0) eingestellt, da es die im Debian-Projekt gestellten Anforderungen zur Veröffentlichung nicht erfüllte. Diese Architektur bezieht sich auf Amigas und ATARIs mit einem Motorola 680x0-Prozessor für $x \geq 2$; mit MMU. Dennoch ist diese Portierung noch immer aktiv und für die Installation verfügbar, auch wenn sie nunmehr nicht mehr Teil der offiziellen Stable-Veröffentlichung ist; sie wird möglicherweise für zukünftige Veröffentlichungen wieder reaktiviert.

Die Unterstützung für *hppa* (Hewlett-Packard's PA-RISC-Maschinen) und *alpha* (Compaq/Digital's Alpha-Systeme) wurde in der Squeeze-Veröffentlichung (Debian 6.0) aus ähnlichen Gründen eingestellt. Die *arm*-Portierung wurde ebenfalls mit dieser Veröffentlichung eingestellt, sie wurde durch die *armel*-Architektur ersetzt.

Die Unterstützung für die 32-Bit *s390*-Portierung (*s390*) wurde beendet und in Jessie (Debian 8) durch *s390x* ersetzt. Außerdem sind die Portierungen IA-64 und Sparc aus der Veröffentlichung entfernt, da sie nicht mehr genügend Unterstützung aus der Entwicklergemeinde bekamen.

Weitere Informationen über verfügbare Portierungen finden Sie auf der Portierungs-Website (<http://www.debian.org/ports/>).

Bezüglich näherer Details zum Booten, Partitionieren der Festplatte, Aktivieren von PCMCIA- (PC-Card) Geräten und ähnlichen Dingen folgen Sie bitte den Anweisungen in der Installationsanleitung, verfügbar über unsere Website unter <http://www.debian.org/releases/stable/installmanual>.

4.2 Welche Kernel beinhalten die Distributionen von Debian GNU/Linux?

Neben Linux bietet Debian eine vollständige Binärdistribution für die folgenden Betriebssystem-Kernel:

- FreeBSD: bereitgestellt durch die *kfreebsd-amd64*- und *kfreebsd-i386*-Portierungen; für 64-Bit-PCs respektive 32-Bit-PCs. Diese Portierungen wurden erstmals in Debian 6.0 Squeeze als *Technologie-Vorschau* veröffentlicht. Allerdings waren sie nicht Teil der Jessie-Veröffentlichung (Debian 8).

Zusätzlich zu diesen wird auch an folgenden Adaptierungen gearbeitet:

- *avr32*, Portierung auf Atmels 32-Bit RISC-Architektur;
- *hurd-i386*, eine Portierung für 32-Bit-PCs. Diese Portierung wird GNU Hurd verwenden, ein neues, durch die GNU-Gruppe zusammengestelltes Betriebssystem.
- *sh*, Portierung auf Hitachi-SuperH-Prozessoren.

Es gab auch Bestrebungen, die Distribution auf den NetBSD-Kernel zu portieren, um somit *netbsd-i386* (für 32-Bit-PCs) und *netbsd-alpha* (für Alpha-Maschinen) anzubieten. Aber diese Portierungen wurde nie veröffentlicht und sind mittlerweile stillgelegt.

Weitere Informationen über verfügbare Portierungen finden Sie auf der Portierungs-Website (<http://www.debian.org/ports/>).

4.3 Wie kompatibel ist Debian mit anderen Linux-Distributionen?

Debian-Entwickler kommunizieren mit Herstellern anderer Linux-Distributionen, um nach Möglichkeit Binärkompatibilität zwischen Linux-Distributionen sicherzustellen. Die meisten kommerziellen Linux-Produkte laufen unter Debian genau so gut wie auf dem System, auf dem sie erstellt wurden.

Debian GNU/Linux folgt dem Linux Filesystem Hierarchy Standard (<http://www.pathname.com/fhs/>). Jedoch gibt es einen gewissen Interpretationsspielraum bei einigen Festlegungen in diesem Standard, was zu kleineren Unterschieden zwischen einem Debian-System und anderen Linux-Systemen führen kann.

Debian GNU/Linux unterstützt für die Linux Standard Base (<http://www.linuxbase.org/>) entwickelte Software. Die LSB ist eine Spezifikation, die es erlaubt, ein Binärpaket unter verschiedenen Distributionen zu verwenden. Die Debian Etch-Veröffentlichung wurde nach den Regeln der LSB-Version 3.1 zertifiziert, siehe die Certification-Webseite der Linux Foundation (<https://www.linux-foundation.org/lsb-cert/productdir.php>). In der *debian-lsb*-Mailingliste (<http://lists.debian.org/debian-lsb/>) erörtert und koordiniert das Debian-Projekt, wie den Anforderungen der Linux Standard Base entsprochen werden kann.

4.4 In welchem Maße ist der Quellcode von Debian mit anderen Unix-Systemen kompatibel?

Für die meisten Anwendungen ist der Linux-Quellcode kompatibel mit anderen Unix-Systemen. Er unterstützt nahezu alles, was in System V Unix-Systemen und den freien und kommerziellen BSD-basierten Systemen verfügbar ist. Jedoch hat solch eine Behauptung im Unix-Umfeld nahezu keine Bedeutung, da es keine Möglichkeit gibt, dies zu überprüfen. Im Bereich der Software-Entwicklung wird vollständige Kompatibilität benötigt statt nur in »nahezu allen« Fällen. Daher

wurden schon vor Jahren Standards als nötig erachtet und heutzutage ist POSIX.1 (IEEE Standard 1003.1-1990) einer der bedeutendsten Standards für Quellcode-Kompatibilität in Unix-artigen Betriebssystemen.

Konformität mit POSIX.1 ist erklärtes Ziel für Linux, aber die POSIX-Standards kosten richtiges Geld und die POSIX.1- (und FIPS 151-2-) Zertifizierung ist sehr teuer; dies erschwerte es Linux-Entwicklern, vollständige POSIX-Konformität zu erreichen. Die Zertifizierungskosten machen es unwahrscheinlich, dass Debian je eine offizielle Konformitätszertifizierung erhält, selbst wenn es die Validierungstests vollständig besteht. (Die Validierungstests sind mittlerweile frei verfügbar, so dass davon ausgegangen werden kann, dass sich mehr Personen mit Angelegenheiten rund um POSIX.1 befassen werden.)

Unifix GmbH (Braunschweig, Deutschland) entwickelte ein Linux-System, das zertifiziert wurde, zu FIPS 151-2 (einer Obermenge von POSIX.1) kompatibel zu sein. Diese Technologie war in Unifix' eigener Distribution namens Unifix Linux 2.0 und in Lasermoon's Linux-FT enthalten.

4.5 Kann ich Debian-Pakete (».deb«-Dateien) auf meinem Red Hat-/Slackware- /... Linux-System verwenden? Kann ich Red Hat-Pakete (».rpm«-Dateien) auf meinem Debian GNU/Linux-System einsetzen?

Die verschiedenen Linux-Distributionen verwenden unterschiedliche Paketformate und Paketverwaltungsprogramme.

Was wahrscheinlich funktioniert: Für einen Linux-Rechner, der mit einer »fremden« Distribution betrieben wird, können Sie sich eines Programmes bedienen, welches Debian-Pakete entpackt. Umgekehrt können Sie auch mit dem Entpacken eines Red-Hat- oder Slackware-Pakets auf einem Debian-basierten Rechner Erfolg haben. Entscheidend dabei ist, wohin die Dateien des Pakets gelangen; mit anderen Worten, inwieweit die Distributionen den Linux Filesystem Hierarchy Standard befolgen. Des weiteren besteht die Möglichkeit, mit Hilfe von alien (<http://packages.debian.org/alien>) Pakete aus einem Paketformat in ein anderes zu konvertieren.

Was wahrscheinlich nicht funktioniert: Die meisten Paketverwaltungsprogramme erstellen administrative Dateien, wenn sie ein Archiv auspacken. Diese administrativen Dateien sind im Allgemeinen nicht standardisiert. Deshalb wird das Auspacken eines Debian-Pakets auf einem »fremden« Rechner unvorhersehbare (und sicher keine sinnvollen) Auswirkungen auf den Paketmanager in diesem System haben. Ebenso könnten Werkzeuge anderer Distributionen erfolgreich ihre Archive auf einem Debian-System auspacken, werden aber wahrscheinlich Fehler im Debian-Paketverwaltungssystem verursachen, wenn die Zeit der Aktualisierung oder Entfernung dieser Pakete kommt, oder wenn einfach nur ausgegeben werden soll, welche Pakete in einem System vorhanden sind.

Ein besserer Weg: Der Linux File System Standard (und damit Debian GNU/Linux) fordert, dass Unterverzeichnisse von `/usr/local/` vollständig unter der Kontrolle des Benutzers stehen. So bietet es sich an, »fremde« Pakete in dieses Verzeichnis zu entpacken und die dort befindlichen Programme selbst aktuell zu halten, zu konfigurieren und gegebenenfalls wieder zu löschen.

4.6 Wie soll ich ein nicht-Debian-Programm installieren?

Dateien im Verzeichnis `/usr/local/` und darunter werden vom Debian-Paketverwaltungssystem ignoriert. Deswegen ist es üblich, den Quellcode eigener Programme in `/usr/local/src/` abzulegen. Sie könnten zum Beispiel die Dateien eines Pakets namens »foo.tar« in das Verzeichnis `/usr/local/src/foo/` extrahieren. Nachdem Sie es kompiliert haben, legen Sie die Binärprogramme in `/usr/local/bin/` ab, die Bibliotheken in `/usr/local/lib/` und die Konfigurationsdateien in `/usr/local/etc/`.

Falls es zwingende Gründe gibt, Programme oder Dateien in einem gewissen Verzeichnis abzulegen, können Sie sie dennoch unter `/usr/local/` speichern und dementsprechend symbolische Links von der benötigten Stelle im Dateisystem zur Position in `/usr/local/` anlegen, zum Beispiel mit

```
ln -s /usr/local/bin/foo /usr/bin/foo
```

Auf jeden Fall sollten Sie, wenn Sie ein Paket erhalten, dessen Copyright die Weitergabe gestattet, in Betracht ziehen, ein Debian-Paket daraus zu erzeugen und es in das Debian-System hochzuladen. Anleitungen, wie man Paketbetreuer wird, sind im Debian-Policy-Handbuch (siehe 'Welche andere Dokumentation gibt es auf einem und für ein Debian-System?' auf Seite 53) verfügbar.

4.7 Warum kann ich keine Programme kompilieren, die libtermcap benötigen?

Debian verwendet die `terminfo`-Datenbank und die `ncurses`-Bibliothek für Terminal-Schnittstellenbefehle anstatt der `termcap`-Datenbank und der `termcap`-Bibliothek.

Die low-level-Bibliothek `terminfo` wurde aus der grundlegenden `ncurses`-Bibliothek ausgegliedert und es wurde ein neues Paket namens `libtinfo5` erstellt. `libncurses` ist auf `libtinfo` verlinkt, so dass Benutzer, die Programme kompilieren, welche Rückmeldungen von der Terminal-Schnittstelle benötigen, Verweise auf `libtermcap` durch solche auf `libncurses` ersetzen sollten.

Um Binärprogramme zu unterstützen, die bereits gegen die `termcap`-Bibliothek gelinkt worden sind und für die Sie den Quellcode nicht haben, müssen Sie den Quellcode der Bibliothek suchen und sie selbst kompilieren. Debian stellte in der Vergangenheit ein Paket namens »`termcap-compat`« bereit, das `libtermcap.so.2` und `/etc/termcap` enthielt. Dies wurde jedoch in 2005 aus der Distribution entfernt.

4.8 Warum kann ich AccelX nicht installieren?

AccelX verwendet die `termcap`-Bibliothek für die Installation. Lesen Sie 'Warum kann ich keine Programme kompilieren, die libtermcap benötigen?' auf dieser Seite.

Kapitel 5

Für Debian-Systeme verfügbare Software

5.1 Welche Art von Anwendungen und Entwicklungs-Software ist für Debian GNU/Linux verfügbar?

Wie die meisten Linux-Distributionen bietet Debian GNU/Linux:

- die Haupt-GNU-Anwendungen für Software-Entwicklung, Dateibearbeitung und Textverarbeitung, darunter gcc, g++, make, texinfo, Emacs, der Bash-Shell und viele aktualisierte Unix-Hilfsprogramme,
- Perl, Python, Tcl/Tk sowie verschiedene zugehörige Programme, Module und Bibliotheken für diese,
- TeX (LaTeX) und Lyx, dvips, Ghostscript,
- das X-Window-System, das eine netzwerkbasierte grafische Benutzerschnittstelle für Linux bereitstellt, und unzählige X-Anwendungen, einschließlich der GNOME-, KDE- und Xfce-Desktop-Umgebungen,
- ein komplettes Sortiment von Netzwerkanwendungen, mit Servern für Internet-Protokolle wie HTTP (WWW), FTP, NNTP (news), SMTP und POP (E-Mail) sowie DNS (Server für die Namensauflösung); relationale Datenbanken wie PostgreSQL, MySQL; außerdem Webbrowser, inklusive der verschiedenen Mozilla-Produkte.
- eine vollständige Palette von Büroanwendungen, wie die Office-Programme von LibreOffice, Gnumeric und andere Tabellenverarbeitungsprogramme, WYSIWYG-Editoren und Kalender.

Mehr als 42304 Pakete, bestehend aus News-Servern und -Leseprogrammen, Sound-Unterstützung, FAX-Programmen, Datenbank- und Tabellen-Programmen, Bildverarbeitungsprogrammen, Kommunikations-, Netzwerk- und E-Mail-Programmen, Webservern sowie Amateurfunkprogrammen und vielem mehr sind in der Distribution enthalten. Weitere 696 Software-Sammlungen sind als Debian-Paket verfügbar, aber wegen Lizenz einschränkungen gehören diese nicht zum offiziellen Teil von Debian.

5.2 Wer schreibt die ganzen Programme?

Für jedes Paket finden Sie die *Autoren* des Programms in der Datei `/usr/share/doc/PAKET/copyright`, wobei PAKET durch den Paketnamen ersetzt werden muss.

Betreuer, die diese Software für das Debian GNU/Linux-System paketiert haben, sind in der control-Datei (siehe 'Was ist eine Debian-control-Datei?' auf Seite 28) aufgelistet, die jedem Paket beiliegt. Das Debian-Änderungsprotokoll (Changelog) in `/usr/share/doc/PAKET/changelog.Debian.gz` erwähnt diejenigen, die ebenfalls an diesem Debian-Paket gearbeitet haben.

5.3 Wie kann ich eine aktuelle Liste der Programme, die für Debian paketiert wurden, bekommen?

Eine vollständige Liste ist auf jedem Debian-Spiegelserver (<http://www.debian.org/distrib/ftplist>) in der Datei `indices/Maintainers` zu finden. Diese Datei enthält die Paketnamen sowie die Namen und E-Mail-Adressen der jeweiligen Betreuer.

Debian's Paket-Website (<http://packages.debian.org/>) fasst die Pakete in rund dreißig Gruppen (Sektionen) zusammen.

5.4 Wie kann ich eine Entwicklungsumgebung installieren, um Pakete zu bauen?

Wenn Sie auf Ihrem Debian-System Pakete bauen (erstellen) möchten, benötigen Sie eine grundlegende Entwicklungsumgebung, inklusive einem C-/C++-Compiler und einigen anderen essentiellen Paketen. Um diese Umgebung zu erstellen, müssen Sie einfach nur das Paket `build-essential` installieren. Dies ist ein Meta- oder Platzhalter-Paket. Es steht für einen Satz von Paketen, die die zum Bau eines Debian-Pakets benötigten Standard-Entwicklungswerkzeuge beinhalten.

Gewisse Programmpakete lassen sich allerdings nur mit zusätzlicher Software bauen, etwa Bibliotheks-Headern oder zusätzlichen Werkzeugen wie `autoconf` oder `gettext`. Debian stellt viele dieser Werkzeuge, die zum Bau von Software-Paketen benötigt werden, als Debian-Pakete bereit.

Herauszufinden, welche Software genau benötigt wird, kann unter Umständen schwierig sein, außer Sie möchten lediglich Debian-Pakete neu bauen. Dies ist nämlich recht einfach, da offizielle Debian-Pakete eine Liste der zusätzlichen Software enthalten müssen (zusätzlich zu den Paketen in `build-essential`), die zum Bau benötigt werden. Diese werden `Build-Dependencies` (Bauabhängigkeiten) genannt. Um alle zum Bau eines bestimmten Quellpakets nötigen Pakete zu installieren und das Quellpaket dann anschließend zu bauen, führen Sie einfach folgendes aus:

```
# apt-get build-dep foo
# apt-get source --build foo
```

Beachten Sie: falls Sie die Linux-Kernel, die von Debian angeboten werden, selbst bauen möchten, sollten Sie auch das `kernel-package`-Paket installieren. Weitere Details finden Sie unter 'Welche Werkzeuge zum Erzeugen eines angepassten Kernels stellt Debian zur Verfügung?' auf Seite 45.

5.5 Was fehlt bei Debian GNU/Linux?

Es existiert eine Auflistung von Paketen, die noch für Debian erstellt werden sollen: die Liste der Arbeit-bedürftigen und voraussichtlichen Pakete (<http://www.debian.org/devel/wnpp/>).

Weitere Informationen über das Hinzufügen fehlender Software finden Sie unter 'Wie kann ich Debian-Mitglied/Debian-Entwickler werden?' auf Seite 57.

5.6 Wieso bekomme ich die Nachricht »ld: cannot find -lfoo«, wenn ich Programme kompiliere? Warum gibt es keine libfoo.so-Dateien in den Debian-Bibliothekspaketen?

Die Debian-Richtlinien verlangen, dass solche symbolischen Links (zu `libfoo.so.x.y.z` oder ähnlichen) in separaten Entwicklungspaketen gepackt werden. Diese Pakete werden üblicherweise »libfoo-dev« oder »libfooX-dev« (wenn das Bibliothekspaket `libfooX` heißt, X steht dabei für eine Zahl) genannt.

5.7 (Wie) Unterstützt Debian Java?

Es sind mehrere *freie* Implementierungen der Java-Technologie als Debian-Pakete verfügbar, die sowohl Java-Development-Kits (Entwicklungsumgebungen) wie auch Runtime-Environments (Laufzeitumgebungen) enthalten. Sie können damit unter Debian Java-Programme schreiben, auf Programmierfehler untersuchen und auch ausführen.

Um ein Java-Applet ausführen zu können, benötigen Sie einen Webbrowser mit der Fähigkeit, diese zu erkennen und auszuführen. Verschiedene in Debian verfügbare Webbrowser, wie Mozilla oder Konqueror, können mit einem Java-Plugin erweitert werden, welches es ermöglicht, Java-Applets auszuführen.

Weitere Informationen finden Sie in der Debian-Java-FAQ (<http://www.debian.org/doc/manuals/debian-java-faq/>).

5.8 Wie kann ich prüfen, ob ich ein Debian-System benutze und welche Version es ist?

Um zu überprüfen, ob Ihr System von Original-Debian-CDs installiert wurde, verwenden Sie den Befehl

```
lsb_release -a
```

Es wird der Name der Distribution (im Feld »Distributor ID«) sowie die Version des Systems (in den Feldern »Release« und »Codename«) angezeigt. Folgendes ist ein Beispiel von einem Debian-System:

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Debian
Description:   Debian GNU/Linux 7.4 (wheezy)
Release:      7.4
Codename:     wheezy
```

Sie können außerdem kontrollieren, ob die Datei `/etc/debian_version` existiert. Sie enthält nur eine einzelne Zeile mit der Versionsnummer der Veröffentlichung, wie sie in dem `base-files`-Paket definiert ist.

Benutzer sollten sich jedoch bewusst machen, dass ein Debian-System aus vielen Teilen besteht, jedes von ihnen kann aktualisiert werden, (meistens) unabhängig voneinander. Jede Debian-Veröffentlichung enthält streng definierte und unveränderbare Inhalte. Aktualisierungen sind separat verfügbar. Um eine kurze Ausgabe des Installationsstatus' eines Pakets `foo` zu bekommen, geben Sie `dpkg --get-selections foo` ein. Eine ausführlichere Beschreibung erhalten Sie mit:

```
dpkg --get-selections foo
```

Die Versionsnummern aller installierten Pakete können Sie sich anzeigen lassen mit:

```
dpkg --get-architecture
```

Beachten Sie, dass die Existenz des `dpkg`-Programms zwar die Installation von Debian-Paketen auf dem jeweiligen System ermöglicht, aber nachdem das Programm auf viele Betriebssysteme und Architekturen portiert wurde, kann aus dem Vorhandensein von `dpkg` nicht sicher gefolgert werden, dass es sich um ein Debian GNU/Linux-System handelt.

5.9 Wie unterstützt Debian andere Sprachen als Englisch?

- Debian GNU/Linux enthält Tastaturbelegungen (Keymaps) für fast zwei Dutzend Tastaturen und stellt im `kbd`-Paket Hilfsprogramme zum Installieren, Anschauen und Modifizieren dieser Tabellen bereit.

Bei der Installation wird der Benutzer gefragt, welche Tastatur er benutzen möchte.

- Nahezu alle in Debian enthaltene Software unterstützt UTF-8 als Zeichensatz. Die alten Zeichensätze wie ISO-8859-1 oder ISO-8859-2 sind veraltet und sollten nicht mehr verwendet werden.
- Momentan werden über die `manpages-LANG`-Pakete Handbuchseiten (`manpages`) in Deutsch, Spanisch, Französisch, Ungarisch, Italienisch, Japanisch, Koreanisch, Niederländisch, Polnisch, Portugiesisch, Russisch, Türkisch und Chinesisch bereitgestellt (wobei LANG dem 2-Zeichen ISO-Code des Landes entspricht). Um auf eine Handbuchseite in einer bestimmten Sprache zuzugreifen, muss der Benutzer die Umgebungsvariable »LC_MESSAGES« auf die entsprechende Sprache setzen.

Im Falle der deutschsprachigen Handbuchseiten muss LC_MESSAGES z.B. auf »german« gesetzt werden. Das `man`-Programm wird dann die deutschen Handbuchseiten unter `/usr/share/man/de/` suchen.

5.10 Wo sind ezmlm/djbdns/qmail?

Dan J. Bernstein pflegte ursprünglich all seine Programme (<http://cr.yip.to/software.html>) unter einer einschränkenden Lizenz zu vertreiben, die das Verteilen von modifizierten Binärdateien nicht gestattete. Im November 2007 allerdings sagte Bernstein: "[...] i have decided to put all of my future and [...] past software into the public domain"; damit gab er all seine Software für die Allgemeinheit frei. Die Bedingungen zur Verteilung seiner Software finden Sie in seiner FAQ from distributors (<http://cr.yip.to/distributors.html>).

Als dies geschrieben wurde (03/2016), war `ezmlm-idx` nur in Experimental verfügbar (`mlmmj` ist ähnlich und in Debian Jessie enthalten). `djbdns` ist nur in Sid (Unstable) verfügbar, Details finden Sie in Fehler #516394 (<https://bugs.debian.org/516394>) und Fehler #796118 (<https://bugs.debian.org/796118>); das Paket `dbndns` enthält eine vergleichbare Alternative. Die `publicfile`-Software ist immer noch freie Software, ein `publicfile-installer`-Paket ist im contrib-Bereich des Debian-Archivs zu finden.

Weitere Software von Dan J. Bernstein (`qmail`, `daemontools`, `ucspi-tcp`) ist in Debian enthalten.

5.11 Wo ist der Player für Flash (SWF)?

Debian enthält mit `gnash` und `swfdec` zwei freie SWF-Movie-Player.

5.12 Wo ist Google Earth?

Google Earth ist auf Googles Website für GNU/Linux erhältlich; allerdings es ist nicht nur nicht-freie Software, sondern darf darüber hinaus grundsätzlich nicht von Dritten abgegeben werden. Das Paket `googleearth-package` (aus dem contrib-Bereich) könnte jedoch bei der Nutzung dieser Software hilfreich sein.

5.13 Wo sind die VoIP-Programme?

Es gibt zwei offene Protokolle, die für Voice over IP verwendet werden: SIP und H.323. Beide sind in einer Vielzahl von Programmen in Debian main implementiert. `ekiga` ist einer der beliebtesten Clients.

5.14 Ich habe eine Karte für drahtloses Netzwerk, die aber nicht mit Linux funktioniert. Was soll ich tun?

Kaufen Sie eine, die funktioniert :)

Alternativ können Sie `ndiswrapper` verwenden, um einen Windows-Treiber in Ihrem Linux-System einzusetzen (falls Sie einen haben). Weitere Informationen finden Sie auf der `ndiswrapper`-Seite im Debian-Wiki (<http://wiki.debian.org/NdisWrapper>).

Kapitel 6

Die Debian FTP-Archive

6.1 Wie viele Debian-Distributionen gibt es?

Es gibt drei große Distributionen: »Stable«, »Testing« und »Unstable«. Die »Testing«-Distribution kann zeitweise »Frozen« (eingefroren) sein (siehe 'Wie erhält Testing den »frozen«-Status?' auf Seite 23). Daneben gibt es noch die Distributionen »Oldstable« und »Experimental«.

Experimental wird für Pakete benutzt, die sich noch in der Entwicklung befinden und daher die Stabilität ihres Systems hochgradig gefährden können. Diese Distribution benutzen Entwickler, welche absolut brandneue Software untersuchen möchten. Normale Benutzer sollten keine Pakete aus Experimental verwenden, weil sich diese selbst für die erfahrensten Benutzer als gefährlich oder schädlich erweisen können.

Für Hilfe bei der Auswahl einer geeigneten Debian-Distribution lesen Sie bitte 'Eine Debian-Distribution auswählen' auf Seite 7.

6.2 Was haben all diese Namen wie Etch, Lenny usw. zu bedeuten?

Dabei handelt es sich einfach um Codenamen. Wenn sich eine Debian-Distribution noch in der Entwicklung befindet, besitzt sie keine Versionsnummer, aber einen Codenamen. Der Zweck dieser Codenamen ist es, das Spiegeln von Debian-Distributionen zu vereinfachen (wenn ein echtes Verzeichnis wie `unstable` plötzlich in `stable` umbenannt werden würde, würden eine Menge an Daten sinnloserweise erneut heruntergeladen werden).

Zur Zeit ist `stable` ein symbolischer Link auf `jessie` (also Debian GNU/Linux 8) und `testing` ein symbolischer Link auf `stretch`. Dies bedeutet, dass `jessie` die derzeitige Stable-Distribution und `stretch` die derzeitige Testing-Distribution ist.

`unstable` wiederum ist ein permanenter symbolischer Link auf `sid`, da `Sid` immer die Unstable-Distribution ist (siehe dazu 'Was ist mit »Sid«?' auf der nächsten Seite).

6.2.1 Welche Codenamen wurden in der Vergangenheit verwendet?

Andere, zusätzlich zu `jessie` und `stretch` bereits verwendete Codenamen sind: `Buzz` für Release 1.1, `Rex` für Release 1.2, `Bo` für Releases 1.3.x, `Hamm` für Release 2.0, `Slink` für Release 2.1, `Potato` für Release 2.2, `Woody` für Release 3.0, `Sarge` für Release 3.1, `Etch` für Release 4.0, `Lenny` für Release 5.0, `Squeeze` für Release 6.0 und `Wheezy` für Release 7.

6.2.2 Woher stammen diese Codenamen?

Bis jetzt wurden immer Charaktere des Films »Toy Story« von Pixar zur Namensgebung herangezogen:

- `Buzz` (Buzz Lightyear) war der Raumfahrer,
- `Rex` war der Tyrannosaurus,
- `Bo` (Bo Peep) war das Mädchen, welches die Schafe gehütet hat,

- *Hamm* war das Sparschwein,
- *Slink* (Slinky Dog) war der Spielzeughund,
- *Potato* war, logischerweise, Mr. Potato,
- *Woody* war der Cowboy,
- *Sarge* war der Sergeant der grünen Plastiksoldaten,
- *Etch* war die Spielzeugtafel (Etch-a-Sketch),
- *Lenny* war das Fernglas,
- *Squeeze* hießen die dreiäugige Aliens,
- *Wheezy* war der Gummipinguin mit der roten Fliege,
- *Jessie* war das jodelnde Cowgirl,
- *Stretch* war der Gummioctopus mit den Saugern an seinen acht Armen.
- *Sid* war der böartige Junge von nebenan, der immer die Spielzeuge kaputt machte.

Die Entscheidung (<https://lists.debian.org/debian-devel/1996/06/msg00515.html>), Toy-Story-Namen zu benutzen, wurde von Bruce Perens getroffen (<https://lists.debian.org/debian-user/1997/04/msg00011.html>), der zu der Zeit Debian-Projektleiter war und ebenfalls bei Pixar (der Firma, die die Filme produziert hat) arbeitete.

6.3 Was ist mit »Sid«?

Sid oder *Unstable* ist der Ort, wo die meisten Pakete erstmals hochgeladen werden. Es wird nie direkt veröffentlicht werden, da zu veröffentlichende Pakete erst in *Testing* eingefügt werden, um dann später in *Stable* übernommen zu werden. Sid enthält Pakete für bereits veröffentlichte und unveröffentlichte Architekturen.

Der Name »Sid« kommt ebenfalls aus dem Animationsfilm »Toy Story«: Sid war der Junge von Nebenan, der immer die Spielzeuge zerstörte.

1

6.4 Was enthält das stable-Verzeichnis?

- `stable/main/`: Dieses Verzeichnis enthält die Pakete, welche zur Zeit die neueste Veröffentlichung des Debian GNU/Linux-Systems darstellen.
All diese Pakete entsprechen den Debian-Richtlinien für freie Software (http://www.debian.org/social_contract#guidelines) und sind damit frei benutzbar und verteilbar.
- `stable/non-free/`: Dieses Verzeichnis enthält Pakete, deren Copyright-Bedingungen die Verbreitung auf die eine oder andere Art einschränken.
Einige Pakete z.B. haben Lizenzbedingungen, die eine kommerzielle Verbreitung verbieten. Wiederum andere können weitergegeben werden, sind aber tatsächlich Shareware und keine freie Software. Die Lizenzbedingungen jedes dieser Pakete müssen genau gelesen und wahrscheinlich verhandelt werden, bevor eines der Pakete verteilt werden darf, z.B. auf einer CD-ROM.
- `stable/contrib/`: Dieses Verzeichnis enthält Pakete, die den DFSG entsprechen und *frei verteilbar* sind, aber von Paketen abhängen, die *nicht frei* und deshalb nur in `non-free` zu finden sind.

¹Als es das heutige Sid noch nicht gab, hatte die Organisation der FTP-Sites eine große Schwachstelle: Es galt die Annahme, dass beim Erstellen einer Architektur im derzeitigen Unstable die Veröffentlichung erfolgte, sobald diese Distribution das neue Stable wurde. Allerdings ist dies für viele Architekturen nicht der Fall, mit dem Ergebnis, dass diese Verzeichnisse zum Release-Termin verschoben werden mussten. Dies war extrem unpraktisch, da das Verschieben sehr viel Bandbreite verbrauchte. Die Archiv-Administratoren umgingen dieses Problem jahrelang, indem sie Programme für unveröffentlichte Architekturen in einem speziellem Verzeichnis namens »sid« lagerten. Für diese unveröffentlichten Architekturen wurde, sobald sie dann veröffentlicht wurden, ein Link vom derzeitigen Stable auf Sid gesetzt. Ab diesem Zeitpunkt wurden sie ganz normal innerhalb des Unstable-Zweiges erstellt. Diese Anordnung verwirrte allerdings die Benutzer. Mit der Einführung von Paket-Pools (siehe auch 'Was befindet sich in dem `pool`-Verzeichnis?' auf Seite 24) wurden binäre Pakete an den anerkannten Orten des Pools gespeichert, unabhängig von der Distribution. Dadurch wird verhindert, dass bei der Veröffentlichung einer Distribution eine enorme Bandbreite in Anspruch zu nehmen ist. (Allerdings verlangt die Entwicklung dauerhaft einen gewissen Datenverkehr).

6.5 Was enthält das testing-Verzeichnis?

Pakete landen im »testing«-Verzeichnis, nachdem sie zu einem gewissen Grad in Unstable getestet wurden.

Diese Pakete müssen identisch für alle Architekturen vorliegen, auf denen sie gebaut wurden. Es darf auch keine Abhängigkeit vorliegen, welche sie uninstallierbar machen würde. Des Weiteren müssen sie weniger veröffentlichungskritische Fehler aufweisen als die aktuelle Version in Unstable. Auf diese Art hoffen wir, dass Testing immer nahe daran ist, ein Release-Kandidat zu werden.

Weitere Informationen über den Status von Testing und über die einzelnen Pakete finden Sie unter <http://www.debian.org/devel/testing>.

6.5.1 Wie erhält Testing den »frozen«-Status?

Sobald die Testing-Distribution weit genug fortgeschritten ist, erhält sie durch den Release-Manager den »frozen«-Status. Die Verzögerungszeiten bis zur Aufnahme von Paketen nach Testing werden verlängert, um so wenig wie möglich neue Fehler von Unstable nach Testing zu lassen.

Nach einiger Zeit wird die Testing-Distribution dann wirklich »frozen«, also eingefroren. Dies bedeutet, dass alle neuen Pakete, die nach Testing sollen, zurückgehalten werden, außer sie beheben veröffentlichungskritische Fehler. Die Testing-Distribution kann auch während sogenannter »Testzyklen« in diesem Zustand verweilen, wenn die Veröffentlichung kurz bevor steht.

Wenn Testing »frozen« wird, tendiert auch Unstable dazu, teilweise einzufrieren. Das kommt daher, dass die Entwickler sich dabei zurückhalten, in großem Stil neue Software nach Unstable hochzuladen, und zwar aufgrund der Möglichkeit, dass die eingefrorene Software in Testing noch kleinere Korrekturen benötigt oder veröffentlichungskritische Fehler behoben werden müssen, die verhindern, dass Testing zu Stable wird.

Alle Fehler in der Testing-Distribution, die ein Paket an der Freigabe hindern oder die ganze Veröffentlichung verhindern, werden mitprotokolliert. Um mehr zu erfahren, schauen Sie in die Debian Testing Release-Informationen (<http://www.debian.org/releases/testing/>).

Sobald die Anzahl der Fehler sich einem akzeptablen Wert nähert, deklariert man die eingefrorene Testing-Distribution zur Stable-Distribution und veröffentlicht sie mit einer Versionsnummer.

Der wichtigste Fehlerzähler ist der »Release Critical bug count«, der Zähler für die veröffentlichungskritischen Fehler. Sie können ihn auf der Statusseite für veröffentlichungskritische Fehler (<http://bugs.debian.org/release-critical/>) verfolgen. Gemeinsames Ziel für eine Veröffentlichung ist No RC Bugs (keine veröffentlichungskritischen Fehler) (<http://wiki.debian.org/ReleaseGoals/NoRCBugs>), was bedeutet, dass die Distribution keine Fehlerberichte der Kategorien »critical« (kritisch), »grave« (gravierend) oder »serious« (ernst) haben soll. Eine vollständige Liste von Problemen, die als kritisch angesehen werden, finden Sie im RC-Policy-Dokument (http://release.debian.org/testing/rc_policy.txt).

Mit jedem neuen Release ist die vorhergegangene »Stable«-Distribution überholt und wird in das Archiv verschoben. Weitere Informationen finden Sie im Distributions-Archiv (<http://www.debian.org/distrib/archive>).

6.6 Was enthält das unstable-Verzeichnis?

Das »unstable«-Verzeichnis enthält eine Momentaufnahme des derzeitigen Entwicklungssystems. Benutzer können die Pakete darin ohne weiteres ausprobieren oder verwenden, sollten aber darauf eingestellt sein, dass diese unter Umständen nur bedingt einsatzreif sind. Der Vorteil von Unstable liegt darin, dass alle Pakete immer auf dem aktuellsten Stand der GNU/Linux-Entwicklung sind. Wenn allerdings etwas kaputt geht, sollten Sie wissen, wie sie mit den Bruchstücken umgehen müssen.

Unstable enthält ebenfalls die Unterverzeichnisse main, contrib und non-free. Die Pakete werden darin nach den bei Stable beschriebenen Kriterien abgelegt.

6.7 Was haben all die Verzeichnisse in den Debian-FTP-Archiven zu bedeuten?

Diese Verzeichnisse beinhalten die durch Debian GNU/Linux bereitgestellte Software und sind auf jedem Debian-Spiegel anzutreffen.

Der Verzeichnisname `dists` steht kurz für »Distributionen«. In diesem Verzeichnis sind die aktuellen und frühere Debian-Distributionen hinterlegt.

Das `pool`-Verzeichnis enthält die eigentlichen Pakete, siehe dazu auch 'Was befindet sich in dem `pool`-Verzeichnis?' auf dieser Seite.

Ergänzend gibt es folgende Verzeichnisse:

/tools/: enthält DOS-Werkzeuge zum Erstellen von Boot-Disketten, zum Partitionieren Ihrer Festplatte, zum Packen/Entpacken von Dateien und zum Booten von Linux.

/doc/: enthält die grundlegende Debian-Dokumentation, wie z.B. diese FAQ, die Anleitungen zum Umgang mit der Fehlerdatenbank, usw.

/indices/: enthält verschiedene Auflistungen, darunter eine der Paketbetreuer sowie `override`-Dateien mit gewissen Merkmalen der Pakete.

/project/: enthält hauptsächlich Material für Entwickler und verschiedene Dateien.

6.8 Was haben die ganzen Verzeichnisse in `dists/stable/main` zu bedeuten?

In jedem Hauptverzeichnis² gibt es drei Zusammenstellungen von Unterverzeichnissen, die Dateien mit Auflistungen der Binärpakete enthalten.

Da sind zum einen die *binary-irgendwas*-Verzeichnisse, welche Dateien mit Auflistungen der Binärpakete aller verfügbaren Computerarchitektur enthalten, z.B. `/binary-i386/` für Pakete der Intel x86-Architektur oder `/binary-sparc/` für Pakete, die auf Sun SPARCStations laufen.

Die vollständige Liste, welche Architekturen bei den Debian-Veröffentlichungen berücksichtigt wurden, ist auf der Debian-Webseite (<http://www.debian.org/releases/>) zu finden. Für die derzeit aktuelle Veröffentlichung finden Sie Details unter 'Auf welchen Hardware-Architekturen/Systemen läuft Debian GNU/Linux?' auf Seite 13.

Das Verzeichnis `binary-*` enthält in den »Packages(.gz, .bz2)« benannten Dateien eine Zusammenfassung von Informationen zu jedem einzelnen Paket der Distribution. Die eigentlichen Binärpakete liegen direkt im `pool`-Verzeichnis.

Des Weiteren existiert ein Unterverzeichnis namens »source/«, das Dateien beinhaltet, welche die Quellpakete der Distribution auflisten. Diese Dateien heißen `Sources(.gz, .bz2)`.

Zu guter Letzt existiert ein Satz von Unterverzeichnissen mit Dateien, die vom Installationssystem benötigte Auflistungen der Pakete enthalten. Diese liegen in `debian-installer/binary-architektur`.

6.9 Wo befindet sich der Quellcode?

Für jede in die Debian-Distributionen aufgenommene Software wird auch der Quellcode bereitgestellt. Es ist sogar so, dass die zugehörigen Lizenzbedingungen meistens *verlangen*, dass der Quellcode zusammen mit dem eigentlichen Programm ausgeliefert wird oder zumindest zur Verfügung steht.

Der Quellcode wird über das `pool`-Verzeichnis (siehe 'Was befindet sich in dem `pool`-Verzeichnis?' auf dieser Seite), zusammen mit den architekturenspezifischen Binärverzeichnissen, verteilt. Um den Quellcode zu erhalten, ohne sich um die FTP-Archiv-Verzeichnisstruktur kümmern zu müssen, können Sie z.B. `apt-get source PAKETNAME` verwenden.

Aufgrund von Einschränkungen in den Lizenzen könnte der Quellcode bei Paketen in »contrib« und »non-free« eventuell nicht verfügbar sein (diese Bereiche gehören aber formal gesehen auch nicht zum Debian-System). In einigen Fällen dürfen nur Binärdateien (»binary blobs«) ohne deren Quellcode verteilt werden (wie z.B. bei `firmware-misc-nonfree`); in anderen Fällen verbietet die Lizenz die Verteilung von im Vornherein gebauten Binärdateien, erlaubt jedoch Quellcode-Pakete, die die Benutzer dann selbst übersetzen können (wie z.B. bei dem Paket `broadcom-sta-dkms`).

6.10 Was befindet sich in dem `pool`-Verzeichnis?

Pakete werden in einem großen, letztlich nach den Namen der Quellpakete untergliederten »Pool« gelagert. Der besseren Handhabbarkeit wegen ist das `pool`-Verzeichnis unterteilt in die Abschnitte (»main«, »contrib« und »non-free«) und dann

²`dists/stable/main, dists/stable/contrib, dists/stable/non-free, dists/unstable/main/ usw.`

sortiert nach dem ersten Buchstaben des Quellpaketes. Diese Verzeichnisse enthalten zahlreiche Dateien: die Binärpakete für jede Architektur und die Quellpakete, von denen die Binärpakete erstellt wurden.

Wo ein Paket abgelegt ist, läßt sich herausfinden, indem man `apt-cache showsrc PAKETNAME` ausführt und dann in der »Directory:«-Zeile nachschaut. Beispielsweise liegt das `apache`-Paket in `pool/main/a/apache`.

Da es sehr viele Bibliothekspakete (mit Namen `lib*`) gibt, ist der Pool hier noch feiner unterteilt, beispielsweise sind `libpaper`-Pakete in `pool/main/libp/libpaper/` gespeichert.

3

6.11 Was ist »Incoming«?

Nachdem ein Entwickler ein Paket hochgeladen hat, bleibt es für eine kurze Zeit in dem »incoming«-Verzeichnis, bis es auf seine Echtheit überprüft wurde und somit in das Archiv darf.

Normalerweise sollte niemand etwas von dort installieren. Allerdings gibt es seltene Notfälle. Das `incoming`-Verzeichnis ist unter <http://incoming.debian.org/> verfügbar. Es ist möglich, Pakete per Hand von dort zu holen, die GPG-Signatur und MD5-Prüfsumme in den `.changes-` und `.dsc`-Dateien zu überprüfen und sie dann zu installieren.

6.12 Wie erstelle ich mein eigenes, apt-taugliches Paketdepot?

Wenn man eigene Debian-Pakete gebaut hat und diese mit den Standard-Debian-Paketwerkzeugen installieren möchte, so ist es möglich, ein eigenes apt-taugliches Paketarchiv zu erstellen. Dies ist auch nützlich, wenn man nicht bei Debian GNU/Linux erhältliche Paketes selbst zur Verfügung stellen möchte. Informationen und Anleitungen, wie Sie dies bewerkstelligen, finden Sie im Debian Wiki (<https://wiki.debian.org/HowToSetupADebianRepository>).

³Früher lagen die Pakete in dem zur jeweiligen Distribution gehörenden Unterverzeichnis von `dists`. Dies verursachte verschiedene Probleme. So zogen Änderungen beträchtlichen Datenverkehr nach sich. Der Paket-Pool stellte hierfür die Lösung dar. Die `dists`-Verzeichnisse blieben als Ort für die Listendateien erhalten, die von Programmen wie `apt` genutzt werden.

Kapitel 7

Grundlagen des Debian-Paketverwaltungssystems

Dieses Kapitel behandelt einige tiefer gehende Internas der Debian-Paketverwaltung. Wenn Sie hauptsächlich an der *Verwendung* der entsprechenden Programme interessiert sind, springen Sie zu 'Die Debian-Paketverwaltungswerkzeuge' auf Seite 35 und/oder 'Wie man sein Debian-System auf aktuellem Stand hält' auf Seite 41.

7.1 Was ist ein Debian-Paket?

Pakete beinhalten im Grundsatz alle notwendigen Dateien, um eine Sammlung zusammengehöriger Befehle und Fähigkeiten zu implementieren. Es gibt zwei Arten von Debian-Paketen:

- *Binärpakete*, die ausführbare Dateien, Konfigurationsdateien, man/info-Seiten, Copyright-Informationen und andere Dokumentation beinhalten. Diese Pakete werden in einem speziellen Debian-Archivformat verteilt (siehe 'Wie ist ein binäres Debian-Paket aufgebaut?' auf der nächsten Seite). Sie sind für gewöhnlich an der Dateierweiterung `».deb«` zu erkennen. Binärpakete können mittels des Debian-Werkzeugs `dpkg` entpackt werden (eventuell auch über Frontends wie `aptitude`). Mehr Details finden Sie auf dessen Handbuchseite.
- *Quellpakete*, welche aus einer `.dsc`-Datei bestehen, die das Quellpaket beschreibt (inklusive der Namen der folgenden Dateien), einer `.orig.tar.gz`-Datei, welche den Original-Quellcode ohne Veränderungen in einem gzip-komprimierten tar-Format enthält, sowie üblicherweise einer `.diff.gz`-Datei mit Debian-spezifischen Änderungen des Codes. Das Dienstprogramm `dpkg-source` packt und entpackt Debian-Quellpakete. Mehr Details sind auf der Handbuchseite zu finden. (Das Programm `apt-get` kann als Frontend für `dpkg-source` genutzt werden.)

Für die Installation von Software benutzt das System die von den Paketbetreuern sorgfältig definierten Abhängigkeiten. Diese Abhängigkeiten sind in der `control`-Datei, die zu jedem Paket gehört, dokumentiert. Zum Beispiel beinhaltet das Paket des GNU C-Compilers (`gcc`) Abhängigkeiten (»depends«) zu dem Paket `binutils`, welches den Linker und den Assembler enthält. Wenn ein Benutzer versucht, `gcc` zu installieren, ohne zuerst `binutils` installiert zu haben, gibt das Paketverwaltungssystem (`dpkg`) die Fehlermeldung aus, dass es das Paket `binutils` benötigt und stoppt die Installation von `gcc`. (Allerdings kann, wer dies nicht hinnehmen möchte, die Prüfung außer Kraft setzen, siehe `dpkg(8)`.) Näheres in 'Was ist damit gemeint, dass ein Paket eine *Depends*-, *Recommends*-, *Suggests*-, *Conflicts*-, *Replaces*- oder *Provides*-Abhängigkeit zu einem anderen Paket hat?' auf Seite 31 weiter unten.

Debian-Paketwerkzeuge können benutzt werden, um:

- Pakete oder Paketeile zu verändern und zu verwalten;
- lokal eine vom Nutzer bevorzugte Version einer Datei in ein Paket einzufügen;
- Entwickler beim Aufbau von Paketarchiven zu unterstützen, und
- Benutzern die Installation von Paketen zu ermöglichen, die sich z.B. auf fernen FTP-Servern befinden.

7.2 Wie ist ein binäres Debian-Pakets aufgebaut?

Ein Debian-Paket oder eine Debian-Archivdatei beinhaltet ausführbare Dateien, Bibliotheken und Dokumentationen, die zu einem Programm oder einer Menge verwandter Programme gehören. Normalerweise hat eine Debian-Archivdatei die Dateierdung `.deb`.

Die Interna des Debian-Paketformats für ausführbare Programme sind auf der Handbuchseite `deb(5)` beschrieben. Dieses interne Format kann sich (von einer Debian GNU/Linux-Veröffentlichung zur anderen) verändern, benutzen Sie daher bitte immer `dpkg-deb(1)`, um `.deb`-Dateien zu bearbeiten.

7.3 Warum sind Debian-Paketdateinamen so lang?

Die Namen aller Debian-Binärpaketdateien sind folgendermaßen aufgebaut: `<foo>_<Versionsnummer>-<Debian-Revisionsnummer>_<Debian-Architektur>.deb`

Bitte beachten Sie, dass `foo` nur nach landläufiger Auffassung den Paketnamen darstellt. Sie können den Paketnamen der Debian-Archivdatei (`.deb`-Datei) auf eine der folgenden Arten herausfinden:

- untersuchen Sie die `packages`-Datei in dem Verzeichnis, in dem es im Debian-FTP-Archiv abgelegt ist. Diese Datei enthält einen Eintrag für jedes Paket. Der erste Abschnitt in jedem Eintrag enthält den formalen Paketnamen.
- benutzen Sie den Befehl `dpkg --info foo_VVV-RRR_AAA.deb` (wobei `VVV` durch die Version, `RRR` durch die Revisionsnummer und `AAA` durch die Architektur des Paketes ersetzt werden muss). Dies gibt unter anderem den zur entpackten Archivdatei gehörenden Paketnamen aus.

Die `VVV`-Komponente ist die Versionsnummer, die vom Original-Entwickler festgelegt worden ist. Hierfür sind keine Standards festgelegt. Für sie sind daher völlig unterschiedliche Formate, von »19990513« bis »1.3.8pre1« in Gebrauch.

Die `RRR`-Komponente ist die Debian-Revisionsnummer, die von einem Debian-Entwickler (oder einem individuellen Benutzer, der sich entschieden hat, das Paket selber zu bauen) festgelegt wurde. Diese Nummer entspricht dem Stand des Debian-Paketes. Eine neue Revisionsnummer kennzeichnet daher Änderungen im Debian-Makefile (`debian/rules`), der Debian-control-Datei (`debian/control`), den Installations- oder Entfernungs-Skripten (`debian/p*`) oder in den Konfigurationsdateien, die mit diesem Paket benutzt werden.

Die `AAA`-Komponente identifiziert den Prozessor, für den das Paket gebaut wurde. Dies ist häufig `amd64`, was für AMD64-, Intel 64 oder Via Nano-Chips steht. Andere mögliche Werte finden Sie, wenn Sie Debians FTP-Verzeichnisstruktur unter 'Was haben all die Verzeichnisse in den Debian-FTP-Archiven zu bedeuten?' auf Seite 23 durchsuchen. Details finden Sie unter »Debian architecture« auf der Handbuchseite `dpkg-architecture(1)`.

7.4 Was ist eine Debian-control-Datei?

Die Spezifikationen zu den Debian-control-Dateien finden Sie im Debian-Policy-Handbuch, Abschnitt 5, siehe 'Welche andere Dokumentation gibt es auf einem und für ein Debian-System?' auf Seite 53.

Hier beispielsweise eine control-Datei des Debian-Pakets »hello«:

```
Package: hello
Version: 2.9-2+deb8u1
Architecture: amd64
Maintainer: Santiago Vila <sanvila@debian.org>
Installed-Size: 145
Depends: libc6 (>= 2.14)
Conflicts: hello-traditional
Breaks: hello-debhelper (<< 2.9)
Replaces: hello-debhelper (<< 2.9), hello-traditional
Section: devel
Priority: optional
Homepage: http://www.gnu.org/software/hello/
Description: example package based on GNU hello
 The GNU hello program produces a familiar, friendly greeting. It
 allows non-programmers to use a classic computer science tool which
 would otherwise be unavailable to them.
.
 Seriously, though: this is an example of how to do a Debian package.
 It is the Debian version of the GNU Project's 'hello world' program
 (which is itself an example for the GNU Project).
```

Das »Package«-Feld zeigt den Paketnamen an. Diesen Namen erwarten die Paketverwaltungswerkzeuge als Eingabe. Er stimmt nicht unbedingt mit dem ersten Teil des Namens der Debian-Archivdatei überein, ähnelt ihm aber gewöhnlich.

Das »Version«-Feld gibt sowohl die Original-Entwickler-Versionsnummer (an erster Stelle) wie auch (im zweiten Teil) die Revisionsnummer des Debian-Paketes an. Dieses wird unter 'Warum sind Debian-Paketdateinamen so lang?' auf der vorherigen Seite näher beschrieben.

Das »Architecture«-Feld gibt den Prozessor-Typ an, für den das Binärpaket kompiliert worden ist.

Das »Depends«-Feld enthält eine Liste von Paketen, die benötigt werden, um dieses Paket erfolgreich installieren zu können.

»Installed-Size« gibt an, wieviel Speicherplatz das installierte Paket auf der Festplatte belegen wird. Dieser Wert wird von den Frontends benutzt, um zu prüfen, ob noch genug Festplattenplatz für die Installation vorhanden ist.

Die »Section«-Zeile gibt an, in welchem Bereich des Debian-FTP-Archives das Paket zu finden ist.

Die »Priority« zeigt, wie wichtig dieses Paket für die Installation ist. Quasi-intelligente Programme wie "apt" oder "aptitude" übernehmen diese Angabe und bilden entsprechende Paketgruppen, zum Beispiel eine Gruppe optionaler Software, siehe 'Was ist ein *Essential*-, *Required*-, *Important*-, *Standard*-, *Optional*- oder *Extra*-Paket?' auf der nächsten Seite.

Das »Maintainer«-Feld enthält die E-Mail-Adresse der Person, die momentan für die Paketbetreuung zuständig ist.

Das »Description«-Feld umreißt das Anwendungsgebiet und die Funktionen eines Paketes.

Mehr Informationen über alle möglichen Felder, die ein Paket haben kann, finden Sie im Debian-Policy-Handbuch, Abschnitt 5 »Control files and their fields«, siehe 'Welche andere Dokumentation gibt es auf einem und für ein Debian-System?' auf Seite 53.

7.5 Was ist ein Debian »conffile«?

»Conffiles« ist eine Liste von Konfigurationsdateien (meistens unter `/etc` zu finden). Diese Dateien werden vom Paketverwaltungswerkzeug bei einer Paketaktualisierung nicht überschrieben. Dies stellt sicher, dass eigene Einstellungen, die in diesen Dateien gesetzt wurden, beibehalten werden. Dies ist notwendig, um den Austausch von Paketen auf laufenden Systemen zu ermöglichen.

Um herauszufinden, welche Dateien bei einem Update erhalten bleiben, benutzen Sie:

```
dpkg --status paket
```

und schauen unter »Conffiles:« nach.

7.6 Was sind Debians »preinst«-, »postinst«-, »prerm«- und »postrm«-Skripte?

Diese Dateien sind ausführbare Skripte, die automatisch vor oder nach einer Paketinstallation bzw. -entfernung ausgeführt werden. Genau wie die `control`-Datei sind all diese Dateien Teil des »control«-Abschnitts der Debian-Archivdatei.

Die einzelnen Dateien sind:

preinst Dieses Skript wird ausgeführt, bevor die Debian-Archivdatei (».deb«-Datei), zu der sie gehört, entpackt wird. Viele »preinst«-Skripte stoppen Dienste, die aktualisiert werden, bis deren Installation bzw. Update abgeschlossen ist (nach dem erfolgreichen Ausführen des »postinst«-Skripts).

postinst Bei Software, die für das konkrete System konfiguriert werden muss, ist es typischerweise ein solches Skript, das (nach dem Transfer der Dateien auf die Festplatte) die Konfiguration des Pakets `foo` abschließt. Oft erfragen »postinst«-Skripte Eingaben vom Benutzer und/oder warnen ihn, dass er das Paket bei Bedarf neu konfigurieren muss, wenn er die Standardwerte akzeptiert. Viele »postinst«-Skripte führen nach der Installation/Aktualisierung die für das Starten bzw. Neustarten der Dienste nötigen Befehle aus.

prerm Dieses Skript stoppt üblicherweise alle Dienste, die mit einem Paket verknüpft sind. Es wird ausgeführt, bevor die Dateien des Paketes gelöscht werden.

postrm Typischerweise modifiziert dieses Skript Links oder andere Dateien, die zu `foo` gehören und/oder entfernt Dateien, die vom Paket erzeugt wurden. (Siehe auch 'Was ist ein virtuelles Paket?' auf der nächsten Seite.)

Momentan können Sie alle control-Dateien in `/var/lib/dpkg/info` finden. Die für das Paket `foo` relevanten Dateien beginnen mit »foo« und haben die Dateierweiterungen »preinst«, »postinst« usw. Die Datei `foo.list` in diesem Verzeichnis enthält eine Liste aller Dateien, die mit dem Paket `foo` installiert worden sind. (Beachten Sie, dass die Pfade der Dateien ein `dpkg`-Interna sind. Sie sich sollten nicht darauf verlassen.)

7.7 Was ist ein *Essential*-, *Required*-, *Important*-, *Standard*-, *Optional*- oder *Extra*-Paket?

Von den Distributionsbetreuern wird jedem Debian-Paket eine *Priorität* zugeordnet, auf die das Paketverwaltungssystem zugreifen kann. Die Prioritäten sind:

- **Required:** Pakete die für das korrekte Funktionieren des Systems benötigt werden.
Dieses schließt alle Werkzeuge mit ein, die notwendig sind, um Systemdefekte zu reparieren. Sie dürfen diese Pakete nicht entfernen, ansonsten kann es passieren, dass Ihr System zusammenbricht und Sie sogar außerstande sind, mittels `dpkg` Sachen wieder zu installieren. Ein nur aus Required-Paketen aufgebautes System ist vermutlich unnütz, aber es bietet genug Funktionalität, um dem Systemadministrator zu ermöglichen, weitere Programme zu installieren.
- **Important**-Pakete sollten auf jedem Unix-ähnlichen System installiert sein.
Andere Pakete, ohne die das System nicht vernünftig laufen kann, finden Sie hier. Das beinhaltet *nicht* Emacs, X11, TeX oder andere große Anwendungen. Diese Pakete stellen lediglich die Basisinfrastruktur dar.
- **Standard**-Pakete sind Standard auf jedem Linux-System, einschließlich eines recht kleinen, aber nicht zu begrenzten Text-Modus-Systems. Es sind Werkzeuge enthalten, um E-Mails zu verschicken (mit `mutt`) oder Dateien von FTP-Servern herunterzuladen.
Programme dieser Priorität werden standardmäßig installiert, wenn der Benutzer nichts anderes ausgewählt hat. Es beinhaltet keine großen Programme, aber den Python-Interpreter und einiges an Server-Software, wie OpenSSH (für ferne Administration) und Exim (für E-Mail-Auslieferung; allerdings kann Exim auch so konfiguriert werden, dass er nur lokale Nachrichten verarbeitet). Auch grundlegende Dokumentation, die für die meisten Benutzer hilfreich sein könnte, ist enthalten.
- **Optional**-Pakete beinhalten alles das, was abseits spezieller Anforderungen zur Verfügung stehen sollte oder wovon unterstellt wird, dass Sie es nutzen wollen, ohne es von vornherein zu kennen.
Dazu gehört X, eine komplette TeX-Distribution und viele andere Programme.
- **Extra:** Pakete, die in Konflikt mit Paketen höherer Priorität stehen, die nur für jemanden interessant sind, der sie schon kennt, oder die spezielle Anforderungen haben, welche sie ungeeignet für »Optional« machen.

Wenn Sie eine Standard-Debian-Installation durchführen, werden alle Pakete mit der Priorität **Standard** oder höher auf Ihrem System installiert. Wenn Sie vordefinierte Programmgruppen (Tasks) auswählen, bekommen Sie auch Pakete mit geringerer Priorität.

Zusätzlich sind einige Pakete als **Essential** markiert. Da diese Pakete für die Grundfunktionalität des Systems absolut notwendig sind, lehnen es die Paketverwaltungswerkzeuge ab, diese zu entfernen.

7.8 Was ist ein virtuelles Paket?

Virtuelle Pakete stellen Verweise auf grundlegende Funktionen des Systems dar und tragen einen entsprechenden, systematischen Namen. Zum Beispiel sind `konqueror` and `firefox-esr` beides Webbrowser, folglich werden beide Programme die Abhängigkeit erfüllen, die ein Programm hat, das einen Webbrowser braucht, um auf einem System richtig zu funktionieren. Beide Pakete erfüllen also die Abhängigkeit des »virtuellen Pakets« namens `www-browser`.

Ebenso bieten `exim4` und `sendmail` beide die Funktionalität eines Mail-Transport-Agents. Wir sagen also, dass beide Programme das »virtuelle Paket« `mail-transport-agent` anbieten. Wenn eines der Programme installiert ist, dann wird die Installation jedes Paketes, das von einem `mail-transport-agent` abhängig ist, durch die Existenz des virtuellen Paketes ermöglicht.

Für den Fall, dass mehr als ein Paket installiert ist, von denen alle dasselbe virtuelle Paket bereitstellen, bietet Debian einen Mechanismus an, der es dem Systemadministrator erlaubt, ein Paket als bevorzugt einzustellen. Der zugehörige Befehl ist `update-alternatives` und wird später in 'Einige Benutzer mögen `mawk`, andere `gawk`; einige mögen `vim`, andere `elvis`; einige `trn`, wieder andere `tin`; wie unterstützt Debian die Vielfalt?' auf Seite 50 näher erläutert.

7.9 Was ist damit gemeint, dass ein Paket eine *Depends*-, *Recommends*-, *Suggests*-, *Conflicts*-, *Replaces*- oder *Provides*-Abhängigkeit zu einem anderen Paket hat?

Das Debian-Paketverwaltungssystem hat eine Reihe von »Paket-Abhängigkeiten«, die (in einer einzigen Markierung) anzeigen, inwieweit ein Programm **A** unabhängig vom Vorhandensein von Programm **B** auf einem gegebenen System arbeiten kann:

- Paket **A** *hängt ab* von Paket **B** (*depends*), wenn **B** unbedingt installiert sein muss, damit **A** läuft. In manchen Fällen hängt **A** überdies von einer bestimmten Version von **B** ab. Meistens handelt es sich dabei um eine Untergrenze, wonach **A**s Abhängigkeit durch jede Version von **B** erfüllt wird, die neuer ist als die angegebene Version.
- Paket **A** *empfiehlt* Paket **B** (*recommends*), wenn der Paketverwalter der Meinung ist, dass die meisten Benutzer **A** nicht ohne die Funktionalität von **B** haben wollen.
- Paket **A** *schlägt* Paket **B** *vor* (*suggests*), wenn **B** Dateien beinhaltet, die auf die Funktionen von **A** bezogen sind und dessen Gebrauchswert für gewöhnlich deutlich steigern.
- Paket **A** *steht in Konflikt* mit Paket **B** (*conflicts*), wenn **A** nicht funktioniert, solange **B** auf dem System installiert ist. Sehr oft sind Konflikte Fälle, in denen **A** Dateien beinhaltet, die eine Verbesserung gegenüber denen aus **B** darstellen. »Conflicts« werden oft mit »Replaces« verbunden.
- Paket **A** *ersetzt* Paket **B** (*replaces*), wenn Dateien, die von **B** installiert wurden, von Dateien aus **A** entfernt und (in manchen Fällen) überschrieben werden.
- Paket **A** *beschädigt* Paket **B** (*breaks*), wenn diese beiden Pakete nicht gleichzeitig auf einem System konfiguriert werden können. Die Paketverwaltung wird es ablehnen, eines davon zu installieren, wenn das andere bereits auf dem System installiert und konfiguriert ist.
- Paket **A** *stellt* Paket **B** *bereit* (*provides*), wenn alle Dateien und Funktionalitäten von **B** in **A** vereinigt sind. Dieser Mechanismus ermöglicht es Benutzern mit einem begrenzten Festplattenplatz, nur den Teil von Paket **A** zu installieren, den sie wirklich benötigen.

Detaillierte Informationen über die Nutzung all dieser Bezeichnungen finden Sie im Policy-Handbuch, Abschnitt 7.2 »Binary Dependencies«, siehe 'Welche andere Dokumentation gibt es auf einem und für ein Debian-System?' auf Seite 53.

7.10 Was bedeutet Pre-Depends (Vor-Abhängigkeit)?

»Pre-Depends« ist eine spezielle Abhängigkeit. Im Fall der meisten Pakete entpackt `dpkg` die Archiv-Datei eines Pakets (also die `.deb`-Datei) unabhängig davon, ob die Dateien, von denen das Paket abhängt, auf dem System existieren oder nicht. Stark vereinfacht bedeutet »entpacken«, das `dpkg` die Dateien aus der Archiv-Datei in Ihrem Dateisystem an der entsprechenden Stelle ablegt. Wenn solch ein Paket von der Existenz anderer Pakete abhängt, lehnt `dpkg` es ab, die Paketinstallation abzuschließen (d.h. die Konfiguration des Pakets wird nicht durchgeführt), bevor die anderen Pakete installiert sind.

Für einige Pakete lehnt `dpkg` jedoch sogar das Entpacken ab, bis bestimmte Abhängigkeiten erfüllt sind. Solche Pakete haben eine sogenannte »Pre-depends«-Abhängigkeit von anderen Paketen. Das Debian-Projekt führte diese Kategorie ein, um ein sicheres Upgrade des Systems vom `a.out`- zum `ELF`-Format zu ermöglichen; dabei war die *Reihenfolge*, in der die Pakete ausgepackt wurden, kritisch. Es gibt andere große Upgrade-Situationen, bei denen diese Methode hilfreich ist, z.B. bei Paketen mit der Priorität »Required« und ihrer Abhängigkeit zu `libc`.

Genau wie zuvor finden Sie weiterführende Informationen dazu im Policy-Handbuch.

7.11 Was bedeutet *unknown*, *install*, *remove purge* und *hold* im Paket-Status?

Diese »Wunsch«-Markierungen zeigen an, was ein Benutzer mit einem Paket tun wollte, z.B. als er `dpkg` aufrief.

Ihre Bedeutungen sind:

- `unknown` (unbekannt) - der Benutzer hat nie angegeben, ob er das Paket wünscht;
- `install` (installieren) - der Benutzer möchte das Paket installiert oder aktualisiert haben;

- `remove` (entfernen) - der Benutzer möchte das Paket entfernt haben, aber die Konfigurationsdateien sollen erhalten bleiben;
- `purge` (vollständig entfernen)- der Benutzer möchte das Paket samt der zugehörigen Konfigurationsdateien entfernt haben;
- `hold` (zurückhalten) - der Benutzer wünscht keine Veränderung an diesem Paket, d.h. das Paket soll in der derzeitigen Version beibehalten werden, wie es ist.

7.12 Wie setze ich ein Paket auf »hold« (zurückhalten)?

Es gibt drei Wege, Pakete zurückzuhalten: mit `dpkg`, `apt` oder `aptitude`.

Mit `dpkg` müssen Sie lediglich die Liste der Paketauswahlen mittels

```
dpkg --get-selections `ls` > selections.txt
```

exportieren. Dann modifizieren Sie die daraus resultierende Datei `selections.txt`: ändern Sie die Zeile, die das Paket beinhaltet, das Sie zurückhalten wollen (in diesem Beispiel `libc6`) von:

```
libc6                                install
```

in:

```
libc6                                hold
```

Speichern Sie die Datei und laden Sie sie mit folgendem Befehl zurück in die »`dpkg`«-Datenbank:

```
dpkg --set-selections < selections.txt
```

Mit `apt` können Sie ein Paket auf Zurückhalten setzen mittels:

```
apt-mark hold Paketname
```

Oder Sie entfernen die Zurückhalten-Markierung mit:

```
apt-mark unhold Paketname
```

Mit `aptitude` können Sie ein Paket auf Zurückhalten setzen mittels:

```
aptitude hold Paketname
```

Oder Sie entfernen die Zurückhalten-Markierung mit:

```
aptitude unhold Paketname
```

7.13 Wie installiere ich ein Quellpaket?

Debian-Quellpakete können nicht im eigentlichen Sinne »installiert« werden. Sie werden lediglich in ein Verzeichnis Ihrer Wahl entpackt, wo Sie dann die Binärpakete daraus erzeugen können.

Quellpakete werden meistens auf denselben Spiegel-Servern angeboten, auf denen auch die Binärpakete zu finden sind. Wenn Sie Ihre `sources.list` (5) für APT so eingerichtet haben, dass die benötigten »`deb-src`«-Zeilen enthalten sind, können Sie jegliches Quellpaket einfach mittels des folgenden Befehls herunterladen:

```
apt-get source foo
```

Um Ihnen beim Bauen der Quelltext-Pakete zu helfen, bieten die Debian Quellpakete den sogenannten »`build-dependencies`«-Mechanismus (Bau-Abhängigkeiten). Das bedeutet, dass die Quellpaket-Betreuer eine Liste anderer Pakete pflegen, die zum Bauen des Paketes benötigt werden. Um zu sehen, wozu dies nützlich ist, probieren Sie einmal dies aus

```
apt-get build-dep foo
```

bevor Sie den Quellcode kompilieren.

7.14 Wie baue ich Binärpakete aus einem Quellpaket?

Der beliebteste Weg hierfür ist die Nutzung verschiedener Wrapper-Werkzeuge. Im folgenden Beispiel wird die Programmsammlung `devscripts` eingesetzt. Installieren Sie dies Paket, falls noch nicht geschehen.

Laden Sie jetzt das Quellpaket herunter:

```
apt-get source foo
```

und wechseln Sie in dessen Verzeichnisbaum:

```
cd foo-*
```

Installieren Sie, falls zum Bauen weitere Pakete gebraucht werden, diese mit:

```
sudo apt-get build-dep foo
```

Beugen Sie Verwirrung vor, wenn später seitens Debian neue Versionen veröffentlicht werden, indem Sie mit folgendem Kommando die selbst erstellte Version kennzeichnen:

```
dch -l local 'Irgendein Text ...'
```

Und dann bauen Sie Ihr Paket:

```
debuild -us -uc
```

Wenn alles korrekt gelaufen ist, sollten Sie jetzt Ihr Paket installieren können mittels

```
sudo dpkg -i ../*.deb
```

Wenn Sie es vorziehen, die Dinge händisch zu erledigen, statt `devscripts` zu benutzen, gehen Sie folgendermaßen vor:

Sie benötigen alle `foo_*.dsc`-, `foo_*.tar.gz`- und `foo_*.diff.gz`-Dateien, um den Quellcode zu kompilieren. (Wobei die innerhalb des Debian-Projekts entstandenen Pakete nicht unbedingt eine `.diff.gz`-Datei beinhalten).

Wenn Sie diese Dateien haben (siehe 'Wie installiere ich ein Quellpaket?' auf der vorherigen Seite) und das Paket `dpkg-dev` auf Ihrem System installiert ist, können Sie mit

```
dpkg-source -x foo_version-revision.dsc
```

das Paket in ein Verzeichnis namens `foo-version` entpacken.

Wollen Sie einfach nur das Paket kompilieren, wechseln Sie in das `foo-version`-Verzeichnis und führen folgenden Befehl aus:

```
dpkg-buildpackage -rfakeroot -b
```

um das Paket zu bauen (beachten Sie, dass dazu das Paket `fakeroot` erforderlich ist). Installieren Sie dann mit

```
dpkg -i ../foo_version-revision_arch.deb
```

das neu gebaute Paket.

7.15 Wie kann ich selbst Debian-Pakete erstellen?

Details hierzu finden Sie im Debian-Leitfaden für neue Paketbetreuer (aus dem `maint-guide`-Paket bzw. unter <http://www.debian.org/doc/devel-manuals#maint-guide>) oder in dem Handbuch für Debian-Paketbetreuer (verfügbar im Paket `debmake-doc` oder unter <https://www.debian.org/doc/devel-manuals#debmake-doc>).

Kapitel 8

Die Debian-Paketverwaltungswerkzeuge

8.1 Welche Programme bietet Debian zur Paketverwaltung an?

Zur Verwaltung der Debian-Pakete gibt es verschiedene Programme, von grafischen oder textbasierten Oberflächen bis hin zu den elementaren Werkzeugen, die benutzt werden, um Pakete zu installieren. Alle verfügbaren Programme bauen auf den elementaren Werkzeugen auf und werden im folgenden - mit den elementaren beginnend und den komplexen schließend - vorgestellt.

Es ist wichtig zu verstehen, dass die komfortablen Paketverwaltungswerkzeuge wie `aptitude` oder `synaptic` auf `apt` aufbauen, welches seinerseits bei der Verwaltung der Pakete auf `dpkg` zurückgreift.

Sehen Sie sich Kapitel 2. Debian-Paketmanagement (<http://www.debian.org/doc/manuals/debian-reference/ch02.en.html>) der Debian-Referenz (<http://www.debian.org/doc/manuals/debian-reference/>) an, falls Sie weitere Informationen über die Paketverwaltungswerkzeuge benötigen. Dieses Dokument ist in verschiedenen Sprachen und Formaten verfügbar, siehe den Eintrag zur Debian-Referenz auf der Übersichtsseite für DDP-Benutzerhandbücher (<http://www.debian.org/doc/user-manuals#quick-reference>).

8.1.1 dpkg

Dies ist das Hauptpaketverwaltungswerkzeug. `dpkg` kann mit vielen verschiedenen Optionen aufgerufen werden. Häufig benutzt werden:

- Alle verfügbaren Optionen anzeigen: `dpkg --help`.
- Die `control`-Datei (und andere Informationen) für ein Paket anzeigen: `dpkg --info foo_VVV-RRR.deb`.
- Ein Paket (inklusive Entpacken und Konfiguration) in das Dateisystem auf der Festplatte installieren: `dpkg --install foo_VVV-RRR.deb`.
- Ein Debian-Archiv (ohne Konfiguration) in das Dateisystem auf der Festplatte entpacken: `dpkg --unpack foo_VVV-RRR.deb`. Bitte beachten Sie, dass das Paket damit *nicht* verwendbar sein dürfte, da Programmdateien oftmals erst nach gewissen Anpassungen ordnungsgemäß zusammenspielen. Dieser Befehl entfernt jede auf der Festplatte installierte Version des Programms und führt das mit dem Paket verbundene »preinst«-Skript (siehe 'Was sind Debians »preinst«, »postinst«, »prerm«- und »postrm«-Skripte?' auf Seite 29) aus.
- Ein bereits ausgepacktes Paket konfigurieren: `dpkg --configure foo`. Neben anderen Sachen ruft diese Option das zugehörige »postinst«-Skript (siehe 'Was sind Debians »preinst«, »postinst«, »prerm«- und »postrm«-Skripte?' auf Seite 29) des Pakets auf. Außerdem aktualisiert es die Dateien, die in der `conffiles`-Datei des Pakets aufgelistet sind. Beachten Sie, dass die »configure«-Option als Argument einen Paketnamen benötigt (z.B. `foo`), *und nicht* den Namen der Debian-Archivdatei (also nicht `foo_VVV-RRR.deb`).
- Eine einzelne Datei namens "blurf" (oder mehrere Dateien namens »blurf*«) aus einem Debian-Archiv extrahieren: `dpkg --fsys-tarfile foo_VVV-RRR.deb | tar -xf - blurf*`.
- Ein Paket entfernen (aber nicht seine Konfigurationsdateien): `dpkg --remove foo`.
- Ein Paket vollständig, inklusive der Konfigurationsdateien entfernen: `dpkg --purge foo`.
- Den Installationszustand aller Pakete anzeigen, auf die der reguläre Ausdruck »foo*« zutrifft: `dpkg --list 'foo*'.`

8.1.2 APT

APT (für *Advanced-Package-Tool*) ist eine leistungsfähige Programmoberfläche zur Debian-Paketverwaltung. Darin enthalten ist das Programm `apt-get`. Es enthält Befehlszeilenprogramme für Suche, Verwaltung und Abfrage von Paketinformationen wie auch für Low-Level-Zugriffe auf tieferliegende Funktionalitäten der `libapt-pkg`-Bibliothek. Lesen Sie dazu das Benutzerhandbuch in `/usr/share/doc/apt-doc/guide.html/index.html`; Sie benötigen dazu das Paket `apt-doc`.

Seit Debian Jessie gibt es für einige häufig verwendete `apt-get`- und `apt-cache`-Befehle ein entsprechendes Äquivalent, das neue Binärprogramm `apt`. Das bedeutet, dass bekannte Befehle wie `apt-get update`, `apt-get install`, `apt-get remove`, `apt-cache search` oder `apt-cache show` jetzt auch einfach über `apt` aufgerufen werden können, also mit Befehlen wie `apt update`, `apt install`, `apt remove`, `apt search` oder `apt show`. Hier ein Überblick über die alten und neuen Befehle:

```
apt-get update          -> apt update
apt-get upgrade        -> apt upgrade
apt-get dist-upgrade   -> apt full-upgrade
apt-get install package -> apt install package
apt-get remove package -> apt remove package
apt-get autoremove     -> apt autoremove
apt-cache search string -> apt search string
apt-cache policy package -> apt list -a package
apt-cache show package  -> apt show package
apt-cache showpkg package -> apt show -a package
```

Das Werkzeug `apt` vereint Funktionalitäten von `apt-get` und `apt-cache` und hat eine nette farbige Ausgabedarstellung, somit ist es für den Benutzer angenehmer zu bedienen. Zur Verwendung in Skripten oder in fortgeschrittenen Anwendungsfällen könnte `apt-get` trotzdem noch empfehlenswert oder nötig sein.

`apt-get` bietet eine einfache Beschaffung und Installation von Paketen aus verschiedenen Quellen mittels Befehlszeile. Im Gegensatz zu `dpkg` erwartet `apt-get` nicht die Debian-Archivdatei, sondern den korrekten Paketnamen als Eingabe und wird ein Paket nur installieren, wenn dies durch eine in `etc/apt/sources.list` aufgeführte Quelle bereitgestellt wird. `apt-get` ruft nach dem Download des `».deb«`-Archivs von den konfigurierten Quellen direkt `dpkg` zur Installation auf.¹

Einige übliche Befehle zur Nutzung von `apt-get` sind:

- Die Liste der Ihrem System bekannten Pakete aktualisieren:

```
apt update
```

(Sie sollten dies regelmäßig ausführen, um Ihre Paketliste aktuell zu halten.)

- Das Paket `foo` mit allen seinen Abhängigkeiten installieren:

```
apt install foo
```

- Das Paket `foo` von Ihrem System entfernen:

```
apt remove foo
```

- Das Paket `foo` inklusive aller Konfigurationsdateien von Ihrem System entfernen:

```
apt purge foo
```

- Um alle Pakete aufzulisten, für die eine neuere Version verfügbar ist, rufen Sie auf:

```
apt list --upgradable
```

- Eine Aktualisierung aller Pakete auf Ihrem System durchführen (ohne zusätzliche Pakete zu installieren oder Pakete zu entfernen):

```
apt upgrade
```

- Alle Pakete auf Ihrem System aktualisieren (inklusive der Installation von zusätzlichen Paketen oder der Entfernung von Paketen, falls nötig):

¹Beachten Sie, dass es einige (Software)-Portierungen gibt, die dies auch für andere Paketverwaltungssysteme wie den Red-Hat Package-Manager (auch als `rpm` bekannt) ermöglichen.

```
apt full-upgrade
```

(Der Befehl `upgrade` beläßt ein Paket in seiner installierten, veralteten Version, falls zur Aktualisierung aufgrund von Abhängigkeiten die Installation zusätzlicher Pakete erforderlich wäre. Der Befehl `full-upgrade` ist dabei weniger konservativ.)

Zum Ausführen von Befehlen, durch die Pakete verändert werden, müssen Sie als `root`-Benutzer angemeldet sein.

Bitte beachten Sie, dass `apt-get` jetzt standardmäßig auch alle Pakete installiert, die von anderen Paketen empfohlen werden (`recommends`-Abhängigkeit). Aufgrund seiner Robustheit wird das Konsolen-Programm `apt-get` zur Einrichtung und für größere System-Upgrades empfohlen.

Die `apt`-Werkzeugsammlung enthält auch `apt-cache` zum Durchsuchen der Paketliste. Im Paketverwaltungssystem können Sie mit einem einfachen Suchbegriff oder einem regulären Ausdruck oder aber durch das Durchsuchen von Abhängigkeiten Pakete finden, die spezielle Funktionen anbieten. Einige übliche Befehle für `apt-cache` sind:

- Pakete finden, deren Beschreibung *Wort* enthält:

```
apt search Wort
```

- Detaillierte Informationen über ein Paket ausgeben:

```
apt show Paketname
```

- Alle Pakete ausgeben, von denen ein angegebenes Paket abhängt:

```
apt-cache depends Paketname
```

- Detaillierte Informationen über die verfügbaren Versionen eines Paketes und die Pakete, die von diesem Paket abhängen, ausgeben:

```
apt-cache showpkg Paketname
```

Für weitere Informationen installieren Sie bitte das `apt`-Paket und lesen `apt(8)`, `apt-get(8)` und `sources.list(5)`, sowie installieren Sie das `apt-doc`-Paket und lesen `/usr/share/doc/apt-doc/guide.html/index.html`.

8.1.3 aptitude

`aptitude` ist ein auf Debian GNU/Linux-Systeme abgestelltes Frontend für die Paketverwaltung durch APT. `aptitude` ist ein textbasiertes Programm, das die `curses`-Bibliothek benutzt. Aktionen können per interaktiver Bedienoberfläche oder über die Befehlszeile gestartet werden.

`aptitude` kann verwendet werden, um Verwaltungsaufgaben auf schnelle und einfache Weise zu erledigen. Es erlaubt dem Benutzer, die Liste der Pakete zu durchsuchen und Aufgaben wie Installation, Aktualisierung und Entfernung von Paketen zu erledigen.

Über die Funktionen von `apt-get` hinaus bietet `aptitude` viele zusätzliche Funktionalitäten:

- `aptitude` bietet einfachen Zugriff auf alle Versionen eines Pakets.
- `aptitude` gibt Ihnen eine einfache Möglichkeit, die Übersicht über veraltete Software zu behalten: sie werden unter »Veraltete und selbst erstellte Pakete« aufgelistet.
- `aptitude` enthält ein ziemlich mächtiges System für die ausführliche Suche nach Paketen und zur Filterung der angezeigten Treffer. Benutzer, die `mutt` benutzen, werden schnell damit zurechtkommen, da die Ausdruckssyntax durch `mutt` inspiriert war.
- `aptitude` kann auch verwendet werden, um die verfügbaren, vordefinierten Programmgruppen (Tasks) zu installieren. Weitere Informationen finden Sie in 'tasksel' auf der nächsten Seite.
- Im Vollbildmodus hat `aptitude` integrierte `su`-Funktionalität und kann als normaler Benutzer ausgeführt werden. Es ruft `su` auf (und fragt nach dem Passwort des `root`-Benutzers, sofern vorhanden), wenn administrative Rechte benötigt werden.

Sie können `aptitude` mit einer visuellen (interaktiven) Oberfläche (führen Sie einfach `aptitude` aus) oder direkt von der Befehlszeile starten. Die genutzte Befehlszeilensyntax ist der von `apt-get` sehr ähnlich. Um zum Beispiel das Paket `foo` zu installieren, können Sie einfach `aptitude install foo` ausführen.

Mit `aptitude` stellt Debian ein Programm für das alltägliche Paketmanagement von der Konsole aus bereit.

Weitere Informationen finden Sie auf der Handbuchseite `aptitude(8)` und im Paket `aptitude-doc-en`.

8.1.4 synaptic

`synaptic` ist ein grafisches Paketverwaltungs-Programm. Es ermöglicht Ihnen das Installieren, Aktualisieren und Entfernen von Paketen auf eine benutzerfreundliche Art und Weise. Neben all den Funktionalitäten, die `aptitude` bietet, erlaubt `synaptic` auch, die Liste der verwendeten Paketquellen zu editieren; auch können Sie alle verfügbare Dokumentation für ein Paket durchsuchen. Schauen Sie auf die Synaptic-Website (<http://www.nongnu.org/synaptic/>), wenn Sie mehr Informationen benötigen.

8.1.5 tasksel

Rechner fallen vielfach klar umrissene Aufgaben zu, doch häufig ergibt sich die Schwierigkeit, den zu einer Aufgabe passenden Satz von Paketen zu finden. Die Debian-Entwickler haben daher Programmgruppen (`Tasks`) definiert, eine Sammlung verschiedener Debian-Pakete für eine spezielle Aufgabe. Diese Programmgruppen können über die Programme `tasksel` oder `aptitude` installiert werden.

Der Debian Installer wird typischerweise automatisch die Programmgruppe für ein Standardsystem installieren sowie eine Arbeitsplatzumgebung (Desktop Environment). Welche Arbeitsplatzumgebung installiert wird, hängt von dem benutzten CD-/DVD-Installationsmedium ab, normalerweise ist dies der GNOME-Desktop (Programmgruppe `gnome-desktop`). Außerdem werden, abhängig von den Auswahlen, die Sie während der Installation getroffen haben, eventuell noch weitere Programmgruppen automatisch installiert. Wenn Sie zum Beispiel eine andere Sprache als Englisch gewählt haben, wird auch die entsprechende Programmgruppe für diese Sprache installiert; wenn der Installer einen Laptop als Installationsziel erkennt, wird die `laptop`-Programmgruppe installiert.

8.1.6 Andere Paketverwaltungswerkzeuge

`dpkg-deb`

Dieses Programm führt Operationen an Debian-Archivdateien (`.deb`) aus. Einige Anwendungsfälle sind:

- Alle Optionen anzeigen: `dpkg-deb --help`.
- Anzeigen, welche Dateien in einem Debian-Archiv enthalten sind: `dpkg-deb --contents foo_VVV-RRR.deb`.
- Die Dateien, die in dem Debian-Archiv enthalten sind, in ein gewünschtes Verzeichnis entpacken: `dpkg-deb --extract foo_VVV-RRR.deb tmp` entpackt jede Datei in `foo_VVV-RRR.deb` in das Verzeichnis `tmp/`. Dies ist eine bequeme Methode, um die Inhalte eines Pakets in einem lokalen Verzeichnis zu prüfen, ohne das Paket in das Root-Dateisystem zu installieren.
- Die Dateien mit den control-Informationen aus dem Paket entpacken: `dpkg-deb --control foo_VVV-RRR.deb tmp`.

Bitte beachten Sie, dass jedes Paket, das lediglich mit `dpkg-deb --extract` entpackt wird, falsch installiert ist. Stattdessen sollten Sie `dpkg --install` benutzen, um das Paket zu installieren.

Weitere Informationen finden Sie auf der Handbuchseite `dpkg-deb(1)`.

8.2 Angeblich ist es mit Debian möglich, ein derzeit laufendes Programm zu aktualisieren. Wie funktioniert das?

Die Implementierung des Dateisystems im Kernel unterstützt auf Debian GNU/Linux-Systemen das Ersetzen von Dateien, auch während sie benutzt werden.

Außerdem stellen wir ein Programm namens `start-stop-daemon` zur Verfügung, das benutzt wird, um Daemons während des Hochfahrens oder beim Wechsel des Runlevels (z.B. vom Mehrbenutzer- in den Einbenutzer-Modus oder zum Runterfahren) zu starten oder zu stoppen. Dasselbe Programm nutzen Installationskripte für Softwarepakete, die unter Einbindung von Hintergrundprozessen oder direkt als solche arbeiten. Diese werden durch `start-stop-daemon` beendet und falls nötig neu gestartet.

8.3 Wie kann ich feststellen, welche Pakete bereits auf einem Debian-System installiert sind?

Um den Status aller in einem Debian-System installierten Pakete zu sehen, führen Sie den Befehl

```
dpkg --list
```

aus. Es wird eine einzeilige Zusammenfassung für jedes Paket ausgegeben, mit einem 2-stelligen Buchstaben-Code (im Kopf erklärt), dem Paketnamen, der *installierten* Version und einer kurzen Beschreibung.

Um den Status jedes Paketes zu sehen, dessen Name mit »foo« beginnt, geben Sie folgenden Befehl ein:

```
dpkg --list 'foo*'
```

Wünschen Sie einen ausführlicheren Status für ein bestimmtes Paket, geben Sie dies ein:

```
dpkg --status Paketname
```

8.4 Wie kann ich alle Dateien auflisten, die in einem installierten Paket enthalten sind?

Um alle mit der Installation des Pakets `foo` auf das System übertragenen Dateien aufzulisten, verwenden Sie den Befehl:

```
dpkg --listfiles foo
```

Beachten Sie, dass Dateien, die von den Installationskripten erzeugt werden, hier nicht auftauchen.

8.5 Wie kann ich herausfinden, welches Paket eine bestimmte Datei angelegt hat?

Um herauszufinden, aus welchem Paket eine bestimmte Datei stammt, führen Sie einen dieser Befehle aus:

- `dpkg --search foo`

Dies sucht nach `foo` in den installierten Paketen. (Dies ist momentan gleichwertig mit der Suche nach allen Dateien mit der Endung `.list` im Verzeichnis `/var/lib/dpkg/info/` und dem Filtern der Ausgabe, so dass alle Pakete, die dies beinhalten, aufgelistet werden plus Umleitungen.)

Eine schnellere Alternative dazu ist das `dlocate`-Werkzeug:

```
dlocate -S foo
```

- `zgrep foo Contents-ARCH.gz`

Hiermit wird nach Dateien gesucht, die die Zeichenkette `foo` in ihrem vollständigen Pfadnamen enthalten. Die Dateien `Contents-ARCH.gz` (wobei `ARCH` die gewünschte Architektur darstellt) befinden sich in den Hauptpaketverzeichnissen (`main`, `non-free`, `contrib`) auf dem Debian-FTP-Server (d.h. unter `/debian/dists/jessie`). Eine `Contents`-Datei bezieht sich nur auf Pakete in den Unterverzeichnissen unterhalb des Verzeichnisses, in dem es abgelegt ist. Deshalb könnte es sein, dass Sie mehr als eine `Contents`-Datei durchsuchen müssen, um das Paket zu finden, das `foo` enthält.

Diese Methode hat gegenüber `dpkg --search` den Vorteil, dass Sie damit Dateien in Paketen finden können, die momentan nicht auf Ihrem System installiert sind.

- `apt-file search foo`

Ähnlich wie bei dem Beispiel vorher sucht dies nach Dateien, welche die Zeichenfolge oder den regulären Ausdruck `foo` in ihrem vollständigen Pfadnamen haben. Der Unterschied zum vorherigen Beispiel ist, dass es nicht notwendig ist, die `Contents-ARCH.gz`-Dateien händisch herunterzuladen, da dies automatisch für alle Quellen in der `/etc/apt/sources.list` erledigt wird, wenn Sie `apt-file update` (als `root`) ausführen.

8.6 Warum wird »foo-data« nicht entfernt, wenn ich »foo« deinstalliere? Wie stelle ich sicher, dass alte unbenutzte Bibliothekspakete restlos entfernt werden?

Einige Software ist in Programmpakete (»foo«) und Datenpakete (»foo-data«) (oder »foo« und »foo-doc«) aufgeteilt. Dies trifft in Debian auf viele Spiele, Multimedia-Anwendungen und Wörterbücher zu und wurde eingeführt, weil einige Benutzer auf die Rohdaten zugreifen wollten, ohne das Programm zu installieren, oder weil das Programm auch ohne die Daten läuft, so dass »foo-data« optional ist.

Ähnliche Situationen tauchen auf, wenn Sie mit Bibliotheken arbeiten: normalerweise werden diese Pakete installiert, wenn Pakete Programme enthalten, die von ihnen abhängen. Wenn das Programmpaket entfernt wird, bleibt das Bibliothekspaket auf dem System. Oder: Wenn das Programmpaket nicht länger z.B. von »libdb4.2« abhängt, aber stattdessen jetzt von »libdb4.3«, kann das »libdb4.2«-Paket installiert bleiben, wenn das Programmpaket aktualisiert wird.

In diesen Fällen hängt »foo-data« nicht von »foo« ab, so dass, wenn Sie das »foo«-Paket entfernen, die meisten Paketverwaltungswerkzeuge es nicht automatisch entfernen. Dasselbe gilt für die Bibliothekspakete. Dies ist nötig, um zirkuläre Abhängigkeiten zu vermeiden. Wenn Sie jedoch `apt-get` (siehe 'APT' auf Seite 36) oder `aptitude` (siehe 'aptitude' auf Seite 37) als Paketverwaltungswerkzeug benutzen, werden sich diese die automatisch installierten Pakete merken und sie entfernen, wenn kein Paket mehr auf Ihrem System existiert, das diese nutzt.

Kapitel 9

Wie man sein Debian-System auf aktuellem Stand hält

Debian legt Wert darauf, dass der Ablauf von Systemaktualisierungen in sich stimmig und sicher ist. Wir tun alles, um Ihnen Schwierigkeiten beim Upgrade auf eine neue Veröffentlichung zu ersparen. Falls für ein Paket wichtige Hinweise vorliegen, werden diese während des Upgrades ausgegeben und dem Benutzer häufig auch Lösungen für mögliche Probleme angeboten.

Es ist ratsam, sich die Veröffentlichungshinweise (Release Notes) durchzulesen, welche Einzelheiten zu den jeweiligen Upgrades enthalten. Diese sind auf allen Debian-CDs/DVDs/Blue-ray-Disks zu finden und auch im WWW unter <http://www.debian.org/releases/stable/releasenotes> verfügbar.

9.1 Wie kann ich mein Debian-System aktuell halten?

Man könnte sich einfach anonym per FTP an einem Debian-Spiegelserver anmelden, die Verzeichnisse durchgehen, sich gewünschte Dateien herunterladen und dann mit `dpkg` installieren. Seien Sie sich darüber im klaren, dass `dpkg` selbst bei laufenden Systemen das Upgrade auf der Stelle einleiten wird. Manchmal ist es erforderlich, dass für ein geändertes Paket eine aktuelle Version eines anderen Paketes installiert werden muss. In diesem Falle wird die Installation fehlschlagen, bis/außer wenn das andere Paket installiert ist.

Viele Leute finden diese Vorgehensweise viel zu zeitraubend, weil Debian sich so schnell entwickelt – üblicherweise werden jede Woche ein Dutzend oder mehr neue Pakete hochgeladen. Kurz vor einer neuen Hauptveröffentlichung ist diese Anzahl noch größer. Um mit dieser Lawine umgehen zu können, bevorzugen viele Leute eine automatisierte Methode. Zu diesem Zweck sind mehrere verschiedene Ansätze verfügbar:

9.1.1 aptitude

`aptitude` ist der empfohlene Paketmanager für Debian GNU/Linux-Systeme und in 'aptitude' auf Seite 37 beschrieben.

Bevor `aptitude` für ein Upgrade benutzt werden kann, muss die `/etc/apt/sources.list`-Datei entsprechend angepasst werden. Wenn Sie eine Aktualisierung auf die neueste stabile Debian-Version durchführen möchten, sollten Sie wahrscheinlich eine Quelle wie die folgende benutzen:

```
http://ftp.de.debian.org/debian stable main contrib
```

`ftp.us.debian.org` (ein Spiegel-Server in den Vereinigten Staaten) kann durch den Namen eines schnelleren Debian-Spiegels in Ihrer Nähe ersetzt werden. Sie finden eine Liste der Spiegel-Server sowie weitere Informationen auf <http://www.debian.org/misc/README.mirrors>.

Oder Sie benutzen den Weiterleitungs-Service `httpredir.debian.org`, der die Aufgabe der Spiegelauswahl für Sie übernimmt. Dabei werden der geografische Standort des Benutzers und weitere Informationen verwendet, um den besten verfügbaren Spiegel auszuwählen, der die Dateien liefern kann. Um Gebrauch davon zu machen, verwenden Sie eine Paketquelle wie diese:

```
http://httpredir.debian.org/debian stable main contrib
```

Weitere Einzelheiten hierzu können Sie der Handbuchseite `sources.list(5)` entnehmen.

Zum Aktualisieren des Systems ist folgender Befehl auszuführen:

```
aptitude update
```

gefolgt von

```
aptitude full-upgrade
```

Beantworten Sie eventuelle Fragen, und das System wird aktualisiert.

Beachten Sie, dass `aptitude` nicht das empfohlene Werkzeug für ein Upgrade auf eine neue Debian GNU/Linux-Veröffentlichung ist. Verwenden Sie stattdessen `apt-get`. Sie sollten vor einem Upgrade auch die Veröffentlichungshinweise (<http://www.debian.org/releases/stable/releasenotes>) lesen. Diese beschreiben zum einen detailliert, wie das Upgrade erfolgen sollte und nennen Punkte, mit denen Sie sich vor dem Upgrade befassen sollten.

Details finden Sie in der Handbuchseite `aptitude(8)` und in der Datei `/usr/share/aptitude/README`.

9.1.2 apt-get und apt-cdrom

Eine Alternative zu `aptitude` ist `apt-get`, ein APT-basiertes Befehlszeilenprogramm (beschrieben in 'APT' auf Seite 36). `apt-get`, das APT-basierte Befehlszeilen-Werkzeug zur Paketverwaltung, bietet einen einfachen und sicheren Weg zur Installation und Aktualisierung von Paketen.

Um `apt-get` zu benutzen, editieren Sie die Datei `/etc/apt/sources.list`, genauso wie für 'aptitude' auf der vorherigen Seite.

Führen Sie dann aus:

```
apt-get update
```

gefolgt von

```
apt-get dist-upgrade
```

Beantworten Sie eventuelle Fragen, und das System wird aktualisiert. Schauen Sie auch auf die Handbuchseite `apt-get(8)` oder in 'APT' auf Seite 36.

Wenn Sie Pakete von CD/DVD/BD installieren möchten, können Sie `apt-cdrom` benutzen. Weitere Einzelheiten sind in den Veröffentlichungshinweisen im Abschnitt »APT-Quellen für optische Medien hinzufügen« zu finden.

Es ist zu beachten, dass die Pakete, nachdem sie heruntergeladen und installiert wurden, immer noch in einem Verzeichnis unterhalb von `/var` verbleiben. Um die Partition vor Überfüllung zu bewahren, sollten Sie daran denken, solche Dateien mit `apt-get clean` oder `apt-get autoclean` zu löschen, oder sie an einen anderen Ort verschieben (Tipp: benutzen Sie hierfür `apt-move`).

9.1.3 mirror

Dieses Perl-Skript und das dazugehörige (optionale) Verwaltungsprogramm namens `mirror-master` kann vom Benutzer angegebene Teile eines Verzeichnisbaums in einer anonymen FTP-Sitzung von einem vorgegebenen Server abrufen.

`mirror` ist besonders dafür geeignet, große Mengen an Software herunterzuladen. Nachdem zum ersten Mal Dateien von einem Server heruntergeladen wurden, wird eine Datei namens `.mirrorinfo` auf dem lokalen Rechner angelegt. Veränderungen im fernen Dateisystem werden automatisch durch `mirror` verfolgt; dabei wird diese Datei mit einer ähnlichen Datei auf dem fernen System verglichen und nur geänderte Dateien werden heruntergeladen.

Das Programm `mirror` ist hauptsächlich dafür geeignet, lokale Kopien von fernen Verzeichnisbäumen zu aktualisieren. Die abgerufenen Dateien müssen keine Debian-Dateien sein. (Da `mirror` ein Perl-Skript ist, läuft es auch auf nicht-Unix-Systemen.) Obwohl es mit dem Programm `mirror` möglich ist, Dateien auszuschließen, deren Namen benutzerdefinierte Zeichenketten enthalten, ist dieses Programm am nützlichsten, wenn komplette Verzeichnisbäume heruntergeladen werden sollen anstatt ausgewählter Pakete.

9.2 Muss ich in den Einzelbenutzermodus wechseln, um ein Paket zu aktualisieren?

Nein. Pakete können direkt aktualisiert werden, sogar in laufenden Systemen. Debian verfügt über ein `start-stop-daemon`-Programm, das - falls notwendig - während eines Paket-Upgrades aufgerufen wird, um laufende Prozesse anzuhalten und wieder zu starten.

9.3 Muss ich all diese .deb-Archivdateien auf meinem Rechner aufbewahren?

Nein. Diese Dateien können nach erfolgreicher Installation der Pakete wieder entfernt werden, z.B. mit `aptitude clean`.

9.4 Wie kann ich protokollieren, welche Pakete ich meinem System hinzugefügt habe? Ich möchte gerne wissen, wann welche Pakete aktualisiert oder entfernt wurden!

Durch das Anhängen der Option `--log=Dateiname` wird `dpkg` veranlasst, ein Statusprotokoll zu erstellen, das alle Änderungen am Installationsstatus und alle Paketoperationen auflistet. Es 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 das Ergebnis, also z.B.

```
2005-12-30 18:10:35 status installed hello 2.1.1-4.
```

Wenn jeder Aufruf von `dpkg` protokolliert werden soll, (auch falls dieser über andere Programme wie `aptitude` geschieht), kann

```
log /var/log/dpkg.log
```

zur `/etc/dpkg/dpkg.cfg`-Datei hinzugefügt werden. Stellen Sie sicher, dass die erstellte Protokolldatei regelmäßig rotiert wird. Wenn Sie `logrotate` benutzen, können Sie dies erreichen, indem Sie eine Datei `/etc/logrotate.d/dpkg` mit folgendem Inhalt erstellen:

```
/var/log/dpkg {  
    missingok  
    notifempty  
}
```

Weitere Details über das Protokollieren von `dpkg`-Operationen finden Sie auf der Handbuchseite `dpkg(1)`.

`aptitude` protokolliert beabsichtigte Installationen, Entfernungen und Aktualisierungen in der Datei `/var/log/aptitude`, wobei die *Ergebnisse* dieser Aktivitäten nicht in dieser Datei aufgezeichnet werden!

Eine andere Möglichkeit, Paketaktivitäten aufzuzeichnen besteht darin, die Paketverwaltungs-Sitzung innerhalb des `script(1)`-Programms laufen zu lassen.

9.5 Kann ich das System automatisch aktualisieren lassen?

Ja. Sie können `cron-apt` benutzen; dieses Werkzeug aktualisiert über einen cron-Job regelmäßig das System. In der Standardeinstellung aktualisiert es lediglich die Paketliste und lädt neue Pakete herunter, installiert diese jedoch nicht.

Bei *Testing*- oder *Unstable*-Systemen wird von der automatischen Aktualisierung von Paketen **abgeraten**, da ein unerwartetes Verhalten oder Paketentfernungen auftreten können, ohne dass dies bemerkt würde.

9.6 Ich habe mehrere Rechner; wie kann ich erreichen, dass ich die Updates nur einmal herunterladen muss?

Falls Sie mehr als einen Debian-Rechner in Ihrem Netzwerk haben, ist `apt-cacher` nützlich, um all Ihre Debian-Systeme aktuell zu halten.

`apt-cacher` reduziert die Bandbreiten-Anforderungen auf Seiten der Debian-Spiegelserver, da »Packages«, »Releases«- und »Sources«-Dateien weniger häufig für Aktualisierungen angefordert werden und jede Datei immer nur einmal heruntergeladen wird, unabhängig von den wirklichen Anfragen des Proxys. `apt-cacher` erstellt automatisch einen Debian-HTTP-Spiegel, basierend auf den Anfragen, die durch den Proxy gehen.

Natürlich können Sie die gleichen Vorteile auch genießen, wenn Sie einen Standard-Caching-Proxy nutzen und alle Ihre Systeme so konfiguriert sind, diesen zu nutzen.

Kapitel 10

Debian und der Kernel

10.1 Kann ich einen Kernel ohne gewisse Debian-spezifische Optimierungen installieren und kompilieren?

Ja.

Es gibt nur ein häufiges Problem: Die Debian C-Bibliotheken werden mit den neuesten *stabilen* Ausgaben der **Kernel-Header** erstellt. Wenn Sie ein Programm mit neueren Kernel-Headern als jenen aus dem Stable-Zweig kompilieren möchten, sollten Sie entweder das Paket, welches die Header enthält (`linux-libc-dev`) aktualisieren oder die neuen Header eines entpackten Baums des neuen Kernels verwenden. Das heißt, wenn die Kernel-Quellen in `/usr/src/linux` liegen, sollten Sie beim Kompilieren `-I/usr/src/linux/include/` an die Befehlszeile anhängen.

10.2 Welche Werkzeuge zum Erzeugen eines angepassten Kernels stellt Debian zur Verfügung?

Benutzer, die einen eigenen angepassten Kernel erstellen möchten (oder müssen), sollten das Debian-Paket-Target verwenden, das in den aktuellen Versionen des Kernel-Build-Systems enthalten ist. Nach dem Konfigurieren des Kernels führen Sie einfach folgenden Befehl aus:

```
make deb-pkg
```

Das neue Kernel-Paket wird in dem Verzeichnis eine Ebene oberhalb des Kernel-Quellcode-Baums erstellt, und es kann mittels `dpkg -i` installiert werden.

Anwender müssen sich den Quellcode des neuesten Kernels oder des Kernels ihrer Wahl separat von der von ihnen bevorzugten Linux-Archiv-Website herunterladen, es sei denn, ein `kernel-source-version`-Debian-Paket ist verfügbar (wobei *version* für die Version des Kernels steht).

10.3 Welche Hilfsmittel stellt Debian für den Umgang mit Modulen bereit?

Eine Konfigurationsdatei, die die Module enthält, welche beim Booten des Systems geladen werden sollen, ist `/etc/modules`. Allerdings müssen Sie diese Datei nur selten verändern.

Weitere Konfigurationsdaten für Kernel-Module werden in dem Verzeichnis `/etc/modprobe.d/` abgelegt. Weitere Informationen über das Format der Dateien finden Sie auf der `modprobe.conf(5)` Handbuchseite.

10.4 Kann ich sicher ein altes Kernel-Paket deinstallieren und wenn ja, wie?

Ja. Das `kernel-image-NNN.prerm`-Skript kontrolliert, ob der Kernel, den Sie gerade verwenden, derselbe ist wie jener, den Sie zu entfernen versuchen. Daher können Sie nicht mehr benötigte Kernel-Image-Pakete mit diesem Befehl entfernen:

```
dpkg --purge linux-image-NNN
```

(Natürlich müssen Sie *NNN* durch die Kernelversion und Revisionsnummer des entsprechenden Kernels ersetzen.)

10.5 Wo finde ich mehr Informationen über Linux-Pakete für Debian?

Weitere Informationen werden im Debian Linux Kernel Handbook (<http://kernel-handbook.alioth.debian.org/>) gepflegt.

Kapitel 11

Anpassen Ihres Debian GNU/Linux-Systems

11.1 Wie kann ich sicherstellen, dass alle Programme dieselbe Papiergröße verwenden?

Installieren Sie das Paket `libpaper1` und es wird nach einer systemweiten Standardpapiergröße fragen. Diese Einstellung wird in der Datei `/etc/papersize` gespeichert.

Benutzer können sich über die Papiergrößen-Einstellung hinwegsetzen, indem sie die `PAPERSIZE`-Umgebungsvariable verwenden. Für Details lesen Sie die Handbuchseite `papersize(5)`.

11.2 Wie lassen sich ohne Abstriche bei der Sicherheit Peripheriegeräte zur Verfügung stellen?

Viele Gerätedateien im `/dev/-`Verzeichnis gehören zu einer vordefinierten Gruppe. Zum Beispiel gehört `/dev/sr0` zu der Gruppe `cdrom`.

Um einem bestimmten Benutzer Zugriff auf eines dieser Geräte einzuräumen, fügen Sie ihn zu der Gruppe hinzu, der das Gerät gehört, also z.B.:

```
adduser user group
```

Auf diese Art müssen Sie die Dateiberechtigungen des Gerätes nicht ändern.

Wenn Sie diese Änderung auf der Shell eines Benutzers oder einer grafischen Benutzerumgebung vornehmen, müssen Sie sich anschließend einmal abmelden und wieder neu anmelden, damit die Änderungen wirksam werden. Um herauszufinden, welchen Gruppen Sie angehören, verwenden Sie den Befehl `groups`.

Hinweis: Seit der Einführung von `udev` kann es vorkommen, dass Ihrerseits geänderte Berechtigungen für Peripheriegeräte nach einem Systemstart wieder zurückgesetzt sind. Falls für Sie relevante Geräte davon betroffen sind, müssen Sie die Regeln in `/etc/udev` anpassen.

11.3 Wie kann ich eine Konsolen-Schriftart auf Debian-Art beim Systemstart laden?

Das Paket `kbd` lässt sich entsprechend konfigurieren. Editieren Sie dazu die Datei `/etc/kbd/config`.

11.4 Wie kann ich die Standardeinstellung eines X11-Programms konfigurieren?

Die Debian-X-Programme installieren ihre Anwendungs-Ressourcen-Daten im Verzeichnis `/etc/X11/app-defaults/`. Wenn Sie eine X-Anwendung global anpassen wollen, schreiben Sie Ihre Anpassungen in diese Dateien. Sie sind als Konfigurationsdateien markiert, so dass ihre Inhalte bei einem Upgrade erhalten bleiben.

11.5 Jede Distribution scheint eine andere Boot-Methode zu haben. Beschreiben Sie jene von Debian.

Wie alle Unix-Systeme bootet Debian durch das Ausführen des Programms `init`¹. `/etc/inittab`, die Konfigurationsdatei für `init`, legt fest, dass `/etc/init.d/rcS` als erstes Skript ausgeführt werden soll. Dieses Skript startet alle Skripte im `/etc/rcS.d/`-Verzeichnis, indem es sie als eigene Unterprozesse laufen lässt; dabei werden Initialisierungen durchgeführt, wie zum Beispiel Dateisysteme prüfen und einbinden, Module laden, Netzwerkdienste starten, die Uhr stellen und weitere.

Nachdem der Boot-Prozess abgeschlossen ist, führt `init` alle Start-Skripte in dem zum Standard-Runlevel gehörenden Verzeichnis aus. Dieses Runlevel wird durch den Eintrag `id` in `/etc/inittab` festgelegt. Wie alle zu System V kompatiblen Unix-Systeme hat Linux 7 Runlevel:

- 0 (das System anhalten),
- 1 (Einzelbenutzer-Modus),
- 2 bis 5 (verschiedene Mehrbenutzer-Modi) und
- 6 (das System neu starten).

Debian-Systeme haben »id=2« als Standardeinstellung, was bedeutet, dass der Standard-Runlevel »2« ist, wenn das Mehrbenutzer-Level gestartet wird, und dass die Skripte in `/etc/rc2.d/` ausgeführt werden.

Debian verwendet abhängigkeits-basierte Boot-Reihenfolgen durch Nutzung von `insserv`; dafür werden die LSB-Header in jedem Skript unterhalb von `/etc/init.d/` verwendet. Außerdem setzt Debian `parallele` (gleichzeitig laufende) Boot-Prozesse mittels `startpar` ein, um den Boot-Vorgang zu beschleunigen.

Letzten Endes sind die Skripte in jedem der Verzeichnisse `/etc/rcN.d/` nur symbolische Verweise zurück auf Skripte in `/etc/init.d/`. Allerdings geben die *Namen* dieser Dateien an, *wie* die Skripte in `/etc/init.d/` ausgeführt werden. Konkret werden vor dem Aktivieren eines Runlevels alle Skripte, die mit einem »K« beginnen, ausgeführt; diese Skripte beenden Dienste. Danach werden alle Skripte, die mit einem »S« beginnen, ausgeführt; diese Skripte starten Dienste. Die zweistellige Zahl, die hinter dem »K« bzw. »S« folgt, bestimmt die Reihenfolge, in welcher die Skripte ausgeführt werden. Jene mit kleinerer Nummer werden zuerst ausgeführt.

Die Methode beruht darauf, dass jedes der Skripte in `/etc/init.d/` eines von fünf möglichen Argumenten, nämlich »start«, »stop«, »reload«, »restart« oder »force-reload« erwartet und den jeweiligen Dienst dann entsprechend steuert. Diese Skripte können auch nach dem Starten des Systems noch benutzt werden, um verschiedene Prozesse zu kontrollieren.

Folgender Befehl mit dem Argument »reload«

```
/etc/init.d/sendmail reload
```

sendet dem `sendmail`-Daemon ein Signal, seine Konfigurationsdatei neu einzulesen.

Beachten Sie, dass `invoke-rc.d` nicht benutzt werden sollte, um die Skripte in `/etc/init.d/` aufzurufen; verwenden Sie stattdessen `service`.

11.6 Welche Möglichkeiten außer `rc.local` gibt es zum Anpassen des Bootprozesses?

Das `rc.local`-Skript wird als letztes in jedem Mehrbenutzer-Runlevel ausgeführt. In Debian bewirkt es standardmäßig nichts. Dies gestattet es, den Boot-Prozess an eigene Anforderungen anzupassen, wobei man damit unter Umständen nicht jeder Situation gerecht wird.

Nehmen wir an, ein System soll beim Hochfahren oder bei Aktivierung eines bestimmten (System-V-) Runlevels das Skript `foo` ausführen. Dann sollte der Systemadministrator:

- Das Skript `foo` ins Verzeichnis `/etc/init.d/` einfügen.

¹Als dies geschrieben wurde, ist in Unstable für Debian GNU/Linux das Init-System `sysvinit` gerade durch `systemd` ersetzt worden, einem System- und Dienste-Manager für Linux. Weitere Informationen finden Sie unter <https://wiki.debian.org/systemd>.

- Den Debian-Befehl `update-rc.d` mit den passenden Argumenten ausführen, um anzugeben, welche Runlevel den Dienst starten und welche ihn stoppen sollen.
- Das System am besten neu starten, um zu überprüfen, ob der Dienst richtig startet (vorausgesetzt, dass der Dienst in einem der Standard-Runlevel starten soll). Andernfalls starten Sie den Dienst über `»/etc/init.d/foo start«`.

Man könnte zum Beispiel das Skript `foo` beim Systemstart ausführen lassen, indem man es in `/etc/init.d/` einfügt und `update-rc.d foo defaults 19` aufruft. Das Argument `»defaults«` bezieht sich auf die Standard-Runlevel, also 2 bis 5. Solange es keinen LSB-Kommentar-Block gibt, der dem widerspricht, führt das dazu, dass der Dienst in den Runleveln 2 bis 5 gestartet und in den Runleveln 0, 1 und 6 gestoppt wird. (Sämtliche `»LSB Default-Start«`- und `»LSB Default-Stop«`-Anweisungen in `foo` haben Vorrang, wenn die `sysv-rc`-Version von `update-rc.d` verwendet wird, sie werden aber von der `file-rc`-Variante von `update-rc.d` (v0.8.10 und höher) ignoriert.) Das Argument `»19«` führt dazu, dass `foo` aufgerufen wird, sobald alle Skripte mit Nummern kleiner 19 beendet wurden und vor allen Skripten mit Nummer 20 oder höher.

11.7 Wie geht das Paketverwaltungssystem mit Paketen um, die Konfigurationsdateien für andere Pakete enthalten?

Manche Benutzer möchten zum Beispiel einen neuen Server einrichten, indem sie eine Gruppe von Debian-Paketen installieren sowie ein lokal generiertes Paket, welches aus Konfigurationsdateien besteht. Dies ist grundsätzlich keine gute Idee, weil `dpkg` nichts über diese Konfigurationsdateien weiß, wenn sie in einem anderen Paket enthalten sind. `dpkg` könnte daher unpassende Konfigurationen schreiben, wenn eines der ursprünglichen Pakete aktualisiert wird.

Deshalb ist es besser, ein lokales Paket zu erstellen, das die Konfigurationsdateien einer `»Gruppe«` von Debian-Paketen modifiziert. Dann registriert `dpkg` und der Rest des Paketverwaltungssystems, dass diese Dateien durch den lokalen Systemverwalter modifiziert worden sind und versucht nicht, sie bei einer Paketaktualisierung zu überschreiben.

11.8 Wie kann ich eine durch ein Paket installierte Datei außer Kraft setzen, so dass stattdessen eine andere Version verwendet werden kann?

Nehmen wir an, dass ein Systemadministrator oder ein lokaler Benutzer lieber ein Programm namens `login-local` anstatt dem vom Debian-Paket `login` zur Verfügung gestellten Programm `login` verwenden möchte.

Sie sollten **nicht**:

- `/bin/login` mit `login-local` überschreiben.

Das Paketverwaltungssystem weiß nichts über diese Veränderung und wird einfach Ihr angepasstes `/bin/login` überschreiben, wann immer `login` (oder jedes andere Paket, welches `/bin/login` bereitstellt) installiert oder aktualisiert wird.

Tun Sie besser Folgendes:

- Führen Sie folgenden Befehl aus:

```
dpkg-divert --divert /bin/login.debian /bin/login
```

Dies führt dazu, dass bei allen zukünftigen Installationen des Debian-Pakets `login` die Datei `/bin/login` nach `/bin/login.debian` geschrieben wird.

- Und dann:

```
cp login-local /bin/login
```

Dadurch wird Ihr eigenes lokal erstelltes Programm an den richtigen Ort kopiert.

Führen Sie `dpkg-divert --list` aus, um zu sehen, welche Umleitungen auf Ihrem System aktiv sind.

Details dazu finden Sie auf der Handbuchseite `dpkg-divert(8)`.

11.9 Wie kann ich meine lokal erstellten Pakete in die Liste der verfügbaren Pakete, die dem Paketmanagementsystem bekannt sind, aufnehmen?

Führen Sie folgenden Befehl aus:

```
dpkg-scanpackages BIN_DIR OVERRIDE_FILE [PATHPREFIX] > meine_Pakete
```

Dabei ist:

- `BIN_DIR` ein Verzeichnis, in welchem Debian-Archiv-Dateien (üblicherweise mit der Endung ».deb«) gespeichert sind.
- `OVERRIDE_FILE` eine Datei, die von den Betreuern der Distribution editiert worden ist und für Debian-Pakete aus der »Main«-Distribution normalerweise im Debian-FTP-Archiv unter `indices/override.main.gz` liegt. Für lokale Pakete entfällt dieses Argument.
- `PATHPREFIX` eine *optionale* Zeichenkette, die der zu erstellenden Datei `meine_Pakete` vorangestellt werden kann.

Wenn Sie die Datei `meine_Pakete` erstellt haben, teilen Sie dies dem Paketverwaltungssystem über folgenden Befehl mit:

```
dpkg --merge-avail meine_Pakete
```

Wenn Sie APT verwenden, können Sie das lokale Paketdepot auch zu Ihrer `sources.list` (5) -Datei hinzufügen.

11.10 Einige Benutzer mögen mawk, andere gawk; einige mögen vim, andere elvis; einige trn, wieder andere tin; wie unterstützt Debian die Vielfalt?

Es gibt verschiedene Fälle, in denen zwei Pakete zwei verschiedene Versionen eines Programms zur Verfügung stellen, wobei beide dieselben Grundfunktionen beherrschen. Benutzer mögen eines dem anderen aus Gewohnheit vorziehen, oder weil die Benutzerschnittstelle des einen Pakets auf irgendeine Art attraktiver ist als jene eines anderen. Andere Benutzer auf demselben System könnten eine andere Wahl treffen.

Wenn zwei oder mehr Hilfsprogramme dieselben Grundfunktionen aufweisen und eine generell formulierte Paketabhängigkeit erfüllen, ermöglicht es Debian Systemadministratoren oder Benutzern, durch ein »virtuelles« Paketystem festzulegen, welches vorgezogen wird.

Zum Beispiel könnten zwei verschiedene Versionen eines Newsreaders auf einem System existieren. Das Newsserver-Paket könnte »empfehlen«, dass *überhaupt ein* Newsreader auf dem System installiert ist, aber die Wahl von `tin` oder `trn` ist dem jeweiligen Benutzer überlassen. Dies wird erreicht, indem die beiden Pakete `tin` und `trn` das virtuelle Paket `news-reader` bereitstellen. *Welches* der Programme tatsächlich aufgerufen wird, wird durch einen Verweis von `/etc/alternatives/news-reader` (dem Namen des virtuellen Pakets) auf die Binärdatei des ausgewählten Programms (z.B. `/usr/bin/trn`) bestimmt.

Für die Nutzung alternativer Programme reicht eine einzelne symbolische Verknüpfung nicht aus. Der Auswahlmechanismus muss auch Handbuchseiten und sonstige zugehörige Dateien berücksichtigen. Das Perl-Skript `update-alternatives` stellt sicher, dass alle zu einem ausgewählten Paket gehörenden Dateien systemweit als Standard herangezogen werden.

Um zum Beispiel zu kontrollieren, welche ausführbaren Dateien »x-window-manager« bereitstellen, führen Sie folgenden Befehl aus:

```
update-alternatives --display x-window-manager
```

Wenn Sie dies ändern möchten, verwenden Sie

```
update-alternatives --config x-window-manager
```

und folgen den Anweisungen auf dem Bildschirm (einfach gesagt: geben Sie die Zahl für den Eintrag ein, den Sie bevorzugen).

Wenn ein Paket sich selbst aus irgend einem Grund nicht als Fenstermanager registriert (senden Sie einen Fehlerbericht, falls es ein Fehler ist), oder wenn Sie einen Fenstermanager aus dem `/usr/local/-`Verzeichnis verwenden, werden die Auswahlmöglichkeiten auf dem Bildschirm Ihren bevorzugten Eintrag nicht enthalten. Sie können den Verweis aber über Befehlszeilen-Optionen aktualisieren:

```
update-alternatives --install /usr/bin/x-window-manager \  
x-window-manager /usr/local/bin/wmaker-cvs 50
```

Das erste Argument für die »`--install`«-Option ist der symbolische Verweis, der auf `/etc/alternatives/NAME` zeigt, wobei `NAME` das zweite Argument ist. Das dritte Argument ist das Programm, auf welches `/etc/alternatives/NAME` zeigen soll, und das vierte Argument die Priorität (ein größerer Wert bedeutet, dass diese Alternative wahrscheinlicher automatisch ausgewählt wird).

Um eine Alternative, die Sie hinzugefügt haben, wieder zu entfernen, führen Sie einfach folgenden Befehl aus:

```
update-alternatives --remove x-window-manager /usr/local/bin/wmaker-cvs
```


Kapitel 12

Unterstützung für Debian GNU/Linux erhalten

12.1 Welche andere Dokumentation gibt es auf einem und für ein Debian-System?

- Installationsanleitung für die aktuelle Version: siehe <http://www.debian.org/releases/stable/installmanual>.
- Die Debian-Referenz behandelt viele Aspekte der Systemadministration durch Shell-Befehlsbeispiele. Grundanleitungen, Tipps und weitere Informationen für viele verschiedene Themen stehen zur Verfügung, angefangen bei der Systemadministration bis hin zur Programmierung.

Sie finden sie im `debian-reference`-Paket oder unter <http://www.debian.org/doc/user-manuals#quick-reference>.

- Das Policy-Handbuch beinhaltet die Richtlinien für die Distribution, das heißt Struktur und Inhalt des Debian-Archivs, verschiedene Design-Entscheidungen bezüglich des Betriebssystems usw. Es beinhaltet auch die technischen Anforderungen, die jedes Paket erfüllen muss, um in die Distribution aufgenommen zu werden, sowie technische Aspekte der Binär- und Quellpakete für Debian.

Es ist im `debian-policy`-Paket oder unter <http://www.debian.org/doc/devel-manuals#policy> zu finden.

- Dokumentation entwickelt vom Debian-Dokumentationsprojekt. Diese ist auf <http://www.debian.org/doc/> verfügbar und beinhaltet Benutzer-, Administrations- und Sicherheitshandbücher für das Debian GNU/Linux-Betriebssystem.
- Dokumentation zu installierten Debian-Paketen: Die meisten Pakete enthalten Dateien, die nach `/usr/share/doc/PAKETNAME` entpackt werden.
- Dokumentation zu Linux-Projekten: Das Debian-Paket `doc-linux` installiert alle aktuellen Versionen der HOWTOs und mini-HOWTOs vom Linux Documentation Project (<http://www.tldp.org/>).
- Handbuchseiten (`manpages`) im Unix-Stil: Zu den meisten Befehlen gibt es Handbuchseiten, die im gleichen Stil geschrieben sind wie die originalen »man«-Dateien für Unix. Um zum Beispiel die Handbuchseite für den Befehl »ls« aufzurufen, führen Sie `man ls` aus. Mittels `man man` bekommen Sie mehr Informationen dazu, wie Sie Handbuchseiten finden und lesen können.

Neue Debian-Benutzer sollten beachten, dass die Handbuchseiten vieler Systembefehle nur verfügbar sind, wenn folgende Pakete installiert werden:

- `man-db`, welches das `man`-Programm an sich und andere Programme beinhaltet, um die Handbuchseiten zu bearbeiten.
- `manpages`, welches die System-Handbuchseiten selbst beinhaltet. (Vergleichen Sie dazu auch 'Wie unterstützt Debian andere Sprachen als Englisch?' auf Seite 19).
- »info«-Seiten im GNU-Stil: Die Benutzerdokumentation für viele Befehle, besonders für GNU-Werkzeuge, ist nicht als Handbuchseite verfügbar, sondern als »info«-Datei, die mit dem GNU-Programm `info`, innerhalb von GNU Emacs mit `M-x info` oder mit einem anderen Info-Betrachter gelesen werden können.

Der große Vorteil gegenüber den ursprünglichen Handbuchseiten besteht darin, dass es ein Hypertext-System ist. Es wird jedoch *nicht* das WWW benötigt, `info` kann von einer einfachen Textkonsole gestartet werden. Es wurde von Richard Stallman entworfen und ging dem WWW voraus.

Die Softwarepakete »dwww«, »dhelp« und »doccentral« erfassen eine Menge Dokumentation und machen sie über einen Browser zugänglich. Auch hinter »yelp« steht die Idee, in verschiedenen Formaten vorliegende Dokumentation einem zentralen Programm zuzuführen.

12.2 Wo und wie kann man sich online über Debian austauschen?

Ja. Die Unterstützung, die Debian seinen Benutzern bietet, erfolgt hauptsächlich über E-Mail. Dazu hier einige Einzelheiten und Anlaufstellen. Weitere listet Debians Unterstützungs-Webseite (<http://www.debian.org/support>) auf.

12.2.1 Mailinglisten

Es gibt eine Reihe von Debian betreffenden Mailinglisten (<http://www.debian.org/MailingLists/>).

In einem System, auf dem das `doc-debian`-Paket installiert ist, finden Sie eine vollständige Auflistung der Mailinglisten in `/usr/share/doc/debian/mailling-lists.txt`.

Debian-Mailinglisten sind nach `debian-Listenbetreff` benannt. Beispiele hierfür sind `debian-announce` (für Ankündigungen), `debian-user` (für Benutzer) oder `debian-news` (für Neuigkeiten). Um eine Liste wie `debian-Listenbetreff` zu abonnieren, senden Sie eine E-Mail an `debian-Listenbetreff-request@lists.debian.org` mit dem Wort »subscribe« (anmelden) im Betreff. Vergessen Sie zum An- oder Abmelden das `-request` in der E-Mail-Adresse nicht, sonst geht die E-Mail an die Liste selbst, was entweder peinlich oder ärgerlich sein könnte, je nach Sichtweise.

Sie können die Mailingliste auch abonnieren, indem Sie das Web-Formular (<http://www.debian.org/MailingLists/subscribe>) benutzen. Auch zum Abmelden gibt es ein Formular (<http://www.debian.org/MailingLists/unsubscribe>).

Bei Problemen kann der Listenverwalter unter der E-Mail-Adresse `<listmaster@lists.debian.org>` kontaktiert werden.

Die meisten Mailinglisten sind öffentliche Foren. Alle an diese Listen geschickten Mails landen auch im öffentlichen Archiv, worin jeder - selbst ohne Anmeldung - stöbern oder systematisch suchen kann. Senden Sie keinesfalls vertrauliches oder unter dem Urheberrecht Dritter stehendes Material an diese Listen. Darunter fallen auch E-Mail-Adressen. Besonders zu beachten ist die Tatsache, dass Spammer dafür bekannt sind, E-Mail-Adressen zu missbrauchen, die in unseren Mailinglisten veröffentlicht werden. Weitere Informationen sind in den Datenschutzrichtlinien für Mailinglisten (<http://www.debian.org/MailingLists/index#disclaimer>) zu finden.

Archive der Debian-Mailinglisten sind im WWW unter <http://lists.debian.org/> verfügbar.

Was sind die Verhaltensregeln für die Mailinglisten?

Beim Benutzen der Debian-Mailinglisten sind folgende Regeln zu beachten:

- Spam unerwünscht. Weitere Informationen unter Verfahrensweise mit Werbung auf Debians Mailinglisten (<http://www.debian.org/MailingLists/#ads>).
- Nicht beleidigend werden, das ist unhöflich. Die Entwickler von Debian sind Freiwillige, die ihre Zeit, Energie und ihr Geld dafür einsetzen, das Debian-Projekt voranzubringen.
- Keine schmutzige Sprache verwenden; außerdem erhalten einige Leute diese Listen über Funk, wo Fluchen illegal ist.
- Die richtige Liste wählen. *Nie* Anfragen zur An- und Abmeldung an die Mailingliste selbst schicken.¹
- Informationen darüber, wie Fehler gemeldet werden, sind im Abschnitt 'Wie melde ich einen Fehler in Debian?' auf der nächsten Seite zu finden.

12.2.2 Web-Foren

`debianHELP` (<http://debianhelp.org/>) und `Debian User Forums` (<http://forums.debian.net/>) sind Web-Foren, wo Sie Fragen über Debian stellen können; sie werden von anderen Benutzern beantwortet. (Diese Foren sind nicht offizieller Teil des Debian-Projekts.)

¹Die `debian-Listenbetreff-request@lists.debian.org`-Adresse ist dafür zu benutzen.

12.2.3 Wiki

Lösungen für weitverbreitete Probleme, Howtos, Anleitungen, Tipps und weitere Dokumentation finden Sie im Debian Wiki (<http://wiki.debian.org/>), das laufenden Änderungen (auch durch Benutzer selbst) unterworfen ist.

12.2.4 Betreuer

Benutzer können ihre Fragen per E-Mail an die jeweiligen Paketbetreuer schicken. Um den Verantwortlichen des Pakets *xyz* zu erreichen, senden Sie eine E-Mail an xyz@packages.debian.org.

12.2.5 Usenet-Nachrichtengruppen (Newsgroups)

Nicht auf Debian bezogene Fragen sollten auf einer der Linux USENET-Gruppen mit Namen `comp.os.linux.*` oder `linux.*` gestellt werden. Dort gibt es diverse Listen von Linux Usenet-Nachrichtengruppen. Hinzu kommen weitere Anlaufstellen im WWW, z.B. auf den Seiten von Linux Online (<http://www.linux.org/docs/usenet.html>) und LinuxJournal (<http://www.linuxjournal.com/helpdesk.php>).

12.3 Gibt es einen schnellen Weg, um nach Informationen über Debian GNU/Linux zu suchen?

Es gibt eine Vielfalt von Suchmaschinen, die Dokumentation mit Bezug zu Debian liefern:

- Die Suche auf der Debian-Website (<http://search.debian.org/>).
- Google Groups (<http://groups.google.com/>): eine Suchmaschine für Newsgroups.
Um zum Beispiel herauszufinden, welche Erfahrungen Leute bei der Suche nach Treibern für NVIDIA-Grafikkarten unter Debian gemacht haben, können Sie nach `NVIDIA Linux driver` suchen. So werden alle Einträge angezeigt, die diese Zeichenfolgen enthalten, das heißt jene, in denen über dieses Thema diskutiert wurde. Wenn Debian dieser Suchanfrage hinzugefügt wird, werden auch die Einträge angezeigt, die Debian-spezifisch sind.
- Jede andere Suchmaschine, wie z.B. DuckDuckGo (<https://duckduckgo.com/>) oder Google (<http://www.google.com/>), solange Sie die richtigen Suchbegriffe eingeben.
Zum Beispiel können Sie mit der Suchanfrage »evince« eine viel genauere Erklärung über dieses Paket finden, als im kurzen Textfeld in seiner control-Datei steht.

12.4 Gibt es Protokolle von bekannten Fehlern?

Sowohl offene wie auch bereits erledigte Fehlerberichte sind frei zugänglich. Dies ist ein Versprechen, wie auch im Debian-Gesellschaftsvertrag (http://www.debian.org/social_contract) zu lesen ist: »Wir werden keine Probleme verheimlichen«.

Debian GNU/Linux hat ein System zur Meldung und Statuskontrolle von Fehlern eingerichtet, das »Bug Tracking System«, abgekürzt BTS. Dieses führt von Entwicklern und Benutzern übermittelte Fehler und dazugehörige Details einer Datenbank zu, wo sie auch eine Nummer erhalten. Sobald das Problem gelöst ist, wird der Fehlerbericht geschlossen.

Kopien dieser Informationen sind erhältlich unter <http://www.debian.org/Bugs/>.

Ein E-Mail-Server ermöglicht per E-Mail Zugang zur Fehlerdatenbank. Um Anweisungen für dessen Nutzung zu erhalten, senden Sie eine E-Mail an request@bugs.debian.org mit »help« im Textteil der Mail.

12.5 Wie melde ich einen Fehler in Debian?

Wenn Sie einen Fehler in Debian gefunden haben, sollten Sie die Anweisungen dazu lesen, wie ein Fehler in Debian gemeldet wird. Diese Anweisungen sind auf verschiedene Arten erhältlich:

- Über das WWW. Eine Kopie der Anweisungen ist unter <http://www.debian.org/Bugs/Reporting> zu finden.

- Auf jedem Debian-System, auf dem das `doc-debian`-Paket installiert ist, finden sie die Anweisungen in der Datei `/usr/share/doc/debian/bug-reporting.txt`.
- Durch anonymes FTP. Debian-Spiegelserver enthalten die Anweisungen in der Datei `doc/bug-reporting.txt`.

Das Paket `reportbug` kann benutzt werden, es führt Sie durch den Prozess und schickt die Nachricht an die richtige Adresse; außerdem werden automatisch einige zusätzliche Details über Ihr System beifügt. Auch zeigt es eine Liste der bereits für das Paket gemeldeten Fehler an; so können für den Fall, dass der Fehler bereits gemeldet wurde, dem existierenden Fehlerbericht zusätzliche Informationen hinzugefügt werden.

Es wird eine automatische Empfangsbestätigung zum Fehlerbericht an Sie verschickt. Der Bericht bekommt automatisch eine Nummer zur Fehlerverfolgung, wird der Fehlerdatenbank hinzugefügt und an die `debian-bugs-dist`-Mailingliste weitergeleitet.

Kapitel 13

Zum Debian-Projekt beitragen

Sie können verschiedene Dinge spenden (<http://www.debian.org/donations>), um Debian zu helfen: etwa Zeit, um neue Pakete zu entwickeln, vorhandene zu betreuen oder Benutzer zu unterstützen; Ressourcen, um die FTP- und WWW-Archive zu spiegeln, oder Geld, um neue Testsysteme oder Hardware für die Archive kaufen zu können. Lesen Sie auch [Wie können Sie Debian helfen?](http://www.debian.org/intro/help) (<http://www.debian.org/intro/help>).

13.1 Wie kann ich Debian-Mitglied/Debian-Entwickler werden?

Die Entwicklung von Debian ist für jeden zugänglich. Es werden immer neue Benutzer mit den entsprechenden Fähigkeiten und/oder dem Willen zu lernen gesucht. So können Sie die Pflege eines existierenden, »verwaisten« Paketes übernehmen, dessen vorheriger Betreuer sich von der Paketpflege zurückgezogen hat. Außerdem gilt es, neue Pakete zu entwickeln, Dokumentation oder Übersetzungen zu verfassen, die Debian-Website zu betreuen oder Benutzer bei Problemen zu unterstützen.

Wie man ein Debian-Mitglied wird, ist auf der Debian-Website in der Ecke für neue Mitglieder (<http://www.debian.org/devel/join/newmaint>) beschrieben.

13.2 Wie kann ich Ressourcen zum Debian-Projekt beisteuern?

Da das Projekt versucht, eine beträchtliche Menge von Software weltweit rasch und einfach zugänglich zu machen, werden Spiegel-Server benötigt. Es ist wünschenswert, aber nicht unbedingt notwendig, alle Debian-Archive zu spiegeln. Bitte besuchen Sie die Seite über die Größe der Spiegel-Server (<http://www.debian.org/mirror/size>), um Informationen über den benötigten Speicherplatz zu erhalten.

Skripte erledigen den Vorgang des Spiegelns weitgehend automatisch. Vereinzelt Pannen oder Systemänderungen können allerdings menschliches Eingreifen erforderlich machen.

Wenn Sie über eine Breitband-Verbindung zum Internet verfügen, die Ressourcen haben, um die Distribution oder einen Teil davon zu spiegeln, die Zeit für die regelmäßige Wartung des Systems entweder selbst aufbringen wollen oder dieses jemandem antragen können, dann melden Sie sich bitte unter `<debian-admin@lists.debian.org>`.

13.3 Wie kann ich das Debian-Projekt finanziell unterstützen?

Durch Spenden von Sponsoren gelangt das Debian-Projekt an Rechner und andere Hardware, kann Konferenzen und Arbeitstreffen für Entwickler veranstalten und weitere Vorhaben umsetzen. Weitere Informationen finden Sie auf Debians Spenden-Website (<http://www.debian.org/donations>). Dort sind auch die unterschiedlichen Methoden aufgelistet, mittels derer Sie spenden können.

Es können individuelle Spenden an zwei für das Debian-Projekt sehr wichtige Organisationen eingereicht werden. Die Hauptorganisation ist »Software in the Public Interest« in den Vereinigten Staaten, aber es gibt noch weitere.

13.3.1 Software in the Public Interest

Software in the Public Interest (SPI) ist eine nach IRS 501(c)(3) gemeinnützige Organisation. Die Aufgabe der Organisation ist es, Freie Software zu entwickeln und zu verteilen.

Sie ermutigt Programmierer, die GNU General Public License oder eine andere Lizenz, welche die freie Weiterverbreitung und Nutzung von Software erlaubt, zu verwenden. Herstellern von Hardware legt sie nahe, zu Ihren Geräten Dokumentation bereitzustellen, um die Entwicklung von Treibern zu ermöglichen.

SPI ist steuerrechtlich als begünstigt anerkannte Dachorganisation für viele freie und Open-Source-Projekte. Das Debian-Projekt war seit der Gründung von SPI ein assoziiertes Mitglied.

Sie können SPI über <http://www.spi-inc.org/> erreichen.

13.3.2 Weitere Organisationen

There are a number of organizations created in different countries that hold assets in trust for Debian. The donations page (<http://www.debian.org/donations>) lists the trusted organizations individuals can donate to. At the time of this writing there are three of them: Verein zur Förderung Freier Informationen & Software (<http://www.ffis.de/>) (in Germany), the Debian France Association (<https://france.debian.net/>) (in France), and debian.ch (<http://debian.ch/>) (Switzerland and the Principality of Liechtenstein). Additional affiliate organizations in other countries are listed in Organizations (<https://wiki.debian.org/Teams/Auditor/Organizations>) page in the Debian Wiki.

Kapitel 14

Debian GNU/Linux in einem kommerziellen Produkt verwenden

14.1 Darf ich Debian-CDs erstellen und verkaufen?

Ja, das dürfen Sie. Sobald die Beta-Tests beendet sind, können Sie Ihre CD erstellen und alles, was wir *freigegeben* haben, ohne weitere Zustimmung unsererseits veröffentlichen. Sie müssen nichts an uns zahlen. Natürlich müssen alle CD-Hersteller die Lizenzen der Programme in Debian anerkennen. Viele der Programme sind z.B. unter der GPL-Lizenz veröffentlicht, welche verlangt, dass die Quelltexte veröffentlicht werden.

CD-Hersteller, die Geld, Software und Zeit zum Debian-Projekt beitragen, werden von uns in einer Liste veröffentlicht. Außerdem ermuntern wir die Benutzer dazu, bei Herstellern zu kaufen, die auch an Debian spenden. Man sieht also, es ist eine gute Werbung, zu spenden.

14.2 Kann Debian mit unfreier Software (non-free) paketiert werden?

Ja. Während alle Haupt-Komponenten von Debian Freie Software sind, stellen wir ein `non-free`-Verzeichnis für Programme bereit, die nicht frei verbreitet werden dürfen.

Programme, die wir in diesem Verzeichnis ablegen, *können* (unter Berücksichtigung der entsprechenden Lizenzbedingungen oder mit Zustimmung der jeweiligen Autoren) von CD-Herstellern verbreitet werden. Genauso ist es CD-Herstellern gestattet, unfreie Software aus anderen Quellen auf derselben CD in Umlauf zu bringen. Das Verteilen von unfreier und kommerzieller Software auf ein- und derselben CD ist nichts Neues und wird von vielen Herstellern praktiziert. Selbstverständlich legen wir Software-Entwicklern nahe, die von ihnen geschriebene Software als freie Software zu veröffentlichen.

14.3 Ich erstelle eine spezielle Linux-Distribution für einen speziellen Anwendungsbereich. Kann ich Debian GNU/Linux als Linux-Grundsystem verwenden und meine eigenen Programme hinzufügen?

Ja, das ist möglich. Von Debian abgeleitete Distributionen werden zum Einen in enger Zusammenarbeit mit dem Debian-Projekt selbst und zum Anderen von außenstehenden Beteiligten erstellt. Dazu kann das Debian Pure Blends (<https://www.debian.org/blends/>)-Rahmenwerk verwendet werden, um mit Debian zusammen zu arbeiten. DebianEdu/SkoLeLinux (<https://wiki.debian.org/DebianEdu/>) ist z.B. ein solches Projekt.

Es gibt bereits viele angepasste Debian-Distributionen auf dem Markt wie grml, LMDE (Linux Mint Debian Edition), Knoppix und Ubuntu, die alle eine andere Zielsetzung als das Original-Debian GNU/Linux-Projekt haben und trotzdem die meisten unserer Komponenten in ihren Produkten verwenden.

Debian weist einen Mechanismus auf, der es Entwicklern und Systemadministratoren erlaubt, lokale Versionen bestimmter Dateien so zu installieren, dass sie bei einer Systemaktualisierung nicht überschrieben werden. Näheres dazu finden Sie in 'Wie kann ich eine durch ein Paket installierte Datei außer Kraft setzen, so dass stattdessen eine andere Version verwendet werden kann?' auf Seite 49.

14.4 Kann ich mein kommerzielles Programm in einem Debian-»Paket« ablegen, damit es auf jedem Debian-System mühelos installiert werden kann?

Das ist möglich. Das Paketwerkzeug ist Freie Software: Die Pakete können freie oder unfreie Software sein, es kann sie alle installieren.

Kapitel 15

Zu erwartende Änderungen in der nächsten Hauptversion von Debian

Mit jeder neuen Veröffentlichung versucht das Debian-Projekt, den Fokus auf bestimmte Themen zu legen. Diese sind bekannt als »Release Goals« und werden unter <https://wiki.debian.org/ReleaseGoals/> näher beschrieben. Bitte beachten Sie, dass die folgenden Abschnitte unter Umständen nicht ganz aktuell sind; lesen Sie daher die Wiki-Seite, wenn Sie mehr Informationen oder einen aktuellen Status dieser »Release-Goals« benötigen.

15.1 Härten des Systems

Es ist ein erklärtes Ziel des Debian-Projekts, sicherzustellen, dass jedes installierte System gehärtet und sicher ist gegen Angriffe. Es gibt mehrere Möglichkeiten, dies zu erreichen, unter anderem:

- Die Sicherheit innerhalb der Programme erhöhen, indem die Security Hardening Build Flags (<https://wiki.debian.org/ReleaseGoals/SecurityHardeningBuildFlags>) verwendet werden; durch diese speziellen Optionen für den Bau der Pakete werden verschiedene Schutzmechanismen gegen bekannte Sicherheitslücken aktiviert.
- Die Standard-Systemkonfiguration verbessern, um sie weniger verwundbar gegen Angriffe (sowohl lokale wie auch solche von fern) zu machen.
- Aktivieren der Sicherheitsfunktionen, die in neuen Kernel-Versionen enthalten sind.

All dies ist ein ständig fortlaufender Prozess. Beim ersten Punkt wird versucht, mit verschiedenen Build-Optionen zum Härten des Systems (den Security Hardening Build Flags) bekannte Angriffe wie z.B. mittels Pufferüberlauf, vorher-sagbaren Adressbereichen im Arbeitsspeicher usw. zu verhindern. Ziel ist, zumindest alle Pakete abzudecken, die Teil der Basisinstallation sind sowie all diejenigen, die seit 2006 eine Korrektur in Zusammenhang mit einer Sicherheitsankündigung erhalten haben. Seit dem Start der Initiative sind bereits rund 400 Pakete entsprechend modifiziert worden (Stand der Information zum Zeitpunkt, als dies geschrieben wurde). All diese Korrekturen sind in der Fehlerdatenbank festgehalten (<https://bugs.debian.org/cgi-bin/pkgreport.cgi?tag=goal-hardening;users=hardening-discuss@lists.alioth.debian.org>).

15.2 Erweiterte Unterstützung für nicht Englisch sprechende Benutzer

Die technischen Anforderungen, die damit einhergehen, dass nicht Englisch sprechende Benutzer das System bedienen und verwenden können, werden von Debian bereits weitgehend erfüllt, siehe 'Wie unterstützt Debian andere Sprachen als Englisch?' auf Seite 19.

Wir hoffen, weitere Personen zu finden, die uns helfen, mehr Sprachen zu unterstützen und Programme sowie Dokumentation in diese Sprachen zu übersetzen. Viele Programme und Debian-spezifische Dokumentation unterstützen bereits die Internationalisierung, so dass wir jetzt Übersetzer für die Programmtexte und Meldungen benötigen. Allerdings sind bei einigen Programmen auch weiterhin die Voraussetzungen zur Internationalisierung nicht erfüllt.

Das GNU Translation Project <ftp://ftp.gnu.org/pub/gnu/ABOUT-NLS> arbeitet an der Internationalisierung von GNU-Programmen. Verschiedene Projekte wie die Desktop-Umgebungen GNOME oder KDE haben eigene Übersetzer-Teams. Ziel von Debian ist nicht, die Arbeit dieser Projekte zu ersetzen oder zu duplizieren; im Gegenteil: Debian profitiert

von der Arbeit dieser Übersetzer. Allerdings gibt es außerhalb des Tätigkeitfeldes dieser Projekte noch viele Programme, weswegen sie innerhalb von Debian übersetzt werden.

Frühere Debian-Versionen hatten folgende Veröffentlichungsziele:

- I18n-Unterstützung für alle Pakete, die debconf verwenden: Für Pakete, die das Debian-Konfigurations-Management verwenden, wird vorausgesetzt, dass sie vorbereitet sind für die Übersetzung aller Dialoge, die dem Benutzer während der Konfiguration angezeigt werden.
- I18n-Unterstützung für Paketbeschreibungen: Erweitern der Paketmanagement-Programme, um übersetzte Paketbeschreibungen nutzen zu können.
- Verwendung der UTF-8-Codierung für die Dateien debian/changelog und debian/control. So können beispielsweise in ersterer Datei, in der Softwareänderungen rückzuverfolgen sind, Projektbeteiligte aus asiatischen Ländern ihre Namen ohne Komplikationen eintragen.
- Internationalisierung des Debian-Installers, was für gewisse Sprachen die Verwendung einer grafischen Oberfläche erfordert.

15.3 Verbesserungen im Debian Installer

Eine Menge Arbeit ist in das Debian-Installationsprogramm gesteckt worden, was zu einer Reihe größerer Verbesserungen geführt hat. Hier sollen zwei davon aufgeführt werden:

Die Möglichkeit einer direkt vom laufenden Microsoft Windows angestossenen Installation, wodurch es sich erübrigt, BIOS-Einstellungen zu verändern. Nach dem Einlegen einer CD-ROM, DVD-ROM oder dem Einstecken eines USB-Sticks wird automatisch ein Programm ausgeführt, das eine Schritt-für-Schritt-Anleitung zum Start des Debian Installers bietet.

15.4 Weitere (Rechner-)Architekturen

Vervollständigung von Debian für andere Architekturen. Hinweis: Selbst wenn manche Architekturen in einzelnen Debian-Veröffentlichungen nicht enthalten sind, besteht unter Umständen trotzdem die Möglichkeit, Debian auf dieser Architektur zu installieren bzw. zu aktualisieren, indem sid (Unstable) benutzt wird.

15.5 Weitere Kernel

Zusätzlich zu Debian GNU/Hurd wird Debian auch auf BSD-Kernel portiert, insbesondere auf FreeBSD (<http://www.debian.org/ports/kfreebsd-gnu/>). Dieser Port läuft sowohl auf AMD64 (»kfreebsd-amd64«) als auch auf traditionellen Intel-Prozessoren (»kfreebsd-i386«).

Kapitel 16

Allgemeine Informationen über die FAQ

16.1 Autoren

Die erste Version dieser FAQ wurde erstellt und betreut von J.H.M. Dassen (Ray) und Chuck Stickelman. Autoren der überarbeiteten Debian GNU/Linux-FAQ sind Susan G. Kleinmann und Sven Rudolph. Danach wurde die FAQ von Santiago Vila und später von Josip Rodin betreut. Der derzeitige Betreuer ist Javier Fernandez-Sanguino.

Teile der Informationen stammen aus:

- der Debian-1.1-Veröffentlichungsankündigung, von Bruce Perens (<http://www.perens.com/>),
- der Linux-FAQ, von Ian Jackson (<http://www.chiark.greenend.org.uk/~ijackson/>),
- den Archiven der Debian-Mailinglisten (<http://lists.debian.org/>),
- dem dpkg-Programmiererhandbuch und dem Debian-Policy-Handbuch (siehe 'Welche andere Dokumentation gibt es auf einem und für ein Debian-System?' auf Seite 53),
- sowie von vielen Entwickler, Freiwilligen und Beta-Testern,
- aus den löchrigen Gehirnen der Autoren :-)
- und aus dem Ratgeber Choosing a Debian distribution FAQ (http://KamarajuKusumanchi.github.io/choosing_debian_distribution/choosing_debian_distribution.html), den Kamaraju Kusumanchi gnädigerweise unter der GPL freigegeben hat, so dass er als neues Kapitel in diese FAQ eingefügt werden konnte (siehe 'Eine Debian-Distribution auswählen' auf Seite 7).

Die Autoren möchten sich bei allen bedanken, die dieses Dokument möglich gemacht haben. Die deutsche Übersetzung stammt von Christian Britz, Nico Golde, Claas Felix Beyersdorf, Andre Appel, Franziska Schröder, Jonas E. Huber, Martin Horoba, Helge Kreuzmann, Jens Seidel und Benjamin Eckenfels. Sie wurde in 2015 aktualisiert von Holger Wansing, mit tatkräftiger Unterstützung durch Markus Hiereth. Bitte senden Sie Fehler in der Übersetzung an <debian-110n-german@lists.debian.org>.

Jegliche Haftung wird ausgeschlossen. Alle Markenzeichen sind das Eigentum ihrer entsprechenden Eigentümer.

16.2 Feedback

Kommentare und Ergänzungen zu diesem Dokument sind immer willkommen. Bitte senden Sie eine E-Mail an <doc-debian@packages.debian.org> oder schreiben Sie einen wishlist-Fehlerbericht für das Paket `debian-faq` (<http://bugs.debian.org/debian-faq>).

16.3 Verfügbarkeit

Die neueste Version dieses Dokuments finden Sie auf den Debian WWW-Seiten unter <http://www.debian.org/doc/FAQ/>.

Außerdem ist die FAQ zum Download im Text-, HTML-, PostScript- und PDF-Format unter <http://www.debian.org/doc/user-manuals#faq> verfügbar. Dort sind auch einige Übersetzungen zu finden.

Sie können dieses Dokument über das Debian-Paket `debian-faq` installieren. Übersetzungen sind erhältlich über `debian-faq-de`, `debian-faq-fr` und weitere Pakete.

Die SGML-Dateien, aus denen dieses Dokument generiert wurde, sind auch im Quellpaket `debian-faq` oder im SVN verfügbar: `svn://svn.debian.org/svn/ddp/manuals/trunk/debian-faq` oder <http://svn.debian.org/viewsvn/ddp/manuals/trunk/debian-faq/>.

16.4 Dokumentenformat

Dieses Dokument wurde mit der DebianDoc-SGML-DTD geschrieben (umgeschrieben vom LinuxDoc-SGML). Das DebianDoc-SGML-System ermöglicht es, Dateien in vielen verschiedenen Formaten aus einer einzigen Quelldatei zu erstellen; z.B. kann dieses Dokument als HTML, Text, TeX-DVI, PostScript, PDF oder GNU-info angezeigt werden.

Konvertierungswerkzeuge für DebianDoc-SGML sind im Debian-Paket `debiandoc-sgml` verfügbar.