Debian-Referenz

Osamu Aoki <osamu\#at\#debian.org>
Übersetzer: Jens Seidel <tux-master\#at\#web.de>
'Autoren' auf Seite 243

CVS, Don 18. Jan 2007, 11:52:59 UTC

Zusammenfassung

Diese Debian-Referenz (`http://qref.sourceforge.net/`) wurde geschrieben, um einen groben Überblick über das Debian-System in Form einer **Referenz für die Zeit nach der Installation** zu bieten. Sie behandelt viele Aspekte der Systemadministration mittels **Shell-Kommando**-Beispielen. Grundlegende Tutorials, Tipps und andere Informationen werden dargestellt für Themen inklusive der fundamentalen Konzepte des Debian-Systems, System-Installations-Hinweise, Debian-Paketverwaltung, den Linux-Kernel unter Debian, System-Tuning, Aufbau eines Gateway, Texteditoren, CVS, Programmierung und GnuPG für **Nicht-Entwickler**.

Inhaltsverzeichnis

Kapitel 1

Vorwort

Diese Debian-Referenz (`http://qref.sourceforge.net/`) wurde geschrieben, um einen groben Überblick über das Debian-System in Form einer Referenz für die Zeit nach der Installation zu bieten. Es spricht die Leser an, welche bereit sind, Shell-Skripte zu lesen. Ich erwarte, dass der Leser bereits hinreichende Erfahrungen mit Unix-artigen Systemen vor dem Lesen dieses Dokuments gesammelt hat.

Ich traf die bewusste Entscheidung **nicht** alles im Detail zu erklären, wenn es in einer Handbuchseite, einer info-Seite oder einem HOWTO-Dokument gefunden werden kann. Anstatt alles ausführlich zu erklären, habe ich mehr Wert auf direkte praktische Informationen durch Angabe exakter Kommandosequenzen im Text oder Beispielskripte unter `http://www.debian.org/doc/manuals/debian-reference/examples/` gelegt. Sie müssen den Inhalt der Beispiele verstanden haben, bevor Sie sie ausführen. Möglicherweise müssen Sie die Kommandosequenzen auch leicht abändern.

Viele der aufgeführten Informationen entstammen meinem Gedächtnis oder Verweisen auf die ultimativen Referenzen, welche unter ‘Referenzen’ auf Seite 235 aufgelistet sind.

Dieses Dokument war ursprünglich eine “Schnellreferenz” wuchs jedoch schnell. Dennoch ist **Halte es kurz und einfach** mein Leitmotiv.

Für Hilfe im Falle von Notfall-Wartung wird auf ‘Debian-Überlebensbefehle’ auf Seite 70 verwiesen.

1.1 Das offizielle Dokument

Das neueste offizielle Dokument ist im Debian-Archiv im Paket `debian-reference-de` und ebenso unter `http://www.debian.org/doc/manuals/debian-reference/` verfügbar.

Die aktuellste Entwicklerversion ist `http://qref.sourceforge.net/Debian/`. Das Projekt wird auf `http://qref.sourceforge.net/` gehostet, und dieses Dokument ist da zum Herunterladen als einfacher Text und im HTML-, PDF-, SGML- sowie PostScript-Format verfügbar.

1.2 Konventionen in diesem Dokument

Diese Debian-Referenz enthält Informationen in Form von kurzen `bash`-Shell-Kommandos. Die folgenden Notationen werden verwendet:

```
# Kommando im root-account
$ Kommando im Nutzer-account
... Beschreibung der Auswirkungen
```

Diese Shell-Kommandobeispiele verwenden `PS2=" "`. Vergleiche 'Bash – interaktive **GNU**-Standard-Shell' auf Seite 209 für weitere Informationen zur `bash`.

Verweise auf:
- eine UNIX-artige **Handbuchseite** erfolgen in der Form: `bash(1)`.
- eine GNU-**TEXINFO-Seite** erfolgen in der Form: `info libc`.
- ein **Buch** erfolgen in der Form: *Die Programmiersprache C*.
- eine **URL** erfolgen in der Form: `http://www.debian.org/doc/manuals/debian-reference/`.
- eine **Datei** im System erfolgen in der Form: `/usr/share/doc/Debian/reference/`.

Die folgenden Abkürzungen werden verwendet:
- **LDP**: Linux-Dokumentations-Projekt (`http://www.tldp.org/`)
- **DDP**: Debian-Dokumentations-Projekt (`http://www.debian.org/doc/`)

Andere Abkürzungen werden im Text definiert, bevor sie verwendet werden.

In diesem Dokument werden für LDP-Dokumente nur URL-Verweise angegeben. Dennoch sind LDP-Dokumente für Debian paketiert; wenn die Pakete installiert wurden, sind die Dokumente in `/usr/share/doc/HOWTO/` zu finden.

Vergleiche 'Referenzen' auf Seite 235.

1.3 Beispielskripte

Beispielskripte (`http://www.debian.org/doc/manuals/debian-reference/examples/`), die diesem Dokument im `debian-reference-de`-Paket beiliegen, sind unter `/usr/share/doc/Debian/reference/examples/` verfügbar. Der führende Punkt "." in Dateinamen bei versteckten Dateien wird durch einen Unterstrich "_" ersetzt. Eine zusätzliche Erweiterung zum Dateinamen wird verwendet, wenn es mehrere Alternativen gibt.

1.4 Grundlegende Einstellungen

Ist das System mit dem absoluten Minimum an Paketen installiert und wollen Sie den größten Nutzen aus diesem Dokument ziehen, so ist es ratsam die folgenden Kommandos zu starten, um einige andere Pakete mit nützlichen Schlüsseldokumenten zu installieren:

```
# aptitude install mc less ssh vim kernel-package \
                   manpages-dev doc-debian doc-linux-text \
                   debian-policy developers-reference maint-guide \
                   apt-howto harden-doc debian-reference \
                   libpam-doc glibc-doc samba-doc exim4-doc-html
```

1.5 Grundlagen der Debian-Distributionen

Debian betreut gleichzeitig drei verschiedene Distributionen. Diese sind:

- `stable` – Am nützlichsten für einen produktiven Server, da nur Sicherheitsaktualisierungen eingespielt werden. Vergleiche 'Die `stable`-Distribution' auf Seite 6.
- `testing` – Die bevorzugte Distribution für eine Workstation, da sie aktuelle Ausgaben von Desktop-Software enthält, die schon ein wenig getestet wurden. Vergleiche 'Die `testing`-Distribution' auf Seite 7.
- `unstable` – Topaktuell. Die Wahl der Debian-Entwickler. Vergleiche 'Die `unstable`-Distribution' auf Seite 7.

Wenn Pakete in `unstable` keine veröffentlichungskritischen Fehler innerhalb der ersten Woche oder so haben, gelangen sie automatisch nach `testing`.

Debian-Distributionen haben auch Kodenamen wie in 'Debian-Distributions-Kodenamen' auf Seite 8 beschrieben. Bevor Woody im August 2002 veröffentlicht wurde, entsprachen den drei Distributionen Potato, Woody und Sid. Nachdem Woody veröffentlicht wurde, entsprachen sie den drei Distributionen Woody, Sarge und Sid. Wenn Sarge veröffentlicht wurde, werden die `stable`- und `unstable`-Distributionen Sarge und Sid sein; eine neue `testing`-Distribution wird dann erzeugt (zunächst als Kopie von `stable`) und ihr ein neuer Kodenamen zugeordnet.

Abonnieren Sie die Mailingliste `debian-devel-announce@lists.debian.org` (geringes Mail-Aufkommen) für wichtige Ankündigungen über Debian. Vergleiche 'Die Debian-Archive' auf Seite 5.

Möchte man Versionen von Paketen verwenden, die aktueller sind als die Versionen der Distribution die man nutzt, kann man entweder auf eine neuere Distribution, wie in 'Aktualisieren einer Distribution auf `stable`, `testing` oder `unstable`' auf Seite 59 beschrieben, aktualisieren, oder nur die ausgesuchten Pakete aktualisieren. Wenn das Paket nicht einfach aktualisiert werden kann, kann man es wie in 'Portierung eines Pakets auf die `stable`-Distribution' auf Seite 78 erläutert, zurückportieren.

Das Verwenden der `testing`-Distribution kann den Nebeneffekt haben, das Sicherheitsupdates nur langsam eingespielt werden. Solche Pakete werden nach `unstable` hochgeladen und gelangen nur mit einer Verzögerung nach `testing`.

Vermengen Sie verschiedene Distributionen, z.B. `testing` mit `stable` oder `unstable` mit `stable`, so werden eventuell Kernpakete von `testing` oder `unstable` wie `libc6` installiert, die fehlerhaft sein können. Sie wurden gewarnt.

Die Verwendung der `testing`- oder `unstable`-Distribution erhöht das Risiko, auf verschiedene gravierende Fehler zu stoßen. Dieses Risiko kann durch ein Multi-Boot-Schema gemanagt werden, bei dem eine stabilere Debian-Distribution genutzt wird. Alternativ kann auch ein netter Trick mit `chroot`, wie in '`chroot`' auf Seite 119 beschrieben, genutzt werden. Letzteres ermöglicht die Nutzung verschiedener Versionen von Debian simultan auf verschiedenen Konsolen.

Nach dem Erklären einiger Grundlagen der Debian-Distribution in 'Debian-Grundlagen' auf der nächsten Seite, werden einige wichtige Informationen für die Verwendung der aktuellsten Software, unter Ausnutzung der Vorteile der `testing`- und `unstable`-Distributionen von Debian, präsentiert. Die Ungeduldigen sollten sofort mit 'Debian-Überlebensbefehle' auf Seite 70 fortfahren. Glückliches upgraden!

Kapitel 2

Debian-Grundlagen

Dieses Kapitel liefert grundlegende Informationen über das Debian-System für Nicht-Programmierer. Für die ultimativen Informationen vergleiche:

- Debian Policy Manual
- Debian Developer's Reference
- Debian New Maintainers' Guide

aufgeführt unter 'Referenzen' auf Seite 235.

Wenn Sie nach nicht allzu ausführlichen "wie geht" Erklärungen suchen, wechseln Sie direkt nach 'Debian-Paketverwaltung' auf Seite 63 oder in andere relevante Kapitel.

Dieses Kapitel basiert auf Dokumenten aus der "Debian-FAQ", größtenteils umgeschrieben, um dem gewöhnlichen Debian-Systemadministrator den Einstieg zu erleichtern.

2.1 Die Debian-Archive

2.1.1 Verzeichnisstrukturen

Die Software, welche für Debian paketiert wurde, ist in einem der vielen Verzeichnisbäume auf jedem Debian-Mirror (`http://www.debian.org/mirror/`) durch FTP oder HTTP verfügbar.

Die folgenden Verzeichnisse können auf jedem Debian-Mirror unter dem `debian` Verzeichnis gefunden werden:

`dists/`: Dieses Verzeichnis enthält die "Distributionen" und ist für den Zugriff auf die aktuell verfügbaren Pakete in Debian ausgelegt. Einige alte Pakete, die `Contents-*.gz`-Dateien und die `Packages.gz`-Dateien sind immer noch hier zu finden.

`pool/`: Die neue Position aller Debian-Pakete.

`tools/`: DOS-Hilfsmittel zum Erzeugen von Bootdisketten, Partitionieren der Festplatte, Komprimieren/Dekomprimieren von Dateien und zum Booten von Linux.

doc/: Die grundlegenden Debian-Dokumentationen wie die FAQ's, Erläuterungen zum Fehler-Melde-System, usw.

indices/: Enthält die `Maintainers`-Datei (Liste aller Paketbetreuer) und die `override.*`-Dateien.

project/: Hauptsächlich Material, welches nur für Entwickler von Interesse ist, wie z.B.:

project/experimental/: Dieses Verzeichnis enthält Pakete und Hilfsmittel, welche noch entwickelt werden und sich noch im Alpha-Stadium befinden. Benutzer sollten keine Pakete von hier verwenden, da sie selbst für den Erfahrensten gefährlich und schädlich sind.

project/orphaned/: Pakete, welche von ihrem alten Betreuer aufgegeben wurden und aus der Distribution zurückgezogene Software.

2.1.2 Debian-Distributionen

Normalerweise befinden sich drei Debian-Distributionen im `dists`-Verzeichnis. Dies sind die `stable`-, die `testing`- und die `unstable`-Distribution. Manchmal gibt es auch eine `frozen`-Distribution. Jede Distribution ist ein symbolischer Link in das entsprechende Verzeichnis mit einem Kodenamen im `dists`-Verzeichnis.

2.1.3 Die `stable`-Distribution

Pakete der `stable`-Distribution, Debian Sarge (3.1r0), befinden sich im `stable`- (symbolischer Link zu `sarge/`) Verzeichnis:

- `stable/main/`: Dieses Verzeichnis enthält die Pakete, welche formal die aktuellste Ausgabe des Debian-Systems bilden.

  Diese Pakete entsprechen alle den Debian Free Software Leitlinien (DFSG) (`http://www.debian.org/social_contract#guidelines`) (auch verfügbar unter `/usr/share/doc/debian/social-contract.txt` installiert durch `debian-doc`) und sind alle frei verfügbar und verteilbar.

- `stable/non-free/`: Dieses Verzeichnis enthält Pakete, deren Weitergabe in gewisser Weise beschränkt ist. Genauere Informationen sind in den jeweiligen Copyright-Hinweisen zu finden.

  Zum Beispiel verbieten die Lizenzen einiger Pakete die kommerzielle Verteilung. Andere können weitergegeben werden, aber nur als Shareware und nicht als freie Software. Die Lizenzen all dieser Pakete müssen genau studiert und möglicherweise ausgehandelt werden, bevor die Pakete in irgendeiner Form (z.B. auf einer CD-ROM) weitergegeben werden.

- stable/contrib/: Dieses Verzeichnis enthält Pakete, welche DFSG-frei und **frei verteilbar** sind, aber irgendwie von einem Paket abhängen, das **nicht** frei verteilbar und somit nur im non-free Abschnitt zu finden ist.

Gegenwärtig befinden sich neue Pakete in Ergänzung zu obigen Verzeichnissen unterhalb des pool-Verzeichnisses ('Das pool-Verzeichnis' auf Seite 9).

Der aktuelle Status von Fehlern der stable-Distribution ist unter der Stable Problems (http://ftp-master.debian.org/testing/stable_probs.html)-Webseite aufgeführt.

2.1.4 Die testing-Distribution

Pakete der testing-Distribution, Debian Etch, befinden sich im testing- (symbolischer Link zu etch/) Verzeichnis, nachdem sie einige Zeit in unstable getestet wurden. Gegenwärtig befinden sich neue Pakete, im Gegensatz zu den obigen Positionen, unterhalb des pool-Verzeichnisses ('Das pool-Verzeichnis' auf Seite 9). Auch in testing/ gibt es die main-, contrib- und non-free-Unterverzeichnisse, diese entsprechen den Verzeichnissen in stable/.

Diese Pakete müssen auf allen Architekturen auf denen sie zur Verfügung stehen gleich aktuell sein und dürfen keine Abhängigkeiten aufweisen, welche sie nicht installierbar machen; sie müssen auch weniger veröffentlichungskritische Fehler haben, als die Versionen in unstable. Auf diese Art hoffen wir, dass testing fast immer zur Veröffentlichung bereit ist. Mehr Details zu den Test-Mechanismen sind unter http://www.debian.org/devel/testing verfügbar.

Der aktuellste Status der testing-Distribution ist unter folgenden Seiten aufgeführt:
- update Ausflüchte (http://ftp-master.debian.org/testing/update_excuses.html)
- testing Probleme (http://ftp-master.debian.org/testing/testing_probs.html)
- veröffentlichungskritische Fehler (http://bugs.debian.org/release-critical/)
- Basissystem Fehler (http://bugs.qa.debian.org/cgi-bin/base.cgi)
- Fehler in Standard und Task Paketen (http://bugs.qa.debian.org/cgi-bin/standard.cgi)
- andere Fehler und Bemerkungen zu Fehlerausmerzungs-Partys (http://bugs.qa.debian.org/)

2.1.5 Die unstable-Distribution

Pakete, welche zur unstable-Distribution mit dem Kodenamen "Sid" gehören, werden im unstable- (symbolischer Link zu sid/) Verzeichnis aufbewahrt, nachdem sie in das Debian-Archiv geladen wurden und verbleiben hier solange, bis sie nach testing/ verschoben werden. Neue Pakete befinden sich unterhalb des pool-Verzeichnis ('Das pool-Verzeichnis' auf der nächsten Seite). Es gibt auch `main`-, `contrib`- und `non-free`-Unterverzeichnisse in `unstable/`, welche dem selben Zweck dienen wie in `stable/`.

Die `unstable`-Distribution enthält ein Abbild des aktuellsten Entwickler-Systems. Benutzer sind willkommen diese Pakete zu benutzen und zu testen, werden aber über den Status der Einsatzbereitschaft gewarnt. Der Vorteil der Benutzung der `unstable`-Distribution ist, dass man immer auf dem Laufenden mit der aktuellsten Debian-Software ist – geht jedoch etwas schief, so muss man auch damit umgehen können. :-)

Der aktuelle Status der Fehler in der `unstable`-Distribution wird auf der Unstable Problems (`http://ftp-master.debian.org/testing/unstable_probs.html`)-Webseite aufgeführt.

2.1.6 Die `frozen`-Distribution

Wenn die `testing`-Distribution ausgereift ist, so wird aus ihr frozen, was bedeutet, dass kein neuer Code mehr akzeptiert wird, nur noch Bugfixes, wenn nötig. Es wird auch ein neuer Verzeichnisbaum im `dists`-Verzeichnis angelegt und einem neuen Kodenamen zugeordnet. Die eingefrorene Distribution durchläuft nun einige Monate lang Tests mit zwischenzeitlichen Updates und Zwischenausgaben, welche "Test-Zyklen" genannt werden.

Wir führen eine Liste aller Fehler in der `frozen`-Distribution, welche die Veröffentlichung eines Paketes verzögern können, sowie von Fehlern, welche ähnliche Auswirkungen auf die gesamte Ausgabe haben. Sobald die Anzahl der Fehler den maximal zulässigen Wert unterschreitet, wird aus der eingefrorenen Distribution `stable`, sie wird veröffentlicht und die letzte `stable`-Distribution veraltet (und wird ins Archiv verschoben).

2.1.7 Debian-Distributions-Kodenamen

Verzeichnisnamen im `dists`-Verzeichnis, wie `sarge/` und `etch/` sind nur "Kodenamen". Wenn sich eine Debian-Distribution in der Entwicklung befindet, besitzt sie keine Versionsnummer sondern nur einen Kodenamen. Der Grund für diese Kodenamen ist das Spiegeln der Debian-Distributionen zu vereinfachen (wenn ein Verzeichnis wie `unstable` plötzlich zu `stable/` umbenannt wird, so müsste vieles erneut heruntergeladen werden).

Zurzeit ist `stable/` ein symbolischer Link zu `sarge/` und `testing/` ist ein symbolischer Link zu `etch/`. Das bedeutet, dass `Sarge` die aktuelle `stable`-Distribution und `Etch` die aktuelle `testing`-Distribution ist.

`unstable/` ist ein permanenter symbolischer Link zu `sid/`, so wie Sid ständig für die `unstable`-Distribution steht.

2.1.8 In der Vergangenheit verwendete Kodenamen

Andere bereits verwendete Kodenamen sind: "Buzz" für Ausgabe 1.1, "Rex" für Ausgabe 1.2, "Bo" für die Ausgaben 1.3.x, "Hamm" für Ausgabe 2.0, "Slink" für Ausgabe 2.1, "Potato" für Ausgabe 2.2, "Woody" für Ausgabe 3.0 und "Sarge" für Ausgabe 3.1.

2.1.9 Die Herkunft der Kodenamen

Bisher wurden Personen aus dem Film *Toy Story* von Pixar verwendet.

- **Buzz** (Buzz Lightyear) war der Astronaut,
- **Rex** war der Tyrannosaurus,
- **Bo** (Bo Peep, dt. Porzelienchen) war das Mädchen, das sich um die Schafe kümmerte,
- **Hamm** war das Sparschwein (dt. Specki),
- **Slink** (Slinky Dog) war der Spielzeughund,
- **Potato** war natürlich Mr. Potato Head (der Kartoffelkopf, dt. Charly Naseweis),
- **Woody** war der Cowboy,
- **Sarge** war der Anführer der grünen Plastikarmee-Männer,
- **Etch** (Etch-a-Sketch) war die Schreibtafel,
- **Sid** war ein Nachbarsjunge, welcher Spielzeug zerstörte.

2.1.10 Das `pool`-Verzeichnis

Früher wurden Pakete in dem Unterverzeichnis von `dists` aufbewahrt, welches der verwendeten Distribution entsprach. Es stellte sich heraus, dass dies einige Probleme verursachte, wie z.B. große Bandbreitenverschwendung auf Mirrors nach einigen großen Änderungen.

Pakete werden nun in einem großen "Pool" gespeichert, entsprechend dem Namen des Quellpakets strukturiert. Um dies handhaben zu können, wurde der Pool je nach Abschnitt (`main`, `contrib` und `non-free`) sowie dem ersten Buchstaben des Quellpakets unterteilt. Diese Verzeichnisse enthalten verschiedene Dateien: die Binärpakete für jede Architektur und das Quellpaket von welchem die Binärpakete erzeugt wurden.

Man kann herausfinden wo sich ein Paket befindet, indem man ein Kommando wie `apt-cache showsrc` *`Paketname`* aufruft und nach der "Directory:"-Zeile schaut. Das `apache`-Paket wird z.B. unter `pool/main/a/apache/` gespeichert. Da es sehr viele `lib*`-Pakete gibt, werden diese gesondert behandelt: das `libpaper`-Paket wird beispielsweise unter `pool/main/libp/libpaper/` gespeichert.

Die `dists`-Verzeichnisse werden nach wie vor für die Index-Dateien, welche von Programmen wie `apt` verwendet werden, genutzt. Ebenso wurden während dies geschrieben wird, ältere Distributionen noch nicht angepasst um Pools zu nutzen, deswegen werden Sie auch Pfade finden, die den Distributionsnamen wie `potato` oder `woody` im "Directory"-Feld enthalten.

Normalerweise muss man sich um dies nicht kümmern, da neue `apt` und wahrscheinlich ältere `dpkg-ftp` Programme (vergleiche 'Methoden zum Aktualisieren eines Debian-Systems' auf Seite 20) dies problemlos handhaben. Sind Sie an weiteren Informationen interessiert, so sei auf die RFC: Implementation von Paketpools (`http://lists.debian.org/debian-devel-announce/2000/debian-devel-announce-200010/msg00007.html`) verwiesen.

2.1.11 Historische Bemerkungen über Sid

Als das heutige Sid noch nicht existierte, gab es im Debian-Archiv nur einen Zweig für nicht ausgereifte Pakete: es gab die Annahme, dass, wenn eine Architektur im aktuellen `unstable/` hinzukam, sie veröffentlicht wurde, wenn diese Distribution zum neuen `stable`-Zweig wurde. Für viele Architekturen war das nicht der Fall, was dazu führte, dass diese Verzeichnisse während der Veröffentlichung verschoben wurden. Dies war unpraktisch, da die Verschiebung zu einer großen Bandbreitenbelastung führte.

Die Archiv-Administratoren umgingen das Problem einige Jahre, indem sie Binaries für nicht-veröffentlichte Architekturen in einem speziellen Verzeichnis namens `sid` bereitstellten. Für solche noch nicht veröffentlichte Architekturen wurde das erste Mal als sie veröffentlicht wurden, ein Link vom aktuellen `stable/` zu `sid/` angelegt, und später wurden sie wie üblich unter `unstable/` veröffentlicht. Diese Vorgehensweise war zum Teil für die Anwender verworren.

Mit Beginn der Paket-Pools (vergleiche 'Das `pool`-Verzeichnis' auf der vorherigen Seite) während der Entwicklung der Woody-Distribution, wurden Binärpakete unabhängig von der Distribution vorschriftsmäßig im Pool gehalten, so dass die Veröffentlichung einer Distribution nicht länger zu einer großen Bandbreitenverschwendung auf den Mirrors führte (es gibt dennoch während der Entwicklung eine relativ große Bandbreitenauslastung).

2.1.12 Heraufgeladene Pakete in `incoming/`

Heraufgeladene Pakete befinden sich zunächst unter `http://incoming.debian.org/` nachdem sie überprüft wurden, um sicherzustellen, dass sie wirklich von einem Debian-Entwickler stammen, (und sie werden in das `DELAYED`-Unterverzeichnis verschoben, wenn das Paket nicht von einem Entwickler stammt, d.h. ein Non-Maintainer Upload (NMU) ist). Einmal pro Tag werden sie von `incoming/` nach `unstable/` verschoben.

Im Notfall kann man Pakete aus `incoming/` installieren, bevor sie nach `unstable/` kommen.

2.1.13 Wiederauffinden eines älteren Paketes

Während die aktuellsten Debian-Distributionen unter dem `debian`-Verzeichnis auf jedem Debian-Mirror (`http://www.debian.org/mirror/`) zu finden sind, werden die Archive für ältere Debian-Distributionen wie Slink unter `http://archive.debian.org/` oder unterhalb des `debian-archive` Verzeichnis auf jedem Debian-Mirror gehalten.

Ältere `testing`- und `unstable`-Pakete befinden sich auf `http://snapshot.debian.net/`.

2.1.14 Architekturabhängige Verzeichnisse

Innerhalb jedes der wichtigen Verzeichnisbäume (`dists/stable/main`, `dists/stable/contrib`, `dists/stable/non-free`, `dists/unstable/main`, etc.), befinden sich die

Einträge der Binärpakete in Unterverzeichnissen, deren Namen die Prozessorarchitektur kennzeichnen, für die sie kompiliert wurden.

- `binary-all/` für Pakete, welche architekturunabhängig sind. Dies umschließt z.B. Perl-Skripte oder Dokumentationen.
- `binary-`*`Plattform`*`/` für Pakete, welche sich auf einer einzelnen Binärplattform starten lassen.

Es ist anzumerken, dass die aktuellen Binärpakete von `testing` und `unstable` sich nicht mehr länger in diesen Verzeichnissen, dafür aber im `pool`-Verzeichnis befinden. Die Index-Dateien (`Packages` und `Packages.gz`) befinden sich nach wie vor zwecks Abwärtskompatibilität dort.

Für die aktuell unterstützten Binärarchitekturen vergleiche die Release Notes der einzelnen Distributionen. Sie können unter der Release-Notes-Seite für stable (`http://www.debian.org/releases/stable/releasenotes`) und testing (`http://www.debian.org/releases/testing/releasenotes`) gefunden werden.

2.1.15 Der Quellcode

Der Quellcode für alles im Debian-System ist mit darin enthalten. Außerdem **fordern** die Lizenzvereinbarungen der meisten Programme im System, dass der Quellcode zusammen mit den Programmen ausgeliefert wird, oder dass dem Programm Informationen beiliegen, wie er zu erhalten ist.

Normalerweise befindet sich der Quellcode in den `source`-Verzeichnissen, welche parallel zu allen architekturspezifischen Binärverzeichnissen sind oder aktueller im `pool`-Verzeichnis (vergleiche 'Das `pool`-Verzeichnis' auf Seite 9). Um den Quellcode zu erhalten, ohne sich mit der Verzeichnisstruktur des Debian-Archivs auseinandersetzen zu müssen, kann ein Befehl wie `apt-get source` *`Meinpaketname`* genutzt werden.

Einige Pakete wie z.B. `pine`, sind wegen deren Lizenzbedingungen nur im Quellcode erhältlich. (Kürzlich wurde das `pine-tracker`-Paket bereitgestellt, um die Pine-Installation zu vereinfachen.) Die Anweisungen beschrieben in 'Portierung eines Pakets auf die `stable`-Distribution' auf Seite 78 und 'Paketerzeugung' auf Seite 228 bieten einige Möglichkeiten, um ein Paket manuell zu paketieren.

Der Quellcode kann, muss aber nicht für Pakete in `contrib`- und `non-free`-Verzeichnissen, welche formal nicht Teil des Debian-Systems sind, verfügbar sein.

2.2 Das Debian-Paketverwaltungssystem

2.2.1 Überblick über Debian-Pakete

Pakete enthalten im Allgemeinen all die Dateien, welche nötig sind um eine Menge von zusammengehörigen Kommandos oder Eigenschaften zu implementieren. Es gibt zwei Typen von Debian-Paketen:

- **Binärpakete**, welche ausführbare Programme enthalten, Konfigurationsdateien, man/info-Seiten, Copyright-Informationen und andere Dokumentationen. Diese Pakete werden in einem Debian-spezifischen Archivformat verteilt (vergleiche 'Debian-Paketformat' auf dieser Seite); sie zeichnen sich i.a. durch die `.deb`-Dateierweiterung aus. Binärpakete können mit Debians `dpkg`-Programm ausgepackt werden; Details sind in der Handbuchseite beschrieben.
- **Quellpakete**, welche eine `.dsc`-Datei enthalten, die das Quellpaket beschreibt (inklusive der Namen der folgenden Dateien), ebenso wie eine `.orig.tar.gz`-Datei, welche den ursprünglichen unveränderten Quellcode in gzip-komprimiertem tar-Format enthält und gewöhnlich eine `.diff.gz`-Datei, die Debian-spezifische Änderungen zu den Originalquellen enthält. Das Programm `dpkg-source` packt und entpackt Debian-Quellpakete; Details sind in der Handbuchseite enthalten.

Die Installation der Software durch das Paketsystem nutzt "Abhängigkeiten", welche sorgfältig vom Paketbetreuer bestimmt wurden. Diese Abhängigkeiten sind in der `control`-Datei, die jedem Paket zugeordnet ist, enthalten. Das Paket, welches den GNU C Compiler enthält (`gcc`), "hängt" z.B. von dem Paketen `binutils` "ab", das den Linker und Assembler enthält. Versucht ein Benutzer `gcc` zu installieren, ohne zuvor `binutils` installiert zu haben, so wird das Paketverwaltungssystem (dpkg) die Fehlermeldung ausgeben, dass es `binutils` benötigt und die Installation von `gcc` abbrechen. (Dennoch kann dieses Verhalten vom Nutzer geändert werden; vergleiche `dpkg(8)`.) Für weitere Einzelheiten wird auf 'Paketabhängigkeiten' auf Seite 16 verwiesen.

Debian's Paketverwaltungstools können benutzt werden um:

- Pakete oder Teile von Paketen zu manipulieren und handzuhaben,
- dem Nutzer im Aufteilen von Paketen zu helfen, welche mittels kleiner Medien wie Disketten übertragen werden müssen,
- Entwickler beim Erzeugen von Paketarchiven zu helfen und um
- den Nutzern bei der Installation von Paketen, die sich auf entfernten Debian-Archiven befinden, zu helfen.

2.2.2 Debian-Paketformat

Ein Debian-"Paket" oder eine Debian-Archivdatei enthält ausführbare Dateien, Bibliotheken und Dokumentationen, welche einem bestimmten Programm oder einer Menge von zugehörigen Programmen zugeordnet sind. Normalerweise hat eine Debian-Archivdatei einen Dateinamen der mit `.deb` endet. [1]

[1] Das `debian-installer`-Projekt führte Dateinamen ein, die auf `.udeb` enden. Kurzum, es ist ein Mikro-`.deb` Format, das den Debian-Richtlinien für freie Software nicht exakt folgen muss. So fehlt z.B. die Dokumen-

Die internen Einzelheiten dieses Debian-Binärpaketformats werden in der `deb(5)` Handbuchseite beschrieben. Da dieses interne Format in Zukunft geändert werden kann (zwischen verschiedenen Ausgaben von Debian), sollte stets `dpkg-deb(1)` für Änderungen der `.deb`-Dateien verwendet werden.

Zumindest in der Sarge-Distribution kann auf alle Debian-Archivdateien mit den Standard-Unix-Kommandos `ar` und `tar` zugegriffen werden, auch wenn das `dpkg`-Kommando nicht verfügbar ist.

2.2.3 Namenskonventionen für Debians Paketdateinamen

Die Debian-Paketdateinamen folgen den Konventionen:

foo_Versionsnummer-DebianRevisionsnummer`.deb`

wobei *foo* für den Paketnamen steht. Zur Kontrolle kann man den Paketnamen, welcher einer bestimmten Debian-Archivdatei (`.deb`-Datei) zugeordnet ist, durch eine der folgenden Möglichkeiten bestimmen:

- betrachten der "Packages"-Datei in dem Verzeichnis, wo sich die `.deb`-Datei im Debian-Archiv befand. Diese Datei enthält für jedes Paket einen Eintrag, welcher es genau beschreibt; das erste Feld in jedem Eintrag ist der formale Paketname.
- man benutzt das Kommando `dpkg --info` *foo_VVV-RRR*`.deb` (wobei *VVV* und *RRR* die Versions- bzw. Revisionsnummer des abgefragten Pakets sind). Dies zeigt unter anderem den Paketnamen an, welcher der zu entpackenden Archivdatei entspricht.

Der *VVV*-Teil ist die Versionsnummer, die vom Entwickler des Programms vergeben wurde. Es gibt dafür kein allgemein gültiges Format, sowohl "19990513" als auch "1.3.8pre1" ist denkbar.

Der *RRR*-Teil ist die Debian-Revisionsnummer und wird vom Debian-Entwickler angegeben (oder von einem Nutzer, wenn dieser das Paket selbst baut). Diese Nummer entspricht dem Revisionslevel des Debian-Pakets; ein neues Revisionslevel kennzeichnet in der Regel Änderungen in Debians Makefile (`debian/rules`), in Debians Kontrolldatei (`debian/control`), in Installations- oder Deinstallationsskripten (`debian/p*`) oder in den Konfigurationsdateien, die mit dem Paket genutzt werden.

tation und es wird nur vom `debian-installer` genutzt, dem neuen Debian-Installationsprogramm, das für die Sarge-Ausgabe entwickelt wurde. Die Dateiformate von `.deb` und `.udeb` sind identisch. Das Programm `udpkg`, das mit `.udeb`-Paketen umgehen kann, hat eingeschränktere Fähigkeiten als das Standard-`dpkg` und unterstützt weniger Paketbeziehungen. Der Unterschied im Namen besteht, weil die Debian-Archivbetreuer nicht glücklich mit `.debs` im Archiv sind, die den Richtlinien nicht folgen. Deshalb wurde ein unterschiedlicher Name gewählt um dies zu betonen und um es unwahrscheinlicher zu machen, dass Benutzer diese unabsichtlich auf einem richtigen System installieren. `.udebs` werden nur in einer Initial Ramdisk während der Basisinstallation genutzt, um ein sehr beschränktes Debian-System zu erstellen.

2.2.4 Bewahren der lokalen Konfiguration

Die durch den Nutzer konfigurierbaren Dateien werden durch Debians “conffiles”-Mechanismus bewahrt. Konfigurationsdateien (diese befinden sich im Allgemeinen in `/etc/`) werden in den `conffiles` innerhalb von Debians Paketsystem angegeben. Das Paketverwaltungssystem garantiert, dass diese Dateien bei einer Paketaktualisierung (einem Upgrade) nicht überschrieben werden.

Wenn es möglich ist, das System ohne Modifizierungen von Dateien anzupassen, welche zu verschiedenen Debian-Paketen gehören, so ist es in der Regel eine gute Idee, diese nicht zu verändern, selbst wenn es sich um “conffiles” handelt. Dies ermöglicht schnellere und saubere Aktualisierungen.

Um zu bestimmen, welche Dateien während eines Upgrades bewahrt werden, kann man

```
dpkg --status Paket
```

ausführen und nach “Conffiles:” schauen.

Einzelheiten zum Inhalt einer Debian-`conffiles`-Datei sind im Debian Policy Manual, Abschnitt 11.7 (vergleiche ‘Referenzen’ auf Seite 235) zu finden.

2.2.5 Debian-Wartungsskripte

Debian-Wartungsskripte sind ausführbare Skripte, welche automatisch gestartet werden bevor oder nachdem ein Paket installiert wird. Zusammen mit einer Datei namens `control` sind all diese Dateien Teil des “control”-Abschnitts einer Debian-Archivdatei.

Die einzelnen Dateien sind:

preinst Dieses Skript wird ausgeführt, bevor das Paket aus der Debian-Archivdatei (`.deb`) ausgepackt wird. Viele “preinst”-Skripte beenden Dienste von Paketen, welche aktualisiert werden, bis deren Installation oder Upgrade vollzogen ist (d.h. nach der erfolgreichen Ausführung des “postinst”-Skriptes).

postinst Dieses Skript schließt typischerweise jede nötige Konfiguration eines Paketes ab, nachdem es aus der Debian-Archivdatei (`.deb`) ausgepackt wurde. Oft fragen “postinst”-Skripte die Nutzer nach Daten, und/oder weisen sie darauf hin, dass, wenn sie die Standardwerte akzeptieren, sie später die Möglichkeit haben zurückzugehen und das Paket zu rekonfigurieren, sollte dies nötig sein. Viele “postinst”-Skripte führen Kommandos aus, welche nötig sind, um Dienste zu starten oder nach der Installation oder dem Upgrade neu zu starten.

prerm Dieses Skript beendet typischerweise Daemonen, welche dem Paket zugeordnet sind. Es wird ausgeführt, bevor die zum Paket gehörenden Dateien gelöscht werden.

postrm Dieses Skript modifiziert typischerweise Links oder andere Dateien, welche dem Paket zugeordnet sind und/oder entfernen Dateien, welche von ihm erzeugt wurden. (Siehe auch 'Virtuelle Pakete' auf der nächsten Seite.)

Zurzeit können all diese Kontrolldateien im Verzeichnis `/var/lib/dpkg/info` gefunden werden. Die auf das Paket `foo` bezogenen Dateien beginnen mit "foo" und haben die entsprechende Dateierweiterungen "preinst", "postinst", u.s.w. Die Datei `foo.list` in diesem Verzeichnis enthält alle Dateien, welche mit dem Paket `foo` installiert wurden. (Es ist zu beachten, dass die Position dieser Dateien eine interne `dpkg`-Eigenschaft ist und sich in der Zukunft ändern kann.)

2.2.6 Paket-Prioritäten

Jedem Debian-Paket ist eine **Priorität** vom Distributionsbetreuer zugeordnet worden, um die Arbeit des Paketverwaltungssystems zu vereinfachen. Die Prioritäten sind:

- **Erforderliche** Pakete werden benötigt für die zuverlässige Funktionalität des Systems.

  Dies schließt alle Tools, welche nötig sind, um das System zu reparieren, ein. Diese Pakete dürfen nicht entfernt werden, andernfalls kann das System komplett versagen und man ist nicht einmal in der Lage `dpkg` zum Wiederherstellen zu nutzen. Systeme die nur die erforderlichen Pakete enthalten, sind wahrscheinlich ungeeignet für die meisten Aufgaben, jedoch kann der Systemadministrator jederzeit neue Software installieren.

- **Wichtige** Pakete sollten auf jedem Unix-artigen System gefunden werden.

  Andere Pakete ohne die das System nicht gut oder brauchbar arbeitet, haben diese Priorität. Dies schließt **nicht** Emacs, X11, TeX oder andere große Anwendungen ein. Diese Pakete erzeugen nur die nötige Infrastruktur.

- **Standard**pakete sind auf jedem Linuxsystem üblich, inklusive einem kleinen aber nicht zu sehr beschränkten textbasierten System.

  Dies ist der Standardinstallationsumfang, solange der Nutzer nichts anderes wählt. "Standard" enthält nicht viele große Anwendungen, aber es enthält Emacs (das ist mehr eine Infrastruktur als eine Anwendung) und eine geeignete Teilmenge von TeX und LaTeX (sofern dies ohne X möglich ist).

- **Optionale** Pakete enthalten all diese, welche man vernünftigerweise installieren möchte, auch wenn man damit nicht vertraut ist und keine speziellen Anforderungen daran hat.

  Dies schließt X11, eine vollständige TeX-Distribution und viele Anwendungen mit ein.

- **Zusätzliche** Pakete sind entweder nicht mit anderen Paketen mit höherer Priorität verträglich, sind wahrscheinlich nur nützlich, wenn man sie bereits näher kennt oder haben spezielle Anforderungen, die sie für "Optional" ungeeignet machen.

Beachten Sie die Unterschiede zwischen “Priority: required”, “Section: base” und “Essential: yes” in der Paketbeschreibung. “Section: base” bedeutet, dass dieses Paket vor allen anderen in einem neuen System installiert wird. Die meisten der Pakete mit “Section: base” enthalten “Priority: required” oder zumindest “Priority: important” und viele von diesen sind mit “Essential: yes” versehen. “Essential: yes” bedeutet, dass das Paket die Angabe einer bestimmten Option an das Paketmanagementsystem wie `dpkg` erfordert, wenn es aus dem System entfernt werden soll. Beispielsweise sind `libc6`, `mawk` und `makedev` Pakete mit “Priority: required” und “Section: base” aber enthalten nicht “Essential: yes”.

2.2.7 Virtuelle Pakete

Ein virtuelles Paket ist ein allgemeiner Name, welcher für eine Gruppe von Paketen steht, die alle ähnliche Funktionalitäten bereitstellen. Die Programme `tin` und `trn` sind beispielsweise beide News-Reader und einer von beiden sollte deshalb die Abhängigkeiten eines Programms, welches einen News-Reader im System benötigt um richtig funktionieren zu können, erfüllen. Deshalb sagt man, dass beide das “virtuelle Paket” `news-reader` bereitstellen.

Analog stellen `exim` und `sendmail` die Funktionalität eines Mail-Transport-Agent bereit. Deshalb wird das virtuelle Paket `mail-transport-agent` von beiden angeboten. Wenn eines dieser Programme installiert ist, dann wird jedes Programm, das von der Installation eines Mail-Transport-Agent abhängt, mit der Existenz dieses virtuellen Paketes zufrieden sein.

Debian besitzt einen Mechanismus so dass, wenn mehr als ein Paket, welches das selbe virtuelle Paket bereitstellt, auf dem System installiert ist, der Systemadministrator ein Programm bevorzugt auswählen kann. Das entsprechende Kommando ist `update-alternatives` und wird genauer in ‘Alternative Befehle’ auf Seite 82 beschrieben.

2.2.8 Paketabhängigkeiten

Das Debian-Paketsystem enthält “Paketabhängigkeiten”, welche (in einem einzelnen Eintrag) festhalten, wie ein Programm A unabhängig von der Existenz eines Programms B auf einem System arbeiten kann.

- Paket A **hängt** von Paket B **ab**, wenn B unbedingt installiert sein muss, um A starten zu können. In einigen Fällen hängt A nicht nur von B, sondern einer speziellen Version von B ab. In diesem Fall ist die Versionsabhängigkeit im Allgemeinen eine untere Schranke, d.h. A hängt von einer beliebigen Version von B ab, welche aktueller als eine angegebene Version ist.

- Paket A **empfiehlt** Paket B, wenn der Paketbetreuer meint, dass die meisten Nutzer A nicht ohne die von B bereitgestellte Funktionalität haben wollen.

- Paket A **schlägt** Paket B **vor**, wenn B Dateien enthält, welche sich auf die Funktionalität von A beziehen (und diese eventuell erweitern).

- Paket A **kollidiert** mit Paket B, wenn A nicht funktioniert, sofern B auf dem System installiert ist. Sehr oft bestehen solche **Konflikte** darin, dass A Dateien enthält, die gegenüber denen in B verbessert wurden. Der "Konflikt"-Status wird oft mit "ersetzt" kombiniert.
- Paket A **ersetzt** Paket B, wenn Dateien, die von B installiert wurden, von A entfernt und (in einigen Fällen) durch Dateien in A überschrieben werden.
- Paket A **unterstützt** Paket B, wenn alle Dateien und Funktionalitäten von B in A verfügbar sind. Dieser Mechanismus steht für Nutzer mit geringem Plattenplatz zur Verfügung, um nur den Teil von A zu installieren, welcher absolut nötig ist.

Weitere detaillierte Informationen zur Verwendung dieser Terme können im *Packaging Manual* und dem *Policy Manual* gefunden werden.

Es ist zu beachten, dass `dselect` eine bessere Kontrolle über Pakete, die mit **empfiehlt** und **schlägt vor** spezifiziert werden, bietet als `apt-get`, was einfach alle **hängt ab** Pakete wählt und **empfiehlt** und **schlägt vor** ignoriert. Beide Programme nutzen in aktuellen Versionen APT als Back-end.

2.2.9 Die Bedeutung von "pre-depends"

"Pre-depends" ist eine spezielle Abhängigkeit. Im Falle eines gewöhnlichen Pakets entpackt `dpkg` die Archivdatei (d.h. dessen `.deb`-Datei) unabhängig davon, ob die Dateien, von denen es abhängt, im System existieren oder nicht. Entpacken bedeutet im Wesentlichen, dass `dpkg` die Dateien aus der Archivdatei extrahiert und korrekt im Dateisystem platziert. **Hängen** diese Pakete von der Existenz anderer Pakete auf dem System **ab**, dann wird sich `dpkg` weigern, die Installation zu beenden (durch Ausführen des "konfigurieren" Schritts), bis die anderen Pakete installiert sind.

Dennoch gibt es einige Pakete, bei denen sich `dpkg` sogar weigert, sie zu entpacken, bis gewisse Abhängigkeiten aufgelöst sind. Solche Pakete hängen vom Vorhandensein anderer Pakete im Sinne von "pre-depend" ab. Das Debian-Projekt führte diesen Mechanismus ein, um das System sicher vom `a.out`- auf das ELF-Format zu aktualisieren, wobei die **Reihenfolge** in welcher die Pakete entpackt werden, bedeutend ist. Es gibt weitere große Update-Situationen, wo diese Methode nützlich ist, z.B. für Pakete mit der "erforderlich"-Priorität und deren libc-Abhängigkeit.

Erneut wird für detailliertere Informationen über dies auf das *Packaging Manual* verwiesen.

2.2.10 Paket-Status

Der Paket-Status kann "unbekannt", "installieren", "entfernen", "säubern" oder "halten" sein. Diese "gewünschten" Werte kennzeichnen, was der Nutzer mit einem Paket beabsichtigte (entweder durch Anwahl des "[A]uswählen" Punktes in `dselect` oder durch direkten Aufruf von `dpkg`).

Deren Bedeutung ist:

- **unbekannt** - der Nutzer hat niemals angegeben, ob er das Paket will.
- **installieren** - der Nutzer will das Paket installiert oder aktualisiert haben.
- **entfernen** - der Nutzer will das Paket entfernen lassen, ohne das existierende Konfigurationsdateien gelöscht werden.
- **säubern** - der Nutzer will das Paket komplett entfernt haben, inklusive der Konfigurationsdateien.
- **halten** - der Nutzer will das Paket nicht verarbeiten lassen, d.h. er möchte die aktuelle Version im aktuellen Status belassen, unabhängig vom Wert.

2.2.11 Zurückhalten von Paketen von einem Upgrade

Es gibt zwei Mechanismen zum Zurückhalten von Paketen von einem Upgrade, durch `dpkg` oder beginnend mit Woody durch APT.

Mit `dpkg` ist zuerst die Paketauswahlliste zu exportieren:

```
dpkg --get-selections \* > Paketauswahl.txt
```

Dann muss die erzeugte Datei *`Paketauswahl.txt`* editiert werden, indem die Zeile, welche das zu haltende Paket, z.B. `libc6`, enthält, von:

```
libc6                            install
```

auf:

```
libc6                            hold
```

geändert wird. Nach dem Speichern ist die Datei in die `dpkg`-Datenbank zurückzuladen mit:

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

Kennt man den Paketnamen des zu haltenden Pakets, kann man auch einfach

```
echo libc6 hold | dpkg --set-selections
```

starten. Wann immer der Installations-Prozess dieses Paket bearbeitet (zu upgraden versucht), hält er es zurück.

Der selbe Effekt kann mit `dselect` erreicht werden. Dazu ist einfach der Punkt [A]uswählen und dann das Paket zu wählen, dessen Status beibehalten werden soll sowie schließlich '=' (oder 'H') zu drücken. Die Änderungen werden sofort aktiv, nachdem das [A]uswählen Menü beendet wird.

Das APT-System in der Woody-Distribution hat einen neuen "Alternativen" Mechanismus zum Halten von Paketen während des Archivabfrageprozesses unter Verwendung von `Pin-Priority`. Vergleiche die Handbuchseite `apt_preferences(5)` sowie `http://www.debian.org/doc/manuals/apt-howto/` oder das `apt-howto`-Paket; 'Überblick über `/etc/apt/preferences`' auf Seite 69 enthält auch eine kurze Erläuterung.

2.2.12 Quellpakete

Quellpakete befinden sich in einem Verzeichnis namens `source` und können entweder manuell heruntergeladen werden oder durch

```
apt-get source foo
```

(vergleiche die `apt-get(8)` Handbuchseite wie man APT dazu bringt, dies zu tun).

2.2.13 Erzeugen von Binär- aus Quellcodepaketen

Für ein Paket *foo* benötigt man alle *foo_**`.dsc`-, *foo_**`.tar.gz`- und *foo_*.diff*`.gz`-Dateien, um die Quellen zu übersetzen (Bemerkung: es gibt keine `.diff.gz`-Datei für ein natives Debian-Paket).

Sind diese Dateien vorhanden und ist das `dpkg-dev`-Paket installiert, so extrahiert das Kommando

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

das Paket in ein Verzeichnis namens *foo-version*.

Zum Erzeugen des Binärpakets ist Folgendes auszuführen:

```
$ cd foo-version
$ su -c "apt-get update; apt-get install fakeroot"
$ dpkg-buildpackage -rfakeroot -us -uc
```

Und danach

```
# su -c "dpkg -i ../foo_version-revision_arch.deb"
```

zum Installieren des neu gebildeten Pakets. Vergleiche 'Portierung eines Pakets auf die `stable`-Distribution' auf Seite 78.

2.2.14 Erzeugen neuer Debian-Pakete

Für detaillierte Informationen zum Erzeugen neuer Pakete sollte der *New Maintainers' Guide* gelesen werden, verfügbar im `maint-guide` Paket, oder unter `http://www.debian.org/doc/manuals/maint-guide/`.

2.3 Aktualisierung eines Debian-Systems

Einer von Debians Vorzügen ist die Unterstützung eines konsistenten Upgrade-Wegs sowie ein sicherer Upgrade-Prozess und es wird stets versucht, eine ältere Ausgabe problemlos zu aktualisieren. Pakete warnen den Nutzer, sollten bedeutende Bemerkungen während des Upgrade-Prozesses auftreten und bieten oft eine Lösung zu einem möglichen Problem.

Man sollte auch die Release Notes lesen, das Dokument, das die Details zu spezifischen Upgrades enthält. Es wird mit allen Debian-CDs ausgeliefert und ist im WWW unter `http://www.debian.org/releases/stable/releasenotes` oder `http://www.debian.org/releases/testing/releasenotes` verfügbar.

Eine praktische Anleitung zum Aktualisieren wird in 'Debian-Paketverwaltung' auf Seite 63 bereitgestellt. Dieser Abschnitt beschreibt die grundlegenden Einzelheiten.

2.3.1 Methoden zum Aktualisieren eines Debian-Systems

Man kann immer einfach einen anonymen FTP- oder `wget`-Aufruf zu einem Debian-Archiv starten, die Verzeichnisse durchsehen, bis man die gewünschte Datei gefunden hat, diese herunterladen und schließlich mittels `dpkg` installieren. (Es ist zu beachten, dass `dpkg` die aktualisierten Dateien immer in das korrekte Verzeichnis installiert, auch in einem laufenden System.) Manchmal kommt es dennoch vor, dass ein überarbeitetes Paket die Installation einer neuen Version eines anderen Paketes erfordert. In diesem Fall schlägt die Installation fehl, wenn das andere Programm nicht installiert ist.

Viele Personen finden dieses manuelle Vorgehen zu Zeit aufwendig, da Debian sich so schnell entwickelt – typischerweise werden ein Dutzend oder mehr Pakete jede Woche hochgeladen. Diese Zahl ist kurz vor einer neuen Veröffentlichung noch größer. Um dies besser handhaben zu können, bevorzugen viele Personen ein automatisches Programm zum Upgraden. Einige spezialisierte Paketmanagement-Tools sind für diesen Zweck verwendbar.

2.3.2 Überblick über Paketverwaltungstools

Das Debian-Paketverwaltungssystem hat zwei Aufgaben: die Manipulation der Paketdatei und das Herunterladen von Paketdateien aus dem Debian-Archiv. `dpkg` erfüllt die erste Aufgabe, APT und `dselect` die letztere.

2.3.3 `dpkg`

Dies ist das wichtigste Programm zum Manipulieren von Paketdateien. Für eine ausführliche Beschreibung ist `dpkg(8)` zu lesen.

`dpkg` wird mit einigen wichtigen Programmen ausgeliefert.

- `dpkg-deb`: Manipuliert `.deb` Dateien. `dpkg-deb(1)`

- dpkg-ftp: Ein älteres Programm zum Herunterladen von Paketen. dpkg-ftp(1)
- dpkg-mountable: Ein älteres Programm zum Herunterladen von Paketen. dpkg-mountable(1)
- dpkg-split: Teilt ein größeres Paket in kleinere Dateien auf. dpkg-split(1)

dpkg-ftp und dpkg-mountable wurden durch das APT-System ersetzt.

2.3.4 APT

APT, das zukunftsweisende Paket-Tool (Advanced Packaging Tool) ist eine fortschrittliche Schnittstelle zu Debians Paketsystem und besteht aus verschiedenen Programmen, deren Namen oft mit "apt-" beginnen. apt-get, apt-cache und apt-cdrom sind die Kommandozeilen-Tools zum Umgang mit Paketen. Diese werden auch von anderen Programmen, wie dselect und aptitude genutzt.

Für weitere Informationen ist das apt-Paket zu installieren und apt-get(8), apt-cache(8), apt-cdrom(8), apt.conf(5), sources.list(5), apt_preferences(5) (Woody) sowie /usr/share/doc/apt/guide.html/index.html zu lesen.

Weitere Informationen finden sich im APT HOWTO (http://www.debian.org/doc/manuals/apt-howto/). Dazu kann in Sarge apt-howto-de installiert werden und steht dann unter /usr/share/doc/Debian/apt-howto/ zur Verfügung (Woody enthält nur die englische Version apt-howto).

apt-get upgrade und apt-get dist-upgrade installieren nur die Pakete, die unter "Depends:" ("hängt ab:") aufgeführt sind und übersieht alle unter "Recommends:" ("empfiehlt:") und "Suggests:" ("schlägt vor:") gelisteten Pakete. Um dies zu vermeiden, sollte dselect verwendet werden.

2.3.5 dselect

Dieses Programm besitzt eine Menüoberfläche zu Debians Paketverwaltungssystem. Es ist besonders für die Erstinstallation und große Upgrades nützlich. Vergleiche 'dselect' auf Seite 65.

Für weitere Informationen sollte das install-doc-Paket installiert und /usr/share/doc/install-doc/dselect-beginner.en.html oder dselect Documentation for Beginners (http://www.debian.org/releases/woody/i386/dselect-beginner) gelesen werden.

2.3.6 Aktualisieren eines laufenden Systems

Der Kernel (das Dateisystem) in Debian-Systemen unterstützt das Ersetzen von Dateien auch während sie benutzt werden.

Es wird auch ein Programm namens `start-stop-daemon` bereitgestellt, das verwendet wird, um Daemonen während des Bootens zu starten oder zum Stoppen von Daemonen, wenn das Kernel-Runlevel geändert wird (z.B. von Mehrbenutzer- zu Einzelbenutzermodus oder zu "halt"). Das gleiche Programm wird von Installationsskripten genutzt, wenn ein Paket, das einen Daemon enthält, installiert wird, um laufende Daemonen zu stoppen und bei Bedarf neuzustarten.

Es wird darauf hingewiesen, dass das Debian-System den Einzelnutzermodus nicht erfordert, um ein laufendes System zu aktualisieren.

2.3.7 Heruntergeladene und zwischengespeicherte `.deb`-Archiv-Dateien

Hat man Paketdateien manuell auf die Festplatte heruntergeladen (was nicht unbedingt notwendig ist, vergleiche die obige Beschreibung von `dpkg-ftp` oder APT), dann kann man die `.deb`-Dateien nach der Installation der Pakete aus dem System entfernen.

Wenn APT verwendet wird, so werden diese Dateien im `/var/cache/apt/archives`-Verzeichnis zwischengespeichert. Sie können nach der Installation gelöscht werden (`apt-get clean`) oder auf einen anderen Rechner ins `/var/cache/apt/archives` Verzeichnis kopiert werden, um das Herunterladen während mehrerer Installationen zu vermeiden.

2.3.8 Aufbewahren des Datensatzes für Upgrades

`dpkg` bewahrt einen Datensatz aller Pakete, die ausgepackt, konfiguriert, entfernt und/oder gesäubert wurden, auf. Jedoch wird (zurzeit) kein Protokoll der Terminaleingaben, während der Paketmanipulation, erzeugt.

Der einfachste Weg, dieses Problem zu umgehen, ist Sitzungen von `dpkg`, `dselect`, `apt-get`, etc. innerhalb des `script(1)` Programms ablaufen zu lassen.

2.4 Der Debian-Bootvorgang

2.4.1 Das `init`-Programm

Wie alle Unices, wird Debian durch das Programm `init` gestartet. Die Konfigurationsdatei für `init` (dies ist `/etc/inittab`) gibt an, dass das erste zu startende Skript `/etc/init.d/rcS` ist. Dieses Skript startet alle anderen Skripte in `/etc/rcS.d/`, entweder indem diese eingebunden oder explizit als Unterprozess aufgerufen werden, je nach Dateierweiterung. Diese Skripte initialisieren das System indem sie z.B. Dateisysteme überprüfen und einbinden, Module laden, Netzwerk-Dienste starten, die Uhrzeit setzen, u.a. Danach werden zwecks Kompatibilität die Dateien (mit Ausnahme der mit einem '.' im Dateinamen) in `/etc/rc.boot/` ausgeführt. Jedes Skript in diesem Verzeichnis ist normalerweise dem Systemadministrator vorbehalten, die Verwendung dieser in Paketen wird missbilligt. Vergleichen Sie 'Hinweise zur System-Inititalisierung' auf Seite 125 und System run levels and init.d scripts (`http:`

`//www.debian.org/doc/debian-policy/ch-opersys#s-sysvinit`) in den Debian-Richtlinien für weitere Informationen.

2.4.2 Runlevel

Nach dem Bootprozess führt `init` alle Startskripte in einem durch das Standard-Runlevel festgelegten Verzeichnis aus. (Dieses Runlevel wird durch den Eintrag `id` in `/etc/inittab` festgelegt). Wie viele System-V-kompatible Unices hat Linux 7 Runlevel:

- 0 (Anhalten des Systems),
- 1 (Einzelnutzer Modus),
- 2 bis 5 (verschiedene Mehrbenutzer-Modi) und
- 6 (Neustart des Systems).

Für Debian-Systeme gilt `id=2`, was bedeutet, dass das Standard-Runlevel 2 sein wird, wenn der Mehrbenutzer-Modus aktiv ist und die Skripte in `/etc/rc2.d/` werden ausgeführt.

In Wirklichkeit sind die Skripte in den Verzeichnissen `/etc/rcN.d/` nur symbolische Links zu Skripten in `/etc/init.d/`. Dennoch werden die **Namen** der Dateien in jedem der `/etc/rcN.d/`-Verzeichnisse individuell gewählt, um anzugeben **wie** die Skripte in `/etc/init.d/` gestartet werden. Speziell werden bevor ein Runlevel aktiv wird, alle Skripte die mit 'K' beginnen ausgeführt; diese Skripte beenden Dienste. Danach werden alle Skripte die mit 'S' beginnen gestartet; diese Skripte starten Dienste. Die zweistellige dem 'K' oder 'S' folgende Nummer bestimmt die Reihenfolge der Ausführung. Skripte mit kleinerer Nummer werden zuerst ausgeführt.

Dieses Vorgehen funktioniert, da die Skripte in `/etc/init.d/` alle ein Argument akzeptieren, das entweder "start", "stop", "reload", "restart" oder "force-reload" sein kann und eine dem Argument entsprechende Aktion ausführen (starten, stoppen, neuladen, neustarten, erzwinge Neuladen). Diese Skripte können auch ausgeführt werden, nachdem das System gebootet wurde, um verschiedene Prozesse zu kontrollieren.

Zum Beispiel führt das Argument "reload" im Kommando

```
# /etc/init.d/exim4 reload
```

dazu, dass der exim4-Daemon ein Signal zum erneuten Einlesen der Konfigurationsdatei erhält.

2.4.3 Anpassen des Bootvorgangs

Debian verwendet kein BSD typisches `rc.local` Verzeichnis, um den Bootvorgang anzupassen; stattdessen wird folgender Mechanismus angeboten.

Angenommen `foo` sei ein Skript, das während des Startvorgangs oder beim Übergang in ein bestimmtes (System V) Runlevel aufgerufen werden soll. Dann sollte der Systemadministrator:

1 Das Skript `foo` in das Verzeichnis `/etc/init.d/` verschieben.

2 Das Debian-Kommando `update-rc.d` mit entsprechenden Argumenten starten, um Links zwischen den (Kommandozeilen spezifischen) Verzeichnissen `rc?.d` und `/etc/init.d/foo` anzulegen. Dabei bezeichnet *?* eine Nummer von 0 bis 6, die einem der System V Runlevel entspricht.

3 Das System neu booten.

Das Kommando `update-rc.d` setzt Links zwischen Dateien im Verzeichnis `rc?.d` und dem Skript in `/etc/init.d/`. Jeder Link beginnt mit einem 'S' oder 'K', gefolgt von einer Nummer, gefolgt vom Namen des Skripts. Beim Wechsel in das Runlevel *N*, werden Skripte in `/etc/rcN.d/` die mit 'K' beginnen mit `stop` als Argument ausgeführt, gefolgt von den mit 'S' beginnenden Skripten in `/etc/rcN.d/` mit `start` als Argument.

Man kann z.B. das Skript `foo` beim Booten ausführen lassen, indem man es nach `/etc/init.d/` verschiebt und die Links mit `update-rc.d foo defaults 19` erstellt. Das Argument `defaults` bezieht sich auf das Standard-Runlevel, welches zwischen 2 und 5 liegt. Das Argument `19` sichert, dass `foo` vor allen Skripten, welche die Nummern 20 oder größer enthalten, gestartet wird.

2.5 Unterstützung von Vielfalten

Debian unterstützt verschiedene Möglichkeiten zum Anpassen des Systems, ohne das System zu beeinträchtigen.

- `dpkg-divert`, vergleiche 'Der `dpkg-divert`-Befehl' auf Seite 81.
- `equivs`, vergleiche 'Das `equivs`-Paket' auf Seite 81.
- `update-alternative`, vergleiche 'Alternative Befehle' auf Seite 82.
- `make-kpkg` kann viele Boot-Loader anpassen. Vergleiche `make-kpkg(1)` und 'Die Debian-Standardmethode' auf Seite 86.

Alle Dateien unter `/usr/local/` gehören dem Systemadministrator und Debian wird sie nicht verändern. Viele (oder alle) Dateien unter `/etc` sind Konfigurationsdateien und Debian wird sie nicht während eines Upgrades überschreiben, es sei denn der Systemadministrator erlaubt dies ausdrücklich.

2.6 Internationalisierung

Das Debian-System ist internationalisiert und bietet Unterstützung zur Ein- und Ausgabe von Zeichen in vielen Sprachen, beides in der Konsole und unter X. Viele Texte, Handbuchseiten und Systemausgaben wurden in eine ständig wachsende Anzahl von Sprachen übersetzt.

Während der Installation fordert Debian den Nutzer zur Wahl der Installationssprache (und manchmal eines lokalen Dialekts) auf.

Sollte das installierte System nicht alle benötigten Eigenschaften der Sprache unterstützen, soll eine andere Sprache gewählt werden oder wurde eine neue Tastatur angeschlossen, um die Sprache zu unterstützen, vergleiche 'Lokalisation und Sprachen' auf Seite 154.

2.7 Debian und der Kernel

Vergleiche 'Der Linux-Kernel unter Debian' auf Seite 85.

2.7.1 Kompilierung eines Kernel aus Debian-fremden Quellen

Man sollte die Debian-Politik bezüglich der Header verstanden haben, bevor man startet.

Die Debian-C-Bibliotheken wurden mit der aktuellsten **stabilen** Version der **Kernelheader** erstellt.

Zum Beispiel nutzte die Debian 1.2 Ausgabe Version 5.4.13 der Header. Dieses Vorgehen unterscheidet sich von den Linux-Kernelquellpaketen, die auf allen Linux-FTP-Archiv-Seiten verbreitet werden, welche aktuellere Versionen der Header verwenden. Die Kernelheader aus den Kernelquellen befinden sich in `/usr/include/linux/include/`.

Wenn es nötig ist, ein Programm mit aktuelleren Kernelheadern als in `libc6-dev` zu übersetzen, so muss `-I/usr/src/linux/include/` zur Kommandozeile beim Kompilieren hinzugefügt werden. Dies ist z.B. für das Paketieren des automounter-Daemon (`amd`) von Bedeutung. Als neue Kernel einige NFS-bezogene Internals änderten, musste dies `amd` mitgeteilt werden. Dies erforderte die Einbindung der aktuellsten Kernelheader.

2.7.2 Tools zum Erzeugen angepasster Kernel

Nutzern die einen angepassten Kernel erzeugen wollen (oder müssen), wird empfohlen, das Paket `kernel-package` herunterzuladen. Dieses Paket enthält das Skript zur Kernelerstellung und bietet die Möglichkeit, ein Debian kernel-image Paket einfach durch Aufruf von

```
# make-kpkg kernel_image
```

im Kernelquellverzeichnis zu starten. Hilfe ist durch Ausführung von

```
# make-kpkg --help
```

verfügbar und durch die Handbuchseite `make-kpkg(1)` sowie 'Der Linux-Kernel unter Debian' auf Seite 85.

Nutzer müssen den Quellcode für den aktuellsten Kernel (oder den Kernel ihrer Wahl) separat vom bevorzugten Linux-Archiv herunterladen, wenn kein `kernel-source-`*`version`*-Paket (dabei steht *version* für die Kernel-Version) vorhanden ist. Das Debian `initrd`-Bootskript erfordert einen speziellen Kernel-Patch namens `initrd`; vergleiche `http://bugs.debian.org/149236`.

Detaillierte Anweisungen zur Benutzung des `kernel-package`-Pakets sind in der Datei `/usr/share/doc/kernel-package/README.gz` zu finden.

2.7.3 Spezielle Regeln für den Umgang mit Modulen

Debians `modconf` Paket enthält ein Shellskript (`/usr/sbin/modconf`), dass zur Anpassung der Modulkonfiguration genutzt werden kann. Dieses Skript besitzt eine Menü basierte Schnittstelle, die den Nutzer nach Einzelheiten über ladbare Gerätetreiber im System fragt. Die Antworten werden benutzt, um die Datei `/etc/modules.conf` (welche Aliase enthält sowie andere Argumente, die in Verbindung mit verschiedenen Modulen verwendet werden müssen) anzupassen, auf Grund von Dateien in `/etc/modutils/` und `/etc/modules` (die die Module auflistet, die zur Bootzeit geladen werden müssen).

Wie die (neuen) `Configure.help`-Dateien, die nun verfügbar sind, um angepasste Kernel zu unterstützen, kommt das `modconf`-Paket mit einer Reihe von Hilfe-Dateien (in `/usr/share/modconf/`), die detaillierte Informationen über passende Argumente für jedes Modul enthalten. Vergleiche 'Der modularisierte Kernel 2.4' auf Seite 87 für Beispiele.

2.7.4 Deinstallation eines alten Kernel-Pakets

Das `kernel-image-`*`NNN`*`.prerm`-Skript überprüft, ob der aktuell laufende Kernel mit dem zu entfernenden Kernel übereinstimmt. Deshalb können ungewünschte kernel-image Pakete sicher mittels

```
# dpkg --purge --force-remove-essential kernel-image-NNN
```

entfernt werden. (*NNN* ist durch die entsprechende Kernelversion und Revisionsnummer zu ersetzen.)

Kapitel 3

Debian-System-Installationshinweise

Offizielle Dokumentation zur Installation von Debian ist unter `http://www.debian.org/releases/stable/` und `http://www.debian.org/releases/stable/installmanual` zu finden.

Die Entwicklerversionen davon sind `http://www.debian.org/releases/testing/` und `http://www.debian.org/releases/testing/installmanual` (befinden sich in Arbeit und sind manchmal eventuell nicht vorhanden).

Obwohl dieses Kapitel ursprünglich für die Potato-Installation geschrieben wurde, ist der größte Teil des Inhalts auf das Woody-Installationsprogramm aktualisiert worden; beide sind sehr ähnlich. Da Sarge ein komplett anderes Installationsprogramm verwenden wird, sollten Sie dies nur als Anhaltspunkt für die Sarge-Installation verstehen. Einige Schlüsselpakete haben auch geänderte Namen und Prioritäten. Zum Beispiel ist der Standard-MTA von Sarge `exim4` anstatt `exim` und `coreutils` wurde eingeführt, um verschiedene Pakete zu ersetzen. Es ist eventuell nötig, einige Schritte anzupassen.

3.1 Allgemeine Linux-System-Installationshinweise

Vergessen Sie nicht unter `http://www.debian.org/CD/netinst/` zu schauen, wenn Sie nach einem kompakten CD-Image des Debian-Installers suchen.

Die Verwendung der `testing`- oder `unstable`-Distribution führt zum erhöhten Risiko, auf verschiedene Fehler zu stoßen. Dieses Risiko kann durch ein Multi-Boot-Schema gemanagt werden, bei dem eine stabilere Debian-Distribution genutzt wird. Alternativ kann auch ein netter Trick mit `chroot`, wie in '`chroot`' auf Seite 119 beschrieben, genutzt werden. Letzteres ermöglicht die Nutzung verschiedener Debian-Distributionen simultan auf verschiedenen Konsolen.

3.1.1 Grundlegendes zur Hardware-Kompatibilität

Linux ist zur meisten PC-Hardware kompatibel und kann auf fast jedem Rechner installiert werden. Es war für mich so einfach wie eine Windows-95/98/Me-Installation. Die Liste unterstützter Hardware wächst beständig.

Sollten Sie einen Laptop besitzen, schauen Sie unter Linux auf Laptops (`http://www.linux-laptop.net/`) für Hinweise zur Installation zu verschiedenen Marken und Modellen.

Meine Empfehlung zur Desktop-PC-Hardware ist "sei konservativ":

- SCSI statt IDE für die Arbeit, IDE/ATAPI-HD für private Nutzung.
- IDE/ATAPI-CD-ROM (oder -CD-RW).
- PCI statt ISA, insbesondere für die Netzwerkkarte (NIC).
- Verwende eine billige NIC. Tulip für PCI und NE2000 für ISA sind geeignet.
- Vermeide PCMCIA-Karten (Notebook) für die erste Linux-Installation.
- Keine USB-Tastatur, -Maus, … es sei denn Sie lieben die Herausforderung.

Sollten Sie einen langsamen Rechner haben, so ist das vorübergehende Entfernen und Einbauen der Festplatte in einen schnelleren Rechner für die Installation eine gute Idee.

3.1.2 Bestimmung der PC-Hardware und des Chipsatzes

Während der Installation wird man nach der eingebauten Hardware oder dem Chipsatz des PCs gefragt. Manchmal sind diese Informationen nicht leicht zu finden. Eine Möglichkeit ist:

1 Öffnen des PC-Gehäuses.
2 Bestimmen der Aufdrucke auf den großen Chips auf der Graphik- und Netzwerkkarte, sowie den Chips nahe den seriellen und IDE-Anschlüssen.
3 Bestimmen der Kartennamen, die auf der Rückseite der PCI- und ISA-Karten aufgedruckt sind.

3.1.3 Bestimmung der PC-Hardware mit Debian

Die folgenden Kommandos eines Linux-Systems sollten Aufschluss über aktuelle Hardware und deren Konfiguration geben.

```
$ pager /proc/pci
$ pager /proc/interrupts
$ pager /proc/ioports
$ pager /proc/bus/usb/devices
```

Diese Kommandos können während der Installation in der Konsole nach Drücken von Alt-F2 gestartet werden.

Nach der Basisinstallation können optionale Pakete wie `pciutils`, `usbutils` und `lshw` installiert werden, um erweiterte Informationen zum System zu erhalten.

```
$ lspci -v | pager
$ lsusb -v | pager
# lshw | pager
```

Typische Verwendung von Interrupts:

- IRQ0: Zeitausgabe (8254)
- IRQ1: Tastatur-Controller
- IRQ2: Kaskadierung zu IRQ8–IRQ15 bei PC-AT
- IRQ3: zweiter serieller Anschluss (io-port=0x2F8) (`/dev/ttyS1`)
- IRQ4: primärer serieller Anschluss (io-port=0x3F8) (`/dev/ttyS0`)
- IRQ5: frei [Soundkarte (SB16: io-port=0x220, DMA-low=1, DMA-high=5)]
- IRQ6: Diskettenlaufwerks-Controller (io-port=0x3F0) (`/dev/fd0`, `/dev/fd1`)
- IRQ7: paralleler Anschluss (io-port=0x378) (`/dev/lp0`)
- IRQ8: Echtzeituhr (rtc)
- IRQ9: Software-Interrupt (int 0x0A), Umleitung auf IRQ2
- IRQ10: frei [Netzwerkkarte (NE2000: io-port=0x300)]
- IRQ11: frei [(SB16-SCSI: io-port=0x340, SB16-IDE: io-port=0x1E8,0x3EE)]
- IRQ12: PS/2-Maus
- IRQ13: frei (war mathematischer Koprozessor 80287)
- IRQ14: primärer IDE-Controller (`/dev/hda`, `/dev/hdb`)
- IRQ15: sekundärer IDE-Controller (`/dev/hdc`, `/dev/hdd`)

Für alte nicht-PnP-ISA-Karten sollten Sie eventuell IRQ5, IRQ10 und IRQ11 auf "non-PnP" im BIOS einstellen.

Für USB-Geräte werden die Geräteklassen in `/proc/bus/usb/devices` mit `Cls=`*nn* aufgeführt:

- Cls=00 : nicht benutzt
- Cls=01 : Audio (Lautsprecher, etc.)
- Cls=02 : Kommunikation (Modem, Netzwerkkarte, . . .)
- Cls=03 : HID (Human Interface Device (Schnittstellen): Tastatur, Maus, Joystick)
- Cls=07 : Drucker
- Cls=08 : Massenspeicher (Diskettenlaufwerk, CD/DVD Laufwerk, Festplatte, Flash Speicher, . . .)
- Cls=09 : Hub (USB Hub)
- Cls=255 : Hersteller spezifisch

Ist die Geräteklasse eines Gerätes nicht 255, so wird es von Linux unterstützt.

3.1.4 Bestimmen der PC-Hardware mit anderen Betriebssystemen

Hardware-Informationen können auch von anderen Betriebssystemen erhalten werden:

Man kann eine andere kommerzielle Linux-Distribution installieren. Die Hardwareerkennung ist bei diesen oft besser, als dies bei Debian zurzeit der Fall ist. (Diese Situation sollte ausgewogener sein, wenn `debian-installer` mit Sarge eingeführt wird.)

In einem Windows-System kann die Hardware-Konfiguration durch Rechts-Klick auf "Mein Computer" unter Eigenschaften / Gerätemanager erhalten werden. Die Ressourcen wie IRQ, I/O-Portadressen sowie DMA sollten festgehalten werden. Einige alte ISA-Karten müssen eventuell unter DOS konfiguriert werden.

3.1.5 Ein Lilo-Mythos

“Lilo ist auf 1024 Zylinder beschränkt.” Falsch!

Das neuere `lilo` das nach Debian-Potato verwendet wird, bietet Unterstützung für lba32. Ist das BIOS des Motherboards aktuell genug um lba32 zu unterstützen, so sollte `lilo` in der Lage sein, auch von außerhalb der alten 1024-Zylindergrenze zu booten.

Es muss nur sichergestellt werden, dass die Zeile “lba32” irgendwo am Anfang der `lilo.conf`-Datei hinzugefügt wird, wenn noch eine alte `lilo.conf` vorhanden ist. Man vergleiche `/usr/share/doc/lilo/Manual.txt.gz`.

3.1.6 GRUB

Der neue Bootloader `grub` des GNU-Hurd-Projekts kann auf einem Debian-Woody-System wie folgt installiert werden:

```
# apt-get update
# apt-get install grub-doc
# mc /usr/share/doc/grub-doc/html/
... lesen Sie den Inhalt
# apt-get install grub
# pager /usr/share/doc/grub/README.Debian.gz
... lesen Sie dies :)
```

Um das GRUB-Menü anzupassen, ist `/boot/grub/menu.lst` zu editieren. Man vergleiche ‘Setzen von GRUB-Boot-Parametern’ auf Seite 98 zum Setzen von Bootparametern während des Bootvorgangs, da sich dies leicht von der `lilo`-Konfiguration unterscheidet.

3.1.7 Wahl der Boot-Disketten

In Potato nutzte ich IDEPCI für die normale Desktopinstallation. In Woody bevorzuge ich bf2.4. Beide benutzen eine Version von `boot-floppies` zum Erzeugen von Boot-Disketten.

Besitzen Sie eine PCMCIA-Netzwerkkarte, so müssen Sie die Standard-Bootkonfiguration nutzen (dies enthält die größte Anzahl von Disketten und alle Treibermodule) und die Netzwerkkarte im PCMCIA-Menü konfigurieren; man darf sie nicht im Standard-Netzwerk-Dialog einrichten.

Für spezielle Systeme kann es notwendig sein, eine angepasste Rettungsdiskette zu erzeugen. Dies kann durch Ersetzen des Kernelimages namens “linux” auf der Debian-Rettungsdiskette, durch ein anderes komprimiertes Kernelimage, das speziell für den Rechner erzeugt wurde, erreicht werden. Details sind in `readme.txt` auf der Rettungsdiskette zu finden. Diese nutzt das MS-DOS-Dateisystem, so dass ein beliebiges System zum Auslesen und Editieren verwendet werden kann. Dies sollte für Personen mit spezieller Netzwerkkarte, etc. leicht zu bewältigen sein.

In Sarge wird wohl `debian-installer` und/oder `pgi` zur Erzeugung von Bootdisketten verwendet werden.

3.1.8 Installation

Folgen Sie den offiziellen Anweisungen, die unter `http://www.debian.org/releases/stable/installmanual` oder `http://www.debian.org/releases/testing/installmanual` (befindet sich in Arbeit und ist manchmal eventuell nicht vorhanden) gefunden werden können.

Sollten Sie ein System mittels `boot-floppies` aus der `testing` Distribution installieren, so kann es sein, dass Sie während der Installation eine Konsole mittels Alt-F2 öffnen müssen, um in der Datei `/etc/apt/sources.list` manuell Einträge von "stable" nach "testing" zu ändern, um die APT-Quellen anzupassen.

Ich tendiere dazu, `lilo` in Partitionen wie `/dev/hda3` zu installieren, während `mbr` in `/dev/hda` installiert wird. Dies minimiert das Risiko, Bootinformationen zu überschreiben.

Während des Installationsvorgangs wähle ich folgende Werte.

- MD5-Passwörter "yes"
- shadow-Passwörter "yes"
- Install "advanced" (dselect **) und select
  - Abwahl von emacs (falls gewählt), nvi, tex, telnet, talk(d);
  - Auswahl von mc, vim und entweder nano-tiny oder elvis-tiny. Vergleiche '`dselect`' auf Seite 65. Selbst wenn Sie ein Emacs-Anhänger sind, vermeiden Sie es zunächst und begnügen Sie sich mit nano während der Installation. Man sollte auch andere große Pakete, so wie TeX (Potato wählte dies aus) zu diesem Moment noch nicht installieren. Vergleiche 'Rettungseditoren' auf Seite 195 für den Grund, nano-tiny oder elvis-tiny jetzt zu installieren.
- Alle Konfigurationsfragen während der Installationsdialoge der einzelnen Pakete mit "y" beantworten (ersetzt aktuellen Wert).
- `exim`: wähle 2 für Rechner, da ich meine E-Mails durch den SMTP-Server meines Internetproviders (ISP) verschicke.

Für weitere Informationen über dselect vergleiche '`dselect`' auf Seite 65.

3.1.9 Hosts und IP im LAN

Beispiel einer LAN-Konfiguration (Klasse-C-Subnet: 192.168.1.0/24):

```
Internet
```

```
      |
      +--- Externer ISP für POP-Service (Zugriff mittels fetchmail)
      |
     Access point ISP bietet DHCP-Service und SMTP relay service an
      |                      :
     Kabelmodem          (Dialup)
      |                      :
   LAN-Gateway-Maschine, externer Port: eth0 (IP durch ISP's DHCP)
    altes Notebook (IBM Thinkpad, 486 DX2 50MHz, 20MB RAM)
    Linux 2.4 Kernel mit ext3 Dateisystem.
    "ipmasq" Paket (mit umfangreichem Patch, NAT und Firewall)
    "dhcp-client" Paket, konfiguriert für eth0 (überschreibt DNS-Werte)
    "dhcp" Paket konfiguriert für eth1
    starte "exim" als smarthost (Mode 2)
    starte "fetchmail" in großen Intervallen (zur Sicherheit)
    starte "bind" als Cache-Nameserver fürs Internet vom LAN
                  als Nameserver fürs LAN-Netzwerk vom LAN
    starte "ssh" auf Port 22 und 8080 (Verbindung von überall)
    starte "squid" als Cache-Server für das Debianarchiv (für APT)
   LAN-Gateway-Maschine, interner Port: eth1 (IP = 192.168.1.1, fest)
                             |
            +--- LAN-Switch (100base T) ---+
            |                              |
   Einige feste IP-Clients im LAN Einige DHCP-Clients im LAN
   (IP = 192.168.1.2-127, fest)   (IP = 192.168.1.128-200, dynamisch)
```

Man vergleiche 'Netzwerk-Konfiguration' auf Seite 163 für die Details zur Konfiguration des Netzwerks und 'Aufbau eines Gateway-Routers' auf Seite 190 für die Details zur Konfiguration des LAN-Gateway-Servers.

3.1.10 Benutzerkonten

Um ein konsistentes Verhalten zwischen verschiedenen Systemen zu gewährleisten, sind die ersten paar Benutzerkonten in meinem System stets gleich.

Ich erzeuge immer zuerst ein normales Benutzerkonto mit einem Namen wie "admin" (uid=1000) und leite die E-Mails an root an dies weiter. Diesem Konto wird die Gruppe `adm` zugeordnet (siehe "'Warum GNU `su` nicht die `wheel` Gruppe unterstützt"' auf Seite 128) und es können ihm einige root-Privilegien mit `su` (PAM-benutzend) oder mit `sudo` zugeordnet werden. Vergleiche 'Hinzufügen eines neuen Nutzers' auf Seite 48 für Details.

3.1.11 Anlegen von Dateisystemen

Partitionierung der Festplatte

Ich bevorzuge die Verwendung verschiedener Partitionen für verschiedene Verzeichnisbäume, um Schäden bei Systemabstürzen vorzubeugen. Ein Beispiel dafür ist:

```
/          == (/ + /boot + /bin + /sbin)
           == 50MB+
/tmp       == 100MB+
/var       == 100MB+
/home      == 100MB+
/usr       == 700MB+ mit X
/usr/local == 100MB
```

Die Größe des `/usr`-Verzeichnis hängt sehr von installierten X-Windows-Anwendungen und der Dokumentation ab. `/usr/` kann rund 300MB groß sein, wenn man nur mit der Konsole arbeitet, wobei 2GB–3GB keine unübliche Größe ist, wenn man viele Gnome-Anwendungen installiert hat. Wird `/usr/` zu groß, so ist das Verschieben von `/usr/share/` auf eine andere Partition die effektivste Lösung. Mit den neuen großen vorgefertigten Linux 2.4er-Kernel, kann `/` mehr als 200MB benötigen.

Zum Beispiel ist der aktuelle Status meines Internet-Gateway-Rechners wie folgt (Ausgabe des `df -h` Kommandos):

```
Filesystem          Größe Benut  Verf Ben% Eingehängt auf
/dev/hda3            300M  106M  179M  38% /
/dev/hda7            100M   12M   82M  13% /home
/dev/hda8            596M   53M  513M  10% /var
/dev/hda6            100M  834k   94M   1% /var/lib/cvs
/dev/hda9            596M  222M  343M  40% /usr
/dev/hda10           596M  130M  436M  23% /var/cache/apt/archives
/dev/hda11           1.5G  204M  1.2G  14% /var/spool/squid
```

(Der große reservierte Bereich für `/var/spool/squid/` ist für einen Proxy-Cache für den Paketdownload.)

Es folgt die `fdisk -l` Ausgabe zur Veranschaulichung einer möglichen Partitionierung:

```
# fdisk -l /dev/hda # Kommentar

/dev/hda1          1      41    309928+   6  FAT16 # DOS
/dev/hda2         42      84    325080   83  Linux # nicht benutzt
/dev/hda3  *      85     126    317520   83  Linux # root Partition
/dev/hda4        127     629   3802680    5  Extended
```

```
/dev/hda5       127     143    128488+  82  Linux swap
/dev/hda6       144     157    105808+  83  Linux
/dev/hda7       158     171    105808+  83  Linux
/dev/hda8       172     253    619888+  83  Linux
/dev/hda9       254     335    619888+  83  Linux
/dev/hda10      336     417    619888+  83  Linux
/dev/hda11      418     629   1602688+  83  Linux
```

Einige nicht benutzte Partitionen existieren. Diese sind für die Installation einer zweiten Linux-Distribution oder als Erweiterung für wachsende Verzeichnisbäume vorgesehen.

Mounten von Dateisystemen

Das korrekte Mounten der obigen Dateisysteme wird durch folgende `/etc/fstab` erreicht:

```
# /etc/fstab: statische Dateisystem-Informationen.
#
# Dateisystem Mountverzeichnis Typ  Optionen     dump Durchgang
/dev/hda3     /                ext2  defaults,errors=remount-ro 0 1
/dev/hda5     none             swap  sw                     0 0
proc          /proc            proc  defaults               0 0
/dev/fd0      /floppy          auto  defaults,user,noauto   0 0
/dev/cdrom    /cdrom        iso9660  defaults,ro,user,noauto 0 0
#
# einige separate Partitionen
/dev/hda7     /home            ext2  defaults               0 2
/dev/hda8     /var             ext2  defaults               0 2
/dev/hda6     /var/lib/cvs     ext2  defaults               0 2
# noatime beschleunigt den Dateizugriff beim Lesen
/dev/hda9     /usr             ext2  defaults,noatime       0 2
/dev/hda10    /var/cache/apt/archives ext2 defaults         0 2

# sehr große Partition für Proxy-Cache
/dev/hda11    /var/spool/squid ext2 rw                      0 2

# DOS-Backup
/dev/hda1     /mnt/dos         vfat  rw,noauto              0 0
# Backup des Linux Systems
/dev/hda2     /mnt/linux       ext2  rw,noauto              0 0
#
# nfs-mounts
mickey:/      /mnt/mickey      nfs   ro,noauto,intr         0 0
goofy:/       /mnt/goofy       nfs   ro,noauto,intr         0 0
# minnie:/ /mnt/minnie smbfs ro,soft,intr,credentials={filename} 0 2
```

Für NFS wird `noauto,intr` kombiniert mit der standardmäßigen `hard`-Option benutzt. Damit ist es möglich, aufgrund abgebrochener Verbindung hängende Prozesse mit Strg-C zu beenden.

Für einen mit Samba (smbfs) angebundenen Windows Rechner ist `rw,auto,soft,intr` eine gute Option. Siehe auch 'Samba-Konfiguration' auf Seite 41.

Für ein Diskettenlaufwerk ist `noauto,rw,sync,user,exec` zu empfehlen, da dies Beschädigungen des Dateisystems durch einem unbeabsichtigten Diskettenauswurf vor einem umount-Aufruf verhindert, allerdings verlangsamt dies auch den Schreibzugriff.

Autofs-mount

Wichtige Punkte zu auto-mount:

- Laden Sie das `vfat`-Modul, so dass `/etc/auto.misc` die Zeile `-fstype=auto` enthalten kann:
  ```
  # modprobe vfat # vor dem Zugriff auf die Diskette
   ... oder um dies zu automatisieren
  # echo "vfat" >> /etc/modules
   ... jetzt das System neu starten.
  ```
- Die Datei `/etc/auto.misc` ist wie folgt anzupassen:
  ```
  floppy -fstype=auto,sync,nodev,nosuid,gid=100,umask=000 :/dev/fd0
   ... wobei gid=100 für "users" steht.
  ```
- Erzeugen Sie nun die Links `cdrom` und `floppy` in `/home/`*`user`*, die auf `/var/autofs/misc/cdrom` bzw. `/var/autofs/misc/floppy` verweisen.
- Nun ist *user* der "users"-Gruppe hinzuzufügen.

NFS-mount

Der externe Linux-NFS-Server (goofy) befindet sich hinter einer Firewall (gateway). Ich habe eine sehr lockere Sicherheitspolitik in meinem LAN, da ich der einzige Nutzer bin. Zum Aktivieren des NFS-Zugangs muss auf der NFS-Serverseite die Datei `/etc/exports` erzeugt werden:

```
# /etc/exports: die Zugriffskontrollliste für Dateisysteme, auf die
#               NFS-Clients zugreifen sollen. Siehe exports(5).
/       (rw,no_root_squash)
```

Dies ist nötig, um den NFS-Server zu konfigurieren, nachdem die NFS-Server- und Client-Pakete installiert und aktiviert wurden.

Der Einfachheit halber erzeuge ich üblicherweise eine einzige 2GB Partition für eine experimentelle oder schnelle zweite Linuxinstallation. Ich teile wahlweise die swap- und `/tmp`-Partitionen für diese Installationen. Ein aus mehreren Partitionen bestehendes System ist für diese Verwendung zu kompliziert. Wenn nur ein einfaches Konsolen-System benötigt wird, so sind 500MB mehr als ausreichend.

3.1.12 DRAM-Speicher-Richtlinien

Es folgen einige grobe Anhaltspunkte für DRAM.

```
  4MB:  Absolutes Minimum für den Linux-Kernel.
 16MB:  Minimum für ein angemessenes Konsolen-System.
 32MB:  Minimum für ein einfaches X-System.
 64MB:  Minimum für ein X-System mit GNOME/KDE.
128MB:  Empfohlen für komfortables Arbeiten mit X und GNOME/KDE.
256MB (oder mehr): Warum nicht, wenn man es sich leisten kann?
        DRAM ist billig.
```

Verwendung der Bootoption `mem=4m` (oder in lilo.conf `append="mem=4m"`) zeigt, wie sich ein System mit 4MB Speicher verhält. Ein lilo-Bootparameter wird für Systeme mit mehr als 64MB Speicher und einem alten BIOS benötigt.

3.1.13 Swap-Speicher

Ich verwende die folgenden Richtlinien für Swap:

- Jede Swap-Partition ist < 128MB (bei Verwendung alter 2.0 Kernel), bzw. < 2GB (bei Verwendung aktueller Kernel)
- Insgesamt = entweder das 1- bis 2-fache des installierten RAM oder 128MB bis 2GB wird empfohlen
- Die Swap-Partitionen sind auf verschiedene Festplatten zu verteilen und mit den Optionen `sw,pri=1` in `/etc/fstab` zu mounten. Dies stellt sicher, dass der Kernel ein "striping RAID" der Swap-Partitionen nutzt und einen maximalen Durchsatz erreicht.
- Nach Möglichkeit sollte der mittlere Bereich der Festplatte verwendet werden.

Selbst wenn er niemals benötigt wird, ist etwas Swap-Speicher (128MB) empfehlenswert, da das System langsamer wird, bevor es wegen Speichermangel abstürzt.

3.2 Bash-Konfiguration

Ich passe folgende Shell-Startskripte an meine Gewohnheiten an:

```
/etc/bash.bashrc        wird durch eigene Version ersetzt
/etc/profile            wird beibehalten (bis auf $PS1: \w -> \W)
/etc/skel/.bashrc       wird durch eigene Version ersetzt
/etc/skel/.profile      wird durch eigene Version ersetzt
/etc/skel/.bash_profile wird durch eigene Version ersetzt
~/.bashrc               wird für alle Nutzer ersetzt
~/.profile              wird für alle Nutzer ersetzt
~/.bash_profile         wird für alle Nutzer ersetzt
```

Vergleiche Details dazu in meinen Beispielskripten (http://www.debian.org/doc/manuals/debian-reference/examples/). Ich bevorzuge ein transparentes System, deshalb setze ich umask auf 002 oder 022.

PATH wird durch die folgenden Konfigurationsdateien in der Reihenfolge

```
/etc/login.defs  - bevor die Shell PATH setzt
/etc/profile     (startet eventuell /etc/bash.bashrc)
~/.bash_profile  (startet eventuell ~/.bashrc)
```

gesetzt.

3.3 Maus-Konfiguration

3.3.1 PS/2-Mäuse

Im Falle einer Maus mit einem PS/2-Anschluss am ATX-Motherboard, sollte der Signalfluss wie folgt sein:

```
Maus -> /dev/psaux -> gpm -> /dev/gpmdata = /dev/mouse -> X
```

Hier wurde ein symbolischer Link /dev/mouse verwendet, der auf /dev/gpmdata verweist, um einige Konfigurationsprogramme zufriedenzustellen und die erneute Konfiguration zu vereinfachen. (Entscheidet man sich beispielsweise dazu, den gpm-Daemon nicht zu nutzen, so muss der symbolische Link /dev/mouse auf /dev/psaux geändert werden, nachdem gpm entfernt wurde.)

Dieser Signalfluss erlaubt, die Tastatur und die Maus zu entfernen und beim Anstecken mit gpm neu zu initialisieren. X wird davon nicht negativ beeinflusst!

Das Protokoll des Signalflusses zwischen gpm-Ausgabe und X-Eingabe kann auf zwei Arten implementiert werden, als spezielles Protokoll wie z.B. “ms3” (serielles Microsoft-3-Tasten-Mausprotokoll) oder “raw” (nutzt das selbe Protokoll wie die angeschlossene Maus) und diese Wahl diktiert die Wahl des in der X-Konfiguration benutzten Protokolls.

Ich werde die Konfiguration anhand einer 3-Tasten-Logitech (traditionelle Unix-artige) -PS/2-Maus demonstrieren und im folgenden Beispiel beschreiben.

Sind Sie einer derjenigen, dessen Graphikkarte nicht durch den neuen X-Server Version 4, sondern nur durch Version 3 unterstützt wird (einige ATI-64-Bit-Karten), so muss /etc/X11/XF86Config anstatt von /etc/X11/XF86Config-4 im Folgenden konfiguriert werden.

Der ms3-Protokoll-Ansatz

```
/etc/gpm.conf            | /etc/X11/XF86Config-4
=========================+======================================
device=/dev/psaux        | Section "InputDevice"
responsiveness=          |  Identifier "Konfigurierte Maus"
repeat_type=ms3          |  Driver     "mouse"
type=autops2             |  Option     "CorePointer"
append=""                |  Option     "Device"   "/dev/gpmdata"
sample_rate=             |  Option     "Protocol" "IntelliMouse"
                         | EndSection
```

Wenn diese Methode gewählt wird, so erfolgt die Anpassung des Maustyps allein in der Datei `gpm.conf` und die X-Konfiguration bleibt gleich. Man vergleiche meine Beispielskripte (`http://www.debian.org/doc/manuals/debian-reference/examples/`).

Der raw-Protokoll-Ansatz

```
/etc/gpm.conf            | /etc/X11/XF86Config-4
=========================+======================================
device=/dev/psaux        | Section "InputDevice"
responsiveness=          |  Identifier "Configured Mouse"
repeat_type=raw          |  Driver     "mouse"
type=autops2             |  Option     "CorePointer"
append=""                |  Option     "Device"   "/dev/mouse"
sample_rate=             |  Option     "Protocol" "MouseManPlusPS/2"
                         | EndSection
```

Wenn diese Methode gewählt wird, so erfolgt die Anpassung des Maustyps in `gpm.conf` sowie in der X-Konfiguration.

Wie man andere Mäuse einrichtet

Der `gpm`-Gerätetyp `autops2` sollte die meisten PS/2-Mäuse erkennen. Unglücklicherweise funktioniert dies nicht immer und ist in vor-Woody-Versionen nicht verfügbar. In diesen Fällen sollte man `ps2` oder `imps2` in `gpm.conf` anstatt von `autops2` probieren. Um die von `gpm` unterstützten Mäuse zu bestimmen, kann man `gpm -t help` starten. Man vergleiche `gpm(8)`.

Wird eine 2-Tasten-PS/2-Maus verwendet, so sollte das X-Protokoll die Option `Emulate3Buttons` nutzen. Der Unterschied des Protokolls zwischen der 2- und 3-Tasten-Maus wird automatisch erkannt und für `gpm` eingerichtet, nachdem die mittlere Maustaste einmalig gedrückt wurde.

Für das X-Protokoll mit 'Der raw-Protokoll-Ansatz' auf dieser Seite oder ohne `gpm` nutzt man:

- `IntelliMouse`: Maus mit seriellem Anschluss (`gpm` Repeater mit “ms3”)
- `PS/2`: PS/2-Anschluss-Maus (dies immer zuerst ausprobieren)
- `IMPS/2`: beliebige PS/2-Anschluss-Mäuse (2, 3 oder Scroll Mäuse, besser)
- `MouseManPlusPS/2`: Logitech PS/2-Anschluss-Maus
- `...`

Ausführlicheres ist unter Mouse Support in XFree86 (`http://www.xfree86.org/current/mouse.html`) zu finden.

Eine typische Microsoft Scroll-Maus funktioniert am besten mit:

```
/etc/gpm.conf             | /etc/X11/XF86Config-4
=========================+=======================================
device=/dev/psaux         | Section "InputDevice"
responsiveness=           |  Identifier "Konfigurierte Maus"
repeat_type=raw           |  Driver     "mouse"
type=autops2              |  Option     "CorePointer"
append=""                 |  Option     "Device"   "/dev/mouse"
sample_rate=              |  Option     "Protocol" "IMPS/2"
                          |  Option     "Buttons" "5"
                          |  Option     "ZAxisMapping" "4 5"
                          | EndSection
```

Für einige aktuelle dünne Toshiba-Notebooks, kann das Aktivieren von `gpm` vor PCMCIA im System-V-Init-Skript helfen, wenn das System sonst hängenbleibt. Eigenartig aber wahr.

3.3.2 USB-Mäuse

Stellen Sie sicher, dass Sie alle benötigten Kernel-Elemente einkompiliert oder als Modul vorliegen haben:

- Unter “Input core support”:
  - “Input core support” (CONFIG_INPUT, `input.o`),
  - “Mouse support” (CONFIG_INPUT_MOUSEDEV, `mousedev.o`),
- Unter “USB support”:
  - “Support for USB” (CONFIG_USB, `usbcore.o`),
  - “Preliminary USB device filesystem” (CONFIG_USB_DEVICEFS),
  - “UHCI” oder “OHCI” (CONFIG_USB_UHCI || CONFIG_USB_UHCI_ALT || CONFIG_USB_OHCI, `usb-uhci.o` || `uhci.o` || `usb-ohci.o`),
  - “USB Human Interface Device (full HID) support” (CONFIG_USB_HID, `hid.o`) und
  - “HID input layer support” (CONFIG_USB_HIDINPUT)

Die kleingeschriebenen Namen sind Modulnamen.

Wird devfs nicht genutzt, so ist eine Gerätedatei `/dev/input/mice` mit major 13 und minor 63 wie folgt anzulegen:

```
# cd /dev
```

```
# mkdir input
# mknod input/mice c 13 63
```

Für typische **USB**-Scroll-Mäuse, ist folgende Konfiguration zu empfehlen:

```
/etc/gpm.conf            | /etc/X11/XF86Config-4
=========================+=====================================
device=/dev/input/mice   | Section "InputDevice"
responsiveness=          |  Identifier "Konfigurierte Maus"
repeat_type=raw          |  Driver     "mouse"
type=autops2             |  Option     "SendCoreEvents" "true"
append=""                |  Option     "Device"   "/dev/input/mice"
sample_rate=             |  Option     "Protocol" "IMPS/2"
                         |  Option     "Buttons" "5"
                         |  Option     "ZAxisMapping" "4 5"
                         | EndSection
```

Man vergleiche das Linux USB Projekt (`http://www.linux-usb.org/`) für weitere Informationen.

3.3.3 Touchpad

Obwohl das Touchpad eines Laptops standardmäßig eine 2-Tasten-PS/2-Maus emuliert, ermöglicht das `tpconfig`-Paket die volle Kontrolle über das Gerät. Die Einstellung `OPTIONS="--tapmode=0"` in `/etc/default/tpconfig` deaktiviert z.B. das ärgerliche "klicken beim Antippen" Verhalten. Folgende `/etc/gpm.conf` ermöglicht das Nutzen des Touchpads und einer externen USB-Maus in der Konsole:

```
device=/dev/psaux
responsiveness=
repeat_type=ms3
type=autops2
append="-M -m /dev/input/mice -t autops2"
sample_rate=
```

3.4 NFS-Konfiguration

NFS wird durch `/etc/exports` eingerichtet:

```
# apt-get install nfs-kernel-server
# echo "/ *.domainname-für-lan-hosts(rw,no_root_squash,nohide)" \
        >> /etc/exports
```

Man vergleiche meine Beispielskripte für Details (`http://www.debian.org/doc/manuals/debian-reference/examples/`).

3.5 Samba-Konfiguration

Referenzen:

- `http://www.samba.org/`
- `samba-doc`-Paket

Das Einrichten von Samba im "share"-Mode ist viel einfacher, da dies die Laufwerke wie unter WfW anspricht. Aber es wird empfohlen, den "user"-Mode zu verwenden.

Samba kann durch `debconf` oder `vi` konfiguriert werden:

```
# dpkg-reconfigure --priority=low samba # in Woody
# vi /etc/samba/smb.conf
```

Man vergleiche meine Beispielskripte für Details (`http://www.debian.org/doc/manuals/debian-reference/examples/`).

Das Hinzufügen eines neuen Nutzers zur `smbpasswd`-Datei kann mit `smbpasswd` erfolgen:

```
$ su -c "smbpasswd -a Benutzername"
```

Man sollte sicherstellen, dass man verschlüsselte Passwörter für optimale Kompatibilität verwendet.

Der Wert `os level` ist entsprechend zu den folgenden Systemäquivalenzen zu setzen (je höher der Wert, um so größer ist die Priorität als Server):

```
0:      Samba mit geringer Priorität
1:      WfW 3.1, Win95, Win98, Win/Me?
16:     Win NT WS 3.51
17:     Win NT WS 4.0
32:     Win NT SVR 3.51
33:     Win NT SVR 4.0
255:    Samba mit hoher Priorität
```

Es muss sichergestellt werden, dass die Nutzer der Gruppe angehören, welche dem Verzeichnis zugeordnet ist, das den gemeinsamen Zugriff ermöglicht. Außerdem muss der Verzeichnispfad entsprechende Ausführungsrechte haben, um den Zugriff zu ermöglichen.

3.6 Drucker-Konfiguration

Die traditionelle Methode ist `lpr/lpd`. Es gibt allerdings ein neues System: CUPS™ (Common UNIX Printing System). PDQ ist ein anderer Ansatz. Man vergleiche Linux Printing HOWTO (`http://www.tldp.org/HOWTO/Printing-HOWTO.html`) für weitere Informationen.

3.6.1 **lpr/lpd**

Für die `lpr/lpd`-artigen Spooler (`lpr`, `lprng` und `gnulpr`) ist `/etc/printcap` wie folgt für PostScript- und Nur-Text-Drucker anzupassen (die Grundlagen):

```
lp|alias:\
        :sd=/var/spool/lpd/lp:\
        :mx#0:\
        :sh:\
        :lp=/dev/lp0:
```

Bedeutung der obigen Zeilen:
- Kopfzeile: *lp* – Name der Druckerwarteschlange, *alias* = alias
- mx#0 – maximale Dateigröße ist nicht limitiert
- sh – Unterdrücken eines Deckblatts
- lp=/dev/lp0 – lokale Druckerschnittstelle oder port@host für entfernte Drucker

Dies ist eine gute Konfiguration für einen PostScript-Drucker. Auch für das Drucken von einem Windows-Rechner mittels Samba ist dies eine geeignete Konfiguration für alle von Windows unterstützten Drucker (bidirektionale Kommunikation wird nicht unterstützt). Man muss die entsprechende Druckerkonfiguration auf dem Windows-Rechner nutzen.

Sollte kein PostScript-Drucker vorhanden sein, muss man einen Filter mittels `gs` aktivieren. Es gibt viele Konfigurationstools für das Einrichten von `/etc/printcap`. Jede der folgenden Möglichkeiten kommt in Frage:
- `gnulpr`, (`lpr-ppd`) und `printtool` – Ich nutze dies.
- `lpr` und `apsfilter`
- `lpr` und `magicfilter`
- `lprng` und `lprngtool`
- `lprng` und `apsfilter`
- `lprng` und `magicfilter`

Um graphische Konfigurationstools wie `printtool` zu verwenden, vergleiche man 'Administratorrechte unter X' auf Seite 146 um root-Rechte zu erhalten. Mit `printtool` erzeugte Druckerwarteschlangen nutzen `gs` und agieren wie PostScript-Drucker. Beim Zugriff darauf, müssen deshalb PostScript-Druckertreiber verwendet werden. Auf der Windows-Seite ist "Apple LaserWriter" der Standard.

3.6.2 CUPS™

Das Common UNIX Printing System (oder CUPS™) wird mittels `aptitude` durch Auswahl aller Pakete unter "Schnellauswahl (Tasks)" -> "Servers" -> "Druck-Server" installiert (Sarge). Für beste Ergebnisse sollte `aptitude` wie folgt konfiguriert sein: "F10" -> "Optionen" -> "Abhängigkeitsbehandlung" -> "[X] Empfohlene Pakete automatisch installieren".

KDE- und Gnome-Desktop-Umgebungen bieten eine einfache Drucker-Konfiguration. Alternativ kann das System mit einem beliebigen Web-Browser konfiguriert werden, wenn `swat` installiert ist:

```
$ meinBrowser http://localhost:631
```

Ein Drucker kann wie folgt hinzugefügt werden:

- klicken auf “Printers” auf der Hauptseite und danach “Add Printer”,
- Eingabe von “root” für den Nutzernamen und des entsprechenden Passworts,
- folgen Sie den weiteren Anweisungen zum Hinzufügen des Druckers,
- gehen Sie zurück zur “Printers” Seite und klicken Sie “Configure Printer” und
- fahren Sie fort, die Papiergröße, Auflösung und andere Parameter zu spezifizieren.

Weitere Informationen sind unter `http://localhost:631/documentation.html` und `http://www.cups.org/cups-help.html` zu finden.

3.7 CRON für Desktop-PCs

Der Vixie-`cron` wird standardmäßig für die Ausführung von Programmen zu bestimmten Zeiten installiert. Er funktioniert nicht gut, falls Ihr System nicht Tag und Nacht läuft. Für einen Desktop-PC, sollten Sie `anacron` statt `cron` installieren, um dies Problem zu umgehen. Das Paket `fcron` kann alternativ verwendet werden.

In ‘Terminplanung für Prozesse (`cron`, `at`)’ auf Seite 115 finden Sie Informationen zur Konfiguration von CRON-Jobs.

3.8 Andere Installationshinweise

3.8.1 Installation einiger weiterer Pakete nach der Basisinstallation

Sind Sie den bisherigen Anweisungen gefolgt, so haben Sie ein kleines aber funktionsfähiges Debian-System. Es ist nun ein guter Zeitpunkt, um größere Pakete zu installieren.

- Man starte `tasksel`. Vergleiche ‘*Aufgaben-Profile* mit `tasksel` oder `aptitude` installieren’ auf Seite 64.

  Man kann Folgendes bei Bedarf wählen:
  - End-user – X Window System
  - Development – C and C++
  - Development – Python
  - Development – Tcl/Tk
  - Miscellaneous – TeX/LaTeX environment

  - Ich nutze `tasksel` auch oft, um mir einen Überblick über die einzelnen Komponenten eines Menüpunktes mittels <Task Info> zu verschaffen, und diese einzeln mittels `dselect` zu installieren.

- Starten von `dselect`.

  Das erste was man jetzt wohl tun wird, ist den favorisierten Editor und alle Programme die man benötigt, auszuwählen. Man kann mehrere Emacs-Varianten gleichzeitig installieren. Man vergleiche '`dselect`' auf Seite 65 und 'Populäre Editoren' auf Seite 195.

  Man kann auch einige der Standardprogramme durch Programme mit größerer Funktionalität ersetzen.

  - ...

- ...

Ich editiere gewöhnlich `/etc/inittab` zum einfacheren Herunterfahren des Rechners.

```
...
# What to do when CTRL-ALT-DEL is pressed.
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -h now
...
```

3.8.2 Module

Module für Gerätetreiber werden während der Basisinstallation konfiguriert. `modconf` bietet danach eine menübasierte Modulkonfiguration. Dieses Programm ist sehr nützlich, wenn einige Module während der Basisinstallation ausgelassen wurden oder ein neuer Kernel seitdem installiert wurde.

Alle vorzuladenden Module müssen in `/etc/modules` aufgelistet werden. Ich nutze auch `lsmod` und `depmod` um dies manuell zu kontrollieren.

Man sollte auch sicherstellen, einige Zeilen zu `/etc/modules` hinzuzufügen, um IP-Masquerading (FTP, etc.) für 2.4 Kernel zu handhaben. Man vergleiche 'Der modularisierte Kernel 2.4' auf Seite 87, speziell 'Netzwerk-Funktionalität' auf Seite 89.

3.8.3 Grundlegende CD-RW-Einrichtung

Für am IDE-Anschluss angeschlossene CD-RW-Laufwerke mit einem 2.4er-Kernel editieren Sie die folgenden Dateien:

```
/etc/lilo.conf  (append="hdc=ide-scsi ignore=hdc" ist hinzuzufügen und
                 lilo zum Aktivieren zu starten)
/dev/cdrom      (symbolischer Link # cd /dev; ln -sf scd0 cdrom)
/etc/modules    (hinzufügen von "ide-scsi" und "sg", sowie "sr" danach wenn
                 nötig)
```

Man vergleiche 'CD-Brenner' auf Seite 131 für Details.

3.8.4 Großer Speicher und automatisches Abschalten

Editieren Sie /etc/lilo.conf wie folgt, um Bootparameter für großen Speicher (für 2.2 Kernel) und automatisches Abschalten (für APM) zu setzen:

```
append="mem=128M apm=on apm=power-off noapic"
```

Nun muss noch lilo zum Aktivieren dieser Einstellungen gestartet werden. apm=power-off wird für SMP-Kernel verwendet und noapic ist für meine fehlerhafte SMP-Hardware nötig. Das selbe kann direkt durch Angabe der Optionen beim Boot-Prompt erreicht werden. Man vergleiche 'Weitere Tricks mit dem Boot-Prompt' auf Seite 98.

Wurde APM als Modul kompiliert wie in den Standard-Debian-2.4er-Kerneln, so ist insmod apm power_off=1 nach dem Booten zu starten oder /etc/modules wie folgt zu modifizieren:

```
# echo "apm power_off=1" >>/etc/modules
```

Alternativ kann mit ACPI-Unterstützung das selbe mit neueren Kernel erreicht werden und dies scheint SMP freundlicher zu sein (erfordert ein neueres Motherboard). Der 2.4 Kernel auf neueren Motherboards sollte großen Speicher korrekt erkennen.

```
CONFIG_PM=y
CONFIG_ACPI=y
...
CONFIG_ACPI_BUSMGR=m
CONFIG_ACPI_SYS=m
```

auch die folgenden Zeilen sollten in /etc/modules in dieser Reihenfolge hinzugefügt werden:

```
ospm_busmgr
ospm_system
```

Es ist auch möglich, den Kernel mit all diesen auf "y" gesetzten Kerneloptionen zu kompilieren. Auf jeden Fall sind keine Bootparameter mit ACPI nötig.

3.8.5 Eigenartige Probleme beim Zugriff auf einige Webseiten

Aktuelle Linux-Kernel aktivieren standardmäßig ECN, was Zugriffsprobleme bei einigen Webseiten auf schlechten Routern zur Folge haben kann. Überprüfen des ECN-Status:

```
# cat /proc/sys/net/ipv4/tcp_ecn
 ... oder
# sysctl net.ipv4.tcp_ecn
```

Zum Ausschalten ist

```
# echo "0" > /proc/sys/net/ipv4/tcp_ecn
 ... oder
# sysctl -w net.ipv4.tcp_ecn=0
```

zu verwenden. Um TCP-ECN bei jedem Start zu deaktivieren, ist `/etc/sysctl.conf` zu editieren und

```
net.ipv4.tcp_ecn = 0
```

hinzuzufügen.

3.8.6 Dialup-PPP-Konfiguration

Das `pppconfig`-Paket ist zur Nutzung eines dialup- (Modem) PPP-Zugangs zu installieren.

```
# apt-get install pppconfig
# pppconfig
 ... folgen Sie den Anweisungen zur dialup-PPP-Konfiguration
# adduser user_name dip
 ... erlaube user_name Zugriff auf dialup-PPP
```

Dialup-PPP-Zugriff kann durch den Nutzer (*user_name*) mit

```
$ pon ISP_name  # starte PPP-Zugang zum ISP
 ... erfreuen Sie sich des Internets
$ poff ISP_name # beende PPP-Zugang, ISP_name optional
```

gestartet werden. Man vergleiche ‘Konfiguration der PPP-Schnittstelle’ auf Seite 168 für mehr Details.

3.8.7 Andere Konfigurationsdateien in `/etc/`

Da in der Standard-Debian-Installation `/etc/cron.deny` fehlt, sollte diese erzeugt werden (man kann `/etc/at.deny` kopieren).

Kapitel 4

Debian-Tutorials

Dieser Abschnitt bietet eine grobe Orientierung zur Linuxwelt für den Neueinsteiger. Sollten Sie Linux schon eine Weile nutzen, so können Sie Ihr Wissen testen.

4.1 Informationsquellen

Die ultimativen Referenzen für Debian sind im Debian-Dokumentations-Projekt (DDP) (`http://www.debian.org/doc/`) zu finden. Viele dieser Dokumente werden in `/usr/share/doc/` installiert. Man sollte auch in `/usr/share/doc-base/` schauen, dies enthält Verweise auf die Dokumentation auf dem System. Man kann `export CDPATH=.:/usr/share/doc:/usr/src/local` zu `~/.bash_profile` hinzufügen, um einfacher auf die Dokumentationsverzeichnisse zuzugreifen.

Das Linux-Dokumentations-Projekt (LDP) (`http://www.tldp.org/`) hat die maßgeblichsten allgemeinen Linux-Referenzen. Der LDP-Inhalt wird gewöhnlich in `/usr/share/doc/HOWTO/` installiert.

Man kann durch Dokumente auf lokalen und entfernten FTP-Servern mittels F9 im Midnight Commander (vergleiche 'Midnight Commander (MC)' auf Seite 51) navigieren.

4.2 Die Linux-Konsole

4.2.1 Login

In einem gewöhnlichen Linux-System gibt es sechs unabhängige Pseudo-Terminals. Man kann zwischen diesen durch gleichzeitiges Drücken von Alt und F1–F6 wechseln. Jedes Pseudo-Terminal erlaubt ein unabhängiges Einloggen. Das Mehrbenutzersystem ist eine großartige Unix-Eigenschaft und sehr suchterzeugend.

Es ist unter Unix eine gute Angewohnheit, sich für die meisten Arbeiten als normaler Nutzer anzumelden. Ich muss zugeben, dass ich den Administrationszugang (root) öfters als nötig der Einfachheit halber und aus Schlamperei verwendete.

Nun verwende ich regelmäßig einen gewöhnlichen Zugang mit den Kommandos `sudo`, `super` oder `su -c` um begrenzten root-Zugang zu erhalten.

4.2.2 Hinzufügen eines neuen Nutzers

Nach der Systeminstallation richte ich gewöhnlich einen normalen Nutzerzugang ein. Ist der Nutzername "pinguin", so wird

```
# adduser pinguin
```

ihn erzeugen.

Ich nutze das `vigr`-Kommando, um `/etc/group` wie folgt zu editieren:

```
adm:x:4:admin
src:x:40:admin, debian, ...
...
```

Im standardmäßig installierten System, können Mitglieder der `adm`-Gruppe viele Logdateien in `/var/log/` lesen und `xconsole` nutzen. Der `staff`-Gruppe gehört `/home` an, was den Mitgliedern das Verwalten von Nutzerzugängen erlaubt. Der `src`-Gruppe gehört `/usr/src/` an, was zur Kernelkompilation, etc. genutzt wird. Für die offizielle Bedeutung von Nutzern und Gruppen vergleichen Sie Nutzer und Gruppen (`/usr/share/doc/base-passwd/users-and-groups.html`).

Ich persönlich verwende die `adm`-Gruppe für Nutzer, die administrative Aufgaben erfüllen und das exklusive `su`-Privileg haben (vergleichen Sie "'Warum GNU `su` nicht die `wheel` Gruppe unterstützt"' auf Seite 128) und `src` für CVS (siehe 'Concurrent Versions System (CVS) – System für simultane Versionen' auf Seite 201).

Überprüfen Sie `adduser`, `addgroup`, `vipw`, `vipw -s`, `vigr` und `vigr -s` für die korrekte Konfiguration von Nutzern und Gruppen.

4.2.3 Wie man den Rechner herunterfährt

Wie nahezu jedes moderne Betriebssystem, das Dateien im Speicher zwischenspeichert, muss Linux richtig heruntergefahren werden, bevor der Rechner ausgeschaltet werden kann. Das entsprechende Kommando im Mehrbenutzerbetrieb ist:

```
# shutdown -h now
```

Im Einzelnutzermodus ist

```
# poweroff -i -f
```

zu verwenden. Warten Sie bis das System "System halted" ausgibt und schalten Sie danach aus. Wenn APM im BIOS und Linux aktiviert ist, schaltet sich der Rechner selbstständig aus. Man vergleiche 'Großer Speicher und automatisches Abschalten' auf Seite 45 für Details.

4.2.4 Kommandozeilenbasiertes Editieren

Die Standard-Shell `bash` besitzt einige Fähigkeiten zur Bearbeitung von alten Eingaben (der History). Benutzen Sie einfach die Cursor-auf Taste um alte Eingaben zurückzuholen und danach wie üblich die Cursor-Tasten. Andere wichtige Tastenkombinationen sind:

```
Strg-U:             Lösche die Zeile bis vor den Cursor
Strg-D:             Beende die Eingabe
markieren mit linker Maustaste:
                    Auswahl in die Zwischenablage kopieren (gpm)
mittlere Maustaste: Einfügen der Zwischenablage an der
                    Cursorposition (gpm)
Strg-V TAB:         Eingabe eines Tabulators ohne Expandierung.
```

In einer normalen Linux Konsole funktioniert nur die linke Strg-Taste wie erwartet.

4.2.5 Kommandoausführung

Ein typisches Kommando wird durch die folgende Shell-Zeile ausgeführt:

```
$ LC_ALL=de ls -la
```

Hier wird das Programm `ls` im Vordergrund ausgeführt, mit der für Deutsch auf `de` gesetzten Umgebungsvariablen `LC_ALL`. `-la` ist das Kommandozeilen-Argument und bewirkt eine detaillierte Ausgabe. Wird die Kommandozeile mit dem `&`-Zeichen abgeschlossen, so wird das Kommando im Hintergrund ausgeführt. Dies ermöglicht es Nutzern, mehrere Programme in einer einzigen Shell zu starten.

Die Ausführung des Kommandos kann durch folgende Tastenkombinationen beeinflusst werden.

```
Strg-C:          Beenden des Programms
Strg-Z:          Temporäres Unterbrechen des Programms
Strg-S:          Unterbrechen der Bildschirmausgabe
Strg-Q:          Ausgabe wieder aktivieren
Strg-Alt-Entf:   Neustarten/Beenden des Systems (siehe /etc/inittab)
```

Für das Management der Programmausführung vergleiche `bash(1)` für `jobs`, `fg`, `bg` und `kill`.

4.2.6 Die grundlegensten Kommandos zur Erinnerung

Die folgenden Kommandos sind für Unix-Systeme fundamental:

```
ls, ls -al, ls -d, pwd, cd, cd ~Benutzer, cd -,
cat /etc/passwd, less, bg, fg, kill, killall,
uname -a, type Kommandoname, sync, netstat,
ping, traceroute, top, vi, ps aux, tar, zcat,
grep, ifconfig, ...
```

Sie können mehr über diese Kommandos herausfinden, indem Sie sie in der Kommandozeile oder nach `man` bzw. `info` eingeben. Viele Linux Kommandos geben eine kurze Meldung zur Benutzung aus, wenn sie mit einer der folgenden Methoden gestartet werden:

```
$ Kommandoname --help
$ Kommandoname -h
```

`whatis` *`Kommandoname`* gibt eine einzeilige Zusammenfassung zu jedem Kommando mit einer Handbuchseite auf dem System aus.

4.2.7 Das X-Window-System

Das X-Window-System lässt sich von der Konsole mit

```
# exec startx
```

starten. Ein Rechtsklick auf den Hintergrund bringt eine Menüauswahl zum Vorschein.

4.2.8 Wichtige Tastaturkommandos

Einige bedeutende Tastenkombinationen für die Linux-Konsole sind ("Plus", "Minus" beziehen sich auf den numerischen Ziffernblock):

```
Alt-F1 bis F6:       Wechsel auf anderes Pseudo-Terminal
Strg-Alt-F1 bis F6:  Wechsel auf anderes Pseudo-Terminal
                     (aus X-Windows, DOSEMU, etc.)
Alt-F7:              Wechsel zu X-Windows
Strg-Alt-Minus:      Ändern der Auflösung in X-Windows
Strg-Alt-Plus:       Ändern der Auflösung in X-Windows (umgekehrte
                     Reihenfolge)
Strg-Alt-Rücktaste:  Beenden des X11-Server-Programms
Alt-X, Alt-C, Alt-V: In üblichen Windows/Mac-Tastenkombinationen
                     zum Ausschneiden, Kopieren und Einfügen wird
                     die Strg-Taste durch Alt in einigen
                     Programmen wie Netscape Composer ersetzt.
```

4.3 Midnight Commander (MC)

Der Midnight Commander (MC) ist ein GNU “Schweizer Armeemesser” für die Linux-Konsole und andere textbasierte Umgebungen.

4.3.1 Installation von MC

```
# apt-get install mc
```

Danach sollten Sie `~/.bashrc` (oder `/etc/bash.bashrc`, was von `.bashrc` verwendet wird) entsprechend der Handbuchseite `mc(1)` unter `-P` ändern. Dies bewirkt eine Änderung des Arbeitsverzeichnisses von MC beim Beenden.

Befindet man sich in einem Terminal wie z.B. `kon` oder `Kterm` für japanische Unterstützung, das verschiedene graphische Zeichen unterstützt, so hilft eventuell das Hinzufügen von `-a` zu MCs Kommandozeile bei Problemen.

4.3.2 Starten von MC

```
$ mc
```

MC bietet für alle Dateioperationen Menüpunkte an, so dass nur minimaler Aufwand nötig ist.

4.3.3 Dateimanager

Standardmäßig gibt es zwei Verzeichnisansichten mit Dateilisten. Ein anderer nützlicher Modus ist das Setzen des rechten Fensters auf “Info”, um Dateizugriffsrechte und andere Informationen zu sehen. Es folgen einige wichtige Tastenkombinationen. Mit einem laufenden `gpm`-Daemon kann man auch die Maus nutzen. (Stellen Sie sicher, dass Sie die Shift-Taste drücken, um das normale Verhalten von Entfernen und Einfügen im MC zu erhalten.)

- F1: Hilfe
- F3: interner Dateibetrachter
- F4: interner Editor
- F9: aktiviert das Menübar-Menü
- F10: beendet den Midnight Commander
- Tab: wechseln zwischen den beiden Fenstern
- Einf: markieren einer Datei für eine Operation wie Kopieren
- Entf: löschen einer Datei (Seien Sie vorsichtig – setzen Sie MC auf sicheres Löschen.)
- Cursor-Tasten: selbsterklärend

4.3.4 Kommandozeilen-Tricks

- Ein `cd` wechselt das Verzeichnis im dargestellten Fenster.
- Strg-Enter oder Alt-Enter kopiert den Dateinamen in die Kommandozeile. Dies kann mit dem `cp`- oder `mv`-Programm und dem Kommandozeilen-Editor genutzt werden.
- Alt-Tab bestimmt die Dateinamenvervollständigung in der Shell.
- Man kann das Startverzeichnis für beide Fenster als Argumente an MC übergeben, z.B. `mc /etc /root`.
- Esc + *Ziffer* == F*n* (d.h. Esc + '1' = F1, etc.; Esc + '0' = F10).
- Esc == Alt (= Meta, M-), d.h. Esc + 'c' steht für Alt-c.

4.3.5 Editor

Der interne Editor besitzt ein interessantes Entfernen-und-Einfügen-Schema. Drücken von F3 markiert den Start einer Auswahl, ein zweites F3 markiert das Ende der Auswahl und hebt diese hervor. Danach kann man den Cursor bewegen. Beim Betätigen von F6 wird der hervorgehobene Bereich an die Cursorposition verschoben. Drückt man F5, so wird der ausgewählte Bereich an die aktuelle Cursorposition kopiert und eingefügt. F2 sichert die Datei und F10 beendet den Editor. Die meisten Cursortasten funktionieren wie üblich.

Dieser Editor kann direkt mit einer Datei aufgerufen werden:

```
$ mc -e Datei
$ mcedit Datei
```

Es handelt sich nicht um einen Mehr-Fenster-Editor, aber man kann mehrere Linux-Konsolen nutzen, um den selben Effekt zu erreichen. Um zwischen Fenstern zu kopieren, sind die Alt-F*n* Tasten zum Wechseln der virtuellen Konsole und "Datei->Datei einfügen" oder "Datei->In Datei kopieren" zum Verschieben eines Teils einer Datei in eine andere zu verwenden.

Dieser interne Editor kann durch einen beliebig anderen externen Editor ersetzt werden.

Viele Programme nutzen die Umgebungsvariable `EDITOR` oder `VISUAL` um zu bestimmen, welcher Editor zu verwenden ist. Sollten Sie mit `vim` nicht vertraut sein, setzen Sie diese auf `mcedit`, indem Sie folgende Zeilen zu `~/.bashrc` hinzufügen:

```
...
export EDITOR=mcedit
export VISUAL=mcedit
...
```

Ich empfehle, diese wenn möglich auf `vim` zu setzen. Die Verwendung der vi(m)-Kommandos ist das richtige Vorgehen, da sie stets in einer Linux/Unix-Umgebung zu finden sind.

4.3.6 Dateibetrachter

Dies ist ein guter Viewer und prima dazu geeignet, Wörter in Dokumenten zu suchen. Ich nutze dies immer für Dateien im `/usr/share/doc`-Verzeichnis. Dies ist der schnellste Weg, um sich einen Überblick über die mehr als umfangreiche Linux-Dokumentation zu verschaffen. Der Dateibetrachter kann direkt mittels

```
$ mc -v Datei
```

gestartet werden. (Es ist zu beachten, dass einige Pakete die Regeln verletzen und ihre Dokumentation noch unter `/usr/doc/` ablegen.)

4.3.7 Auto-Start-Eigenschaften

Drückt man Enter für eine Datei, so wird ein entsprechendes Programm den Inhalt der Datei verarbeiten. Dies ist eine sehr nützliche Eigenschaft von MC.

```
ausführbare Dateien:  Kommando ausführen
man, html Dateien:    Darstellen des Inhalts mit Dateibetrachter
tar, gz, rpm Dateien: Inhalt als Unterverzeichnis einblenden
```

Damit diese Eigenschaften funktionieren, dürfen betrachtbare Dateien nicht ausführbar sein. Ihr Status kann mittels des `chmod`-Kommandos und dem Dateimenü von MC geändert werden.

4.3.8 Virtuelles FTP-Dateisystem

MC kann zum Zugriff auf Dateien übers Internet mittels FTP genutzt werden. Aktivieren Sie mittels F9 das Menü und drücken Sie dann 'p' für das virtuelle FTP-Dateisystem. Eine URL wird in der Form `Benutzername:Passwort@hostname.domainname` eingegeben und MC stellt daraufhin das entfernte Verzeichnis wie ein lokales dar.

4.4 Grundlagen des GNU/Linux-Dateisystems

Jede Datei und jedes Verzeichnis eines GNU/Linux-Systems ist mit einem Nutzer verknüpft, der sie/es besitzt und einer Gruppe, zu welcher sie/es gehört. Alle Dateiinformationen werden in einem **Inode** genannten Datensatz gespeichert.

4.4.1 Zugriffsrechte von Dateien und Verzeichnissen

Die Zugriffsrechte von Dateien und Verzeichnissen sind separat für die folgenden drei Nutzerkategorien definiert:

- den **Nutzer** (user), der die Datei besitzt (u),
- andere Nutzer der **Gruppe** (group), zu welcher die Datei gehört (g) und
- alle **anderen** (other) Nutzer (o).

Bei Dateien sind je folgende Rechte möglich:

- **lesbar** (readable) (r): zum Auslesen des Inhalts der Datei
- **schreibbar** (writable) (w): zum Modifizieren der Datei
- **ausführbar** (executable) (x): zum Starten der Datei als Kommando

Für Verzeichnisse sind je folgende Rechte möglich:

- **lesbar** (readable) (r): zum Anzeigen des Verzeichnisinhalts
- **schreibbar** (writable) (w): zum Hinzufügen oder Entfernen von Dateien im Verzeichnis
- **ausführbar** (executable) (x): zum Zugriff auf Dateien im Verzeichnis

Das **ausführbar**-Recht des Verzeichnisses bezieht sich nicht nur auf das Lesen von Dateien im entsprechenden Verzeichnis, sondern auch auf das Bestimmen der Attribute wie Größe und Modifizierungszeit.

Um diese und einige andere Informationen zu Dateien und Verzeichnissen zu erhalten, kann `ls` genutzt werden. Vergleiche `ls(1)`. Wird `ls` mit der `-l`-Option gestartet, so werden die folgenden Informationen in der folgenden Reihenfolge angezeigt:

- der **Dateityp** (erstes Zeichen)
  - -: normale Datei
  - d: Verzeichnis (directory)
  - l: symbolischer Link
  - c: zeichenorientiertes Gerät
  - b: blockorientiertes Gerät
- die **Zugriffsrechte** der Datei (die nächsten neun Zeichen sind je drei Zeichen für Nutzer, Gruppe und Andere in dieser Reihenfolge)
- die **Anzahl der harten Links** zur Datei
- der Name des **Nutzers** der die Datei besitzt
- der Name der **Gruppe** zu der die Datei gehört
- die **Größe** der Datei in Bytes
- **das Datum und die Zeit** der Datei (Modifizierungszeit, mtime)
- der **Name** der Datei.

Um den Besitzer einer Datei zu ändern, wird `chown` von root genutzt. Vergleiche `chown(1)`. Zum Ändern der Gruppe einer Datei, wird `chgrp` vom Besitzer der Datei oder root verwendet. Vergleiche `chgrp(1)`. Das Ändern der Zugriffsrechte von Dateien oder Verzeichnissen erfolgt mit `chmod` durch den Dateibesitzer oder root. Vergleiche `chmod(1)`.

Um z.B. einen Verzeichnisbaum dem Nutzer *nutzer* und der Gruppe *gruppe* zuzuordnen, ist als root Folgendes auszuführen:

```
# cd /Pfad/zum/Verzeichnis
```

```
# chown -R nutzer:gruppe .
# chmod -R ug+rwX,o=rX .
```

Es gibt drei weitere spezielle Zugriffsbits:

- **set user ID** (s oder S anstatt Nutzer's x),
- **set group ID** (s oder S anstatt Gruppe's x) und
- **Sticky Bit** (t oder T anstatt Andere's x).

Hier werden in der Ausgabe von `ls -l` diese Bits großgeschrieben, wenn die ausführbar-Bits, die in der Ausgabe nicht mit dargestellt werden können, nicht gesetzt sind.

Das Setzen der **set user ID** einer ausführbaren Datei, erlaubt es einem Nutzer, die Datei mit den Rechten und der ID des Besitzers der Datei auszuführen (z.B. **root**). Ähnlich erlaubt das Setzen des **set group ID**-Rechts einer ausführbaren Datei, dass diese mit der Gruppen-ID und den Gruppenrechten der Datei (z.B. **root**) gestartet wird. Dies kann zu Sicherheitsproblemen führen, deshalb erfordert das Setzen dieser Bits spezielle Vorsicht.

Setzen der **set group ID** eines Verzeichnisses aktiviert das BSD-artige Dateierzeugungs Schema, bei welchem alle im Verzeichnis erstellten Dateien der Gruppe **group** des Verzeichnisses angehören.

Das Setzen des **Sticky Bits** eines Verzeichnisses verhindert das Entfernen einer Datei im Verzeichnis durch einen Nutzer, der nicht der Besitzer ist. Um den Inhalt einer Datei in einem für alle oder auch für Gruppenmitglieder schreibbaren Verzeichnis wie `/tmp` zu sichern, genügt es nicht das **schreibbar**-Recht der Datei zu entfernen, sondern es muss auch das **Sticky Bit** des Verzeichnisses gesetzt werden. Andernfalls kann die Datei entfernt werden und eine neue Datei mit gleichem Namen durch andere Nutzer mit Schreibrechten im Verzeichnis angelegt werden.

Es folgen einige interessante Beispiele für Zugriffsrechte von Dateien.

```
$ ls -l /etc/passwd /etc/shadow /dev/ppp /usr/sbin/pppd
crw-rw----    1 root    dip   108,    0 Jan 18 13:32 /dev/ppp
-rw-r--r--    1 root    root       1051 Jan 26 08:29 /etc/passwd
-rw-r-----    1 root    shadow      746 Jan 26 08:29 /etc/shadow
-rwsr-xr--    1 root    dip      234504 Nov 24 03:58 /usr/sbin/pppd
$ ls -ld /tmp /var/tmp /usr/local /var/mail /usr/src
drwxrwxrwt    4 root    root       4096 Feb  9 16:35 /tmp
drwxrwsr-x   10 root    staff      4096 Jan 18 13:31 /usr/local
drwxrwsr-x    3 root    src        4096 Jan 19 08:36 /usr/src
drwxrwsr-x    2 root    mail       4096 Feb  2 22:19 /var/mail
drwxrwxrwt    3 root    root       4096 Jan 25 02:48 /var/tmp
```

4.4.2 Zeitmarken

Es gibt drei Arten von Zeitmarken für GNU/Linux Dateien:

- **mtime**: die Modifizierungszeit (`ls -l`),

- **ctime**: die Zeit der letzten Statusänderung (`ls -lc`) und
- **atime**: die letzte Zugriffszeit (`ls -lu`).

Es ist zu beachten, dass **ctime** nicht der Zeitpunkt der Dateierzeugung ist.

- Das Überschreiben einer Datei ändert sowohl **mtime**, **ctime** als auch **atime** der Datei.
- Das Ändern der Rechte oder des Besitzers einer Datei ändert **ctime** und **atime** der Datei.
- Das Lesen einer Datei ändert **atime** der Datei.

Beachten Sie, dass selbst ein einfacher Lesezugriff auf eine Datei in einem GNU/Linux-System normalerweise eine Schreiboperation erfordert, um die **atime**-Zeit in der **Inode** anzupassen. Das Mounten eines Dateisystems mit der `noatime`-Option unterlässt dies und beschleunigt den Dateizugriff beim Lesen. Vergleiche `mount(8)`.

4.4.3 Links

Es gibt zwei Methoden, um eine Datei *foo* mit einem anderen Dateinamen *bar* zu verknüpfen.

- Ein **harter Link** (hard link) ist ein anderer Name für eine existierende Datei (`ln foo bar`),
- Ein **symbolischer Link** (symlink) verweist auf eine andere Datei über den Namen (`ln -s foo bar`).

Man vergleiche das folgende Beispiel für Änderungen der Anzahl der Links und die feinen Unterschiede beim Anwenden des `rm`-Kommandos.

```
$ echo "ursprünglicher Dateiinhalt" > foo
$ ls -l foo
-rw-r--r--    1 osamu    osamu           4 Feb  9 22:26 foo
$ ln foo bar     # hard link
$ ln -s foo baz  # symlink
$ ls -l foo bar baz
-rw-r--r--    2 osamu    osamu           4 Feb  9 22:26 bar
lrwxrwxrwx    1 osamu    osamu           3 Feb  9 22:28 baz -> foo
-rw-r--r--    2 osamu    osamu           4 Feb  9 22:26 foo
$ rm foo
$ echo "neuer Dateiinhalt" > foo
$ cat bar
ursprünglicher Dateiinhalt
$ cat baz
neuer Dateiinhalt
```

Alle symbolischen Links haben die Dateizugriffsrechte “rwxrwxrwx” wie im obigen Beispiel gezeigt und deren Zugriffsrechte werden durch die Datei auf die sie zeigen diktiert.

Das Verzeichnis **.** verweist auf das Verzeichnis in dem es sich befindet, die Anzahl der Links für ein neues Verzeichnis beginnt bei 2. Das Verzeichnis **..** verweist auf das übergeordnete Verzeichnis, die Anzahl der Links eines Verzeichnisses erhöhen sich mit dem Erzeugen neuer Unterverzeichnisse.

4.5 Weiterführende Informationen

Es gibt viele gute Referenzen zum Einstieg in Unix. O'Reilly's Bücher sind gewöhnlich gute Anleitungen für alle Computerthemen. Das LDP-Dokument Linux Tipps HOWTO (`http://www.tldp.org/HOWTO/Tips-HOWTO.html`) ist eine andere nützliche Informationsquelle. Man vergleiche 'Unterstützung für Debian' auf Seite 235 für weitere Ressourcen.

Kapitel 5

Aktualisieren einer Distribution auf `stable`, `testing` oder `unstable`

Offizielle Hinweise für die Aktualisierung finden sich unter `http://www.debian.org/releases/stable/releasenotes` und `http://www.debian.org/releases/testing/releasenotes` (befindet sich in Arbeit).

Eine Systemaktualisierung auf die `stable`-, `testing`- oder `unstable`-Distribution kann verschiedene Schritte in der angegebenen Reihenfolge erfordern:

- Aktualisierung auf Woody (wenn das System älter als Woody ist)
- Aktualisierung auf `stable`
- Aktualisierung auf `testing`
- Aktualisierung auf `unstable`

Debian unterstützt keine Aktualisierungen, die einzelne Veröffentlichungen übergehen.

5.1 Aktualisierung von Potato auf Woody

Dies wird separat beschrieben, da Potato's APT nicht über alle in der aktuellen `apt_preferences(5)`-Handbuchseite beschriebenen Fähigkeiten verfügt.

Nach dem Einbinden von Woody-Verweisen (und keinen anderen) zu `/etc/apt/sources.list` werden APT und benötigte Kernpakete auf Woody-Versionen wie folgt aktualisiert:

```
# apt-get update
# apt-get install libc6 perl libdb2 debconf
# apt-get install apt apt-utils dselect dpkg
```

Danach wird der Rest des Systems auf Woody aktualisiert:

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

5.2 Vorbereitung einer Aktualisierung

Die Aktualisierung einer Distribution auf eine andere kann durch Herunterladen der Pakete aus dem Netzwerk wie folgt durchgeführt werden.

Erzeugen einer sauberen Liste von Paket-Depots für `stable`:

```
# cd /etc/apt
# cp -f sources.list sources.list.old
# :>sources.list
# apt-setup noprobe
```

Soll auf `testing` aktualisiert werden, so sind `testing`-Quellen zu dieser neuen Liste hinzuzufügen. Soll auf `unstable` aktualisiert werden, so sind auch `unstable`-Quellen hinzuzufügen.

```
# cd /etc/apt
# grep -e "^deb " sources.list >srcs
# :>sources.list
# cp -f srcs sources.list
# sed -e "s/stable/testing/" srcs >>sources.list
# sed -e "s/stable/unstable/" srcs >>sources.list
# apt-get update
# apt-get install apt apt-utils
```

Vergleiche 'Grundlagen der Debian-Paketverwaltung' auf Seite 64 für die Kunst des Tunings von `/etc/apt/sources.list` und `/etc/apt/preferences`.

5.3 Aktualisierung

Nach den korrekten Anpassungen von `/etc/apt/sources.list` und `/etc/apt/preferences`, wie oben beschrieben, kann mit der Aktualisierung begonnen werden.

Das Verwenden der `testing`-Distribution von Debian kann den Nebeneffekt haben, dass Sicherheitsupdates nur langsam eingespielt werden, da solche Pakete nach `unstable` hochgeladen werden und erst später nach `testing` gelangen.

Man vergleiche mit 'Debian-Paketverwaltung' auf Seite 63 für die Grundlagen und sollten dabei Fehler auftreten mit 'APT-Upgrade-Fehlersuche' auf Seite 71.

5.3.1 Verwendung von `dselect`

Wenn ein System zahlreiche Pakete enthält, die `-dev`-Pakete, etc. nutzen, wird die im Folgenden beschriebene Vorgehensweise mittels `dselect` empfohlen, da hier eine feinere Kontrolle der Pakete möglich ist.

```
# dselect update # muss vor jeder Aktualisierung ausgeführt werden
# dselect select # Auswahl zusätzlicher Pakete
```

All Ihre aktuellen Pakete werden ausgewählt, wenn `dselect` startet. `dselect` verlangt dabei eventuell nach zusätzlichen Paketen, basierend auf `Abhängigkeiten`, `Vorschlägen` und `Empfehlungen`. Möchten Sie keine Pakete hinzufügen, drücken Sie einfach `Q`, um `dselect` erneut zu beenden.

```
# dselect install
```

Sie werden einige Paket-Konfigurationsfragen während dieses Teils des Prozesses beantworten müssen, halten Sie also Ihre Notizen bereit und planen Sie genug Zeit dafür ein. Vergleiche '`dselect`' auf Seite 65.

Nutzen Sie `dselect`. **Dies funktioniert immer :)**

5.3.2 Verwendung von `apt-get`

```
# apt-get update
# apt-get -t stable upgrade
# apt-get -t stable dist-upgrade
# apt-get -t testing upgrade
# apt-get -t testing dist-upgrade
# apt-get -t unstable upgrade
# apt-get -t unstable dist-upgrade
```

Wenn erst einmal Sarge läuft, ist es ratsam `aptitude` statt `apt-get` zu verwenden. (`aptitude` akzeptiert viele der Optionen von `apt-get`, inklusive der obigen.)

Zum Aktualisieren unter Beibehaltung der aktuellen `dselect`-Einstellungen:

```
# apt-get dselect-upgrade
```

Man vergleiche 'Paketabhängigkeiten' auf Seite 16.

Kapitel 6

Debian-Paketverwaltung

Dieses Kapitel basiert auf älterer Version des englischen ursprünglichen Textes. Die englische Version bitte auch überprüfen.

Um die Netzlast in den Debian-Archiven zu reduzieren, sollten Sie einen lokalen HTTP-Proxy mit `squid` für das Herunterladen von Paketen mittels APT einrichten. Wenn nötig muss die `http_proxy`-Umgebungsvariable oder der `http`-Wert in `/etc/apt/apt.conf` gesetzt werden. Dies erhöht die Leistung im Netzwerk enorm, insbesondere dann, wenn viele Debian-Rechner in einem LAN vorhanden sind.

Obwohl die Pinning-Eigenschaft von `apt_preferences(5)` sehr leistungsfähig ist, löst sie nicht alle Abhängigkeitsprobleme, da die Voraussetzungen für Abhängigkeiten an neueren Versionen anderer fundamentaler Programmpakete zerren.

Die Verwendung der in '`chroot`' auf Seite 119 beschriebenen Methode ist geeignet, um gleichzeitig die Stabilität beider Systeme zu sichern und auf die neuesten Versionen von Software zuzugreifen.

Dieses Kapitel bezieht sich auf ein Woody-System, jedoch sollten die Informationen hier auch auf ein Potato-System (mit Ausnahme von `apt_preferences(5)` und Themen die Bezug auf die `/etc/apt/preferences`-Datei nehmen) zutreffen.

6.1 Einführung

Wenn das Lesen der Entwicklerdokumentation zu viel für Sie ist, dann lesen Sie zuerst dieses Kapitel um dann die ganze Vielfalt von Debian (`testing/unstable`) zu nutzen :-)

6.1.1 Grundlegende Werkzeuge

```
dselect   - menügesteuertes Paketverwaltungs-Werkzeug (top level)
dpkg      - installiert Pakete (Paketdatei orientiert)
apt-get   - installiert Pakete (Paketarchiv orientiert, CLI-APT)
```

```
tasksel    - installiert Aufgaben-Profile (Verbund von Paketen)
aptitude   - installiert Pakete (Pakete & Profile, ncurses APT)
deity      - alternatives ncurses APT
synaptic, gsynaptic - GUI-APT-Alternativen
```

Diese Tools arbeiten nicht alle auf dem gleichen Level. `dselect` arbeitet ein Level über APT (der Kommandozeilenbefehl ist `apt-get`) und `dpkg`.

APT nutzt `/var/lib/apt/lists/*` um verfügbare Pakete zu erfassen, `dpkg` jedoch nutzt `/var/lib/dpkg/available`. Wenn Sie Pakete direkt mit `apt-get` oder mit einem ähnlichen Programm, wie z.B. `aptitude` installiert haben, dann stellen Sie sicher, dass Sie die Datei `/var/lib/dpkg/available` mittels `[U]pdate` aus dem `dselect`-Menü oder von der Kommandozeile mit "`dselect update`" updaten, bevor Sie `dselect select`, `tasksel` oder `dpkg -l` ausführen.

Was die Paketabhängigkeiten angeht, so berücksichtigt `apt-get` automatisch alle **benötigten** Pakete, ignoriert aber **empfohlende** und **vorgeschlagene** Pakete. `dselect` bietet jedoch eine menübasierte Kontrolle über die Auswahl von solchen Paketen und fragt zur Auswahl von **benötigten**, **empfohlenden** und **vorgeschlagenen** Paketen. `aptitude` bietet die Möglichkeit, alle **benötigten**, **empfohlenden** und **vorgeschlagenen** Pakete automatisch zu berücksichtigen. Siehe 'Paketabhängigkeiten' auf Seite 16.

6.1.2 Nützliche Tools

```
apt-cache         - durchsucht das Paketarchiv im lokalen Cache
dpkg-reconfigure  - erneutes Konfigurieren von installierten
                    Paketen (sofern sie debconf benutzen)
dpkg-source       - verwaltet Quellpaketdateien
dpkg-buildpackage - automatisiert den Prozess der Erstellung von
                    Paketdateien
...
```

6.2 Grundlagen der Debian-Paketverwaltung

Man kann eine Menge von Paketen genannt *Aufgaben-Profile* oder auch *Task* installieren oder individuelle Pakete. Eine Aktualisierung des Systems kann auch mittels Paketverwaltungswerkzeugen wie weiter unten beschrieben erfolgen. Es wird auch auf 'Debian-System-Installationshinweise' auf Seite 27, 'Aktualisieren einer Distribution auf `stable`, `testing` oder `unstable`' auf Seite 59 und 'Rettungseditoren' auf Seite 195 verwiesen.

6.2.1 *Aufgaben-Profile* mit `tasksel` oder `aptitude` installieren

`tasksel` ist das **Debian-Aufgaben-Profil-Installationsprogramm**, das auch während der Debian-Installation unter dem Punkt "`einfach`" angeboten wird.

Wenn man eine bestimmte Funktion benötigt, die verschiedene Pakete erfordert, so ist das der einfachste Weg dies zu erledigen. Stellen Sie sicher, dass Sie die Befehle wie folgt starten:

```
# dselect update
# tasksel
```

`aptitude` bietet auch Zugang zu *Aufgaben-Profilen*. Dies erlaubt nicht nur die Auswahl von *Aufgaben-Profilen*, sondern ermöglicht es auch, Pakete in diesem *Profil* selektiv durch ein Menü abzuwählen.

6.2.2 Einrichten des APT-Systems

Versuchen Sie ein System mittels verschiedener Distributionen, wie im Folgenden beschrieben, einzurichten, so ist es wahrscheinlich, dass Sie auf einige Paketabhängigkeitskonflikte stoßen. Es ist eine gute Idee, verschiedene Ausgaben nicht zu vermengen. Das Folgende ist für Personen, die gerne experimentieren und dabei Risiken in Kauf nehmen.

Um selektive Upgrades durchzuführen, während man die `testing`-Distribution installiert hat, muss das APT-System (>Woody) wie in 'Aktualisierung von Potato auf Woody' auf Seite 59 beschrieben eingerichtet werden, damit die `apt_preferences(5)` Eigenschaften genutzt werden können.

Fügen Sie zuerst die Quellen für `stable`, `testing` und `unstable` in Ihre `/etc/apt/sources.list`-Datei ein. Editieren Sie danach `/etc/apt/preferences`, um Pin-Priority korrekt einzurichten. [1]

```
Package: *
Pin: release a=stable
Pin-Priority: 500

Package: *
Pin: release a=testing
Pin-Priority: 600

Package: *
Pin: release a=unstable
Pin-Priority: 50
```

6.2.3 dselect

Wenn `dselect` startet, wählt es automatisch alle Pakete aus, die in den Kategorien "Required", "Important" und "Standard" enthalten sind. In der Potato-Distribution waren etliche

[1] Ich behielt die Schilderung hier bei wie sie ist, um Konsistenz mit anderen Teilen zu gewährleisten. Wenn Sie wirklich `testing` oder `unstable` verwenden möchten, sollten Sie Verweise auf `stable` in `/etc/apt/sources.list` und `/etc/apt/preferences` entfernen. Dies ist empfehlenswert, da `testing` als eine Kopie von `stable` startet.

große Pakete wie z.B. teTeX und Emacs in diesen Kategorien enthalten und wurden am besten für die Erstinstallation manuell abgewählt (mittels '_'). In der Woody-Distribution sind diese Pakete nun in der "Optional"-Paketkategorie.

`dselect` hat eine einigermaßen gewöhnungsbedürftige Benutzerschnittstelle. Es gibt gleich vier doppeldeutige Befehle (Großbuchstaben sind gemeint!):

```
Taste  Aktion
Q      Beenden. Momentane Auswahl bestätigen und beenden.
        (Abhängigkeiten vernachlässigen)
R      Rückgängig! Es war nicht so gemeint.
D      Verdammt! Es ist uninteressant was dselect meint. Mach es!
U      Alles auf den vorgeschlagenen Status setzen.
```

Mit D und Q kann eine Auswahl die Konflikte enthält, auf eigenes Risiko gewählt werden. Hiermit sollte vorsichtig umgegangen werden. Zurzeit ist `dselect` das ausgereifteste menübasierte Werkzeug, das fein gegliederte Kontrolle über **Empfehlungen** und **Vorschläge** bietet.

Fügen Sie eine Zeile in `/etc/dpkg/dselect.cfg` ein, die die Option "expert" enthält, um die Ausgabe unnötiger Meldungen zu reduzieren. Für langsame Rechner, ist es empfehlenswert `dselect` auf einem anderen, schnelleren Rechner zu starten um Pakete zu finden und diese mit `apt-get install` zu installieren.

`dselect` erlaubt keinen Zugriff auf Pakete, die nicht durch Pin-Priority begünstigt sind.

6.2.4 `aptitude`

`aptitude` ist ein menübasiertes Paketinstallationprogramm ähnlich zu `dselect`. Dies kann auch als alternativer Kommandozeilen-Befehl an Stelle von `apt-get` genutzt werden. Vergleichen Sie `aptitude(1)`.

`aptitude` akzeptiert folgende Tastenkürzel, die meist Kleinbuchstaben sind.

```
Tastenkürzel  Aktion
F10           Menü
?             Hilfe zu Tastenkürzel
u             Aktualisiere Paketarchivinformation
g             Herunterladen und installieren gewählter Pakete
q             Beenden und Sichern der Änderungen
x             Beenden und Verwerfen der Änderungen
Enter         Informationen zu einem Paket anzeigen
```

`aptitude` erlaubt es, alle Pakete, je nachdem, ob sie **benötigt**, **empfohlen** oder **vorgeschlagen** werden, auszuwählen. Dieses Verhalten kann durch Auswahl von `F10 -> Optionen -> Abhängigkeitsbehandlung` im Menü geändert werden.

`aptitude` erlaubt den Zugriff auf alle Versionen eines Pakets.

6.2.5 Die `apt-cache`- und `apt-get`-Kommandos

Beim Einrichten von `testing` wie im obigen Beispiel beschrieben, kann das System durch folgende Kommandos angepasst werden:

- `apt-get -u upgrade`

  Dies verwendet die `testing`-Distribution und aktualisiert alle Pakete im System, wobei die durch Abhängigkeiten benötigten Pakete von `testing` stammen.

- `apt-get -u dist-upgrade`

  Dies verwendet die `testing`-Distribution und aktualisiert alle Pakete im System, wobei alle Abhängigkeiten aufgelöst werden und die dadurch benötigten Pakete von `testing` stammen.

- `apt-get -u dselect-upgrade`

  Dies verwendet die `testing`-Distribution und aktualisiert alle Pakete im System entsprechend der Auswahl durch `dselect`.

- `apt-get -u install` *`Paket`*

  Dies installiert *Paket* und dessen Abhängigkeiten aus der `testing`-Distribution.

- `apt-get -u install` *`Paket`*`/unstable`

  Dies installiert *Paket* aus der `unstable`-Distribution, während dessen Abhängigkeiten aus der `testing`-Distribution installiert werden.

- `apt-get -u install -t unstable` *`Paket`*

  Dies installiert *Paket* aus der `unstable`-Distribution, während dessen Abhängigkeiten auch aus der `testing`-Distribution mit einer Pin-Priority von 990 für `unstable` installiert werden.

- `apt-cache policy` *`foo bar ...`*

  Dies gibt den Status der Pakete *foo bar ...* aus.

- `apt-cache show` *`foo bar ...`* `| less`

  Dies gibt die Informationen zu den Paketen *foo bar ...* aus.

- `apt-get install` *`foo=2.2.4-1`*

  Dies installiert die spezielle Version *2.2.4-1* des Pakets *foo*.

- `apt-get -u install` *`foo bar-`*

  Dies installiert das Paket *foo* und entfernt das *bar* Paket.

- `apt-get remove` *`bar`*

  Dies entfernt das Paket *bar*, die angepasste Konfiguration wird jedoch beibehalten.

- `apt-get remove --purge bar`

  Dies entfernt das Paket *bar* zusammen mit allen Konfigurationsdateien.

Die Option `-u` in den obigen Beispielen sorgt dafür, dass `apt-get` eine Liste aller Pakete ausgibt, die aktualisiert werden können und fragt den Nutzer bevor es beginnt. Das Folgende setzt die Option `-u` standardmäßig:

```
$ cat >> /etc/apt/apt.conf << .
// Immer zu aktualisierende Pakete anzeigen (-u)
APT::Get::Show-Upgraded "true";
.
```

Die Option `-s` kann zum Simulieren eines Upgrades verwendet werden, ohne dass eine Aktualisierung erfolgt.

6.2.6 Verwenden einer speziellen Version der Debian-Distribution

Abhängig von der von Ihnen bevorzugten Debian-Version kann die Datei `/etc/apt/preferences` in 'Einrichten des APT-Systems' auf Seite 65 entsprechend Ihren Wünchen angepasst werden:

```
für stable:            Pin-Priority von testing auf 50 setzen
für testing:           Einstellungen wie oben gezeigt beibehalten
für testing(unstable): Pin-Priority von unstable auf 500 setzen
für unstable(testing): Pin-Priority von unstable auf 800 setzen
```

Eine Richtlinie für die Wahl der Pin-Priority ist das Wechseln von oben nach unten in der obigen Tabelle, so wie die Zeit von unmittelbar nach einer Veröffentlichung bis zum Einfrieren für die nächste Version vergeht.

Achtung: Das Verwenden der `testing`-Version von Debian hat den Nebeneffekt, dass Sicherheitsupdates nur langsam eingespielt werden.

Wenn verschiedende Versionen von Debian verwendet werden, so wie `testing` mit `stable` oder `unstable` mit `stable`, so werden höchstwahrscheinlich wichtige Pakete unabsichtlich von `testing` oder `unstable` genutzt, die fehlerhaft sein können. Sie wurden gewarnt.

Beispiele von `/etc/apt/preferences`, in welchen einige wichtige Pakete auf ausgereifte Versionen gesetzt werden, während die nicht ganz ausgereiften für nichtessentielle Pakete verwendet werden, sind im Beispielverzeichnis (`http://www.debian.org/doc/manuals/debian-reference/examples/`) als `preferences.testing` und `preferences.unstable` verfügbar. Auf der anderen Seite erzwingt `preferences.stable` einen Versionssprung aller Pakete zu `stable` (Downgrad).

6.2.7 Downgrad aller Pakete zu `stable`

Um alle Pakete auf die `stable`-Distribution zurückzustufen, editieren Sie die Datei `/etc/apt/preferences` wie folgt:

```
Package: *
Pin: release a=stable
Pin-Priority: 1001
```

und führen Sie "`apt-get upgrade`" aus. Dieser Befehl stuft aufgrund der Pin-Priority > 1000 sämtliche Pakete herunter. Aber seien Sie gewarnt, es könnte kleinere Abhängigkeitsprobleme geben.

6.2.8 Überblick über `/etc/apt/preferences`

In dem in 'Einrichten des APT-Systems' auf Seite 65 aufgeführten Beispiel von `/etc/apt/preferences` bedeutet "Package: *" in der ersten Zeile, dass sich der Abschnitt auf alle Pakete bezieht. Man kann auch einen Paketnamen anstatt von "*" verwenden, um die Pin-Priority für das Paket festzulegen.

"Pin: release a=stable" in der zweiten Zeile bedeutet, dass `apt-get` die `Packages.gz`-Datei aus dem Archiv nimmt, das die Zeile "Archive: stable" in seiner `Release`-Datei enthält.

Die letzte Zeile "Pin-Priority: 500" setzt Pin-Priority auf 500.

Wenn es mehrere Pakete mit dem selben Namen gibt, so wird normalerweise das mit der höchsten Pin-Priority installiert.

Die Bedeutung der Pin-Priority ist:

- 1001 und höher: Zurückstufungsprioritäten.

  Ein Zurückstufen (Downgrade) des Pakets ist erlaubt für diesen Bereich der Pin-Priority.
- 100 bis 1000: Standardprioritäten.

  Ein Zurückstufen des Pakets ist nicht erlaubt. Einige wichtige Pin-Priority Werte:
  - 990: die Priorität, die durch die `--target-release` oder `-t` Option von `apt-get(8)` gesetzt wird.
  - 500: die Priorität für alle Standardpaketdateien.
  - 100: die Priorität der zurzeit installierten Paketdateien.
- 0 bis 99: Nicht-automatische Prioritäten. (Diese werden nur genutzt, wenn das Paket nicht installiert ist und keine andere Version verfügbar ist.)
- kleiner als 0: Die Version wird niemals gewählt.

Die selben Effekte wie `--target-release` können durch Setzen von `/etc/apt/apt.conf` mittels

```
# echo 'APT::Default-Release "testing";' >> /etc/apt/apt.conf
```

erreicht werden. Die Kommandozeilenoption `--target-release` und der Wert in `/etc/apt/apt.conf` ersetzen die Einstellung in `/etc/apt/preferences`. Seien Sie vorsichtig damit, wenn Sie gleichzeitig `/etc/apt/preferences` editieren.

6.3 Debian-Überlebensbefehle

Mit diesem Wissen kann jeder ein Leben des **ewigen** "Aktualisierens" führen :-)

6.3.1 Überprüfen von Bugs in Debian und Suche nach Hilfe

Wenn Sie auf Probleme bezüglich eines bestimmten Pakets stoßen, stellen Sie sicher, dass Sie zuerst diese Seiten überprüfen, bevor Sie um Hilfe fragen oder einen Bug-Report erstellen. (`lynx`, `links` und `w3m` sind hier gleichwertig):

```
$ lynx http://bugs.debian.org/
$ lynx http://bugs.debian.org/Paketname # wenn der Name bekannt ist
$ lynx http://bugs.debian.org/Bugnummer # wenn Bugnr. bekannt ist
```

Suchen Sie auch bei Google (`www.google.de`) mit Suchbegriffen wie "site:debian.org".

Wenn Sie nicht weiter kommen, lesen Sie die guten Handbücher. Setzen Sie `CDPATH` wie folgt:

```
export CDPATH=.:/usr/local:/usr/share/doc
```

und dann

```
$ cd Paketname
$ pager README.Debian # falls dies existiert
$ mc
```

Weitere Hilfequellen sind in 'Unterstützung für Debian' auf Seite 235 aufgelistet.

6.3.2 APT-Upgrade-Fehlersuche

Es können Paketabhängigkeitprobleme beim Aktualisieren auf `unstable/testing` entsprechend 'Aktualisierung' auf Seite 60 entstehen. Meistens liegt das daran, dass ein Paket aktualisiert wird, das neue Abhängigkeiten hat, die nicht erfüllt sind. Diese Probleme können unter Verwendung folgender Befehle gelöst werden:

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

Wenn dies nicht hilft das Problem zu lösen, dann wiederholen Sie folgende Befehle solange, bis sich das Problem auflöst:

```
# apt-get upgrade -f       # selbst bei Fehlern mit Upgrade ...
... oder
# apt-get dist-upgrade -f  # ... bzw. Dist-Upgrade fortfahren
```

Einige sehr fehlerhafte Upgrade-Skripte könnten permanente Schwierigkeiten bereiten. In solch einer Situation ist es gewöhnlich besser, wenn man sich die `/var/lib/dpkg/info/`*`Paketname`*`.{post,pre}{inst,rm}`-Skripte der betreffenden Pakete ansieht und anschließend

```
# dpkg --configure -a      # konfiguriert alle nicht vollständig
                           # installierten Pakete
```

ausführt.

Wenn eines der Skripte sich über eine fehlende Konfigurationsdatei beschwert, dann schauen Sie in `/etc/` nach der entsprechenden Datei. Wenn eine mit der Dateiendung `.new` (oder so ähnlich) existiert, entfernen Sie diese Dateiendung einfach (z.B. mit `mv`).

Paketabhängigkeitsprobleme können auftreten, wenn Pakete in `unstable/testing` installiert werden. Diese können auf unterschiedliche Art und Weise umgangen werden.

```
# apt-get install -f Paket # ignoriere falsche Abhängigkeiten
```

Eine Alternative dazu wäre, das `equivs`-Paket zu verwenden. Siehe `/usr/share/doc/equivs/README.Debian` und 'Das `equivs`-Paket' auf Seite 81.

6.3.3 Wiederherstellung mit `dpkg`

Sollte das System einmal wirklich schwer geschädigt worden sein, beispielsweise wenn `dselect` (APT) abstürzt, kann man es auch nur mit Hilfe von `dpkg` wiederherstellen:

```
# cd /var/cache/apt/archives
# dpkg -i libc6* libdb2* perl*
# dpkg -i apt* dpkg* debconf*
# dpkg -i *            # solange, bis keine Fehler mehr auftreten
```

Sollte einmal ein Paket fehlen, können Sie es von Mirror-Seiten (`http://www.debian.org/mirror/`) herunterladen:

```
# mc    # verwenden Sie "FTP-Verbindung" auf Debian-FTP-Server
```

Neue Pakete auf den HTTP/FTP-Servern befinden sich nicht mehr in dem klassischen `dist`-Verzeichnis, sondern im neuen `pool`-Verzeichnis. (Siehe 'Das `pool`-Verzeichnis' auf Seite 9.)

Die Installation des Pakets erfolgt dann folgendermaßen:

```
# dpkg -i /var/cache/apt/archives/Paketdatei.deb
```

Sollte ein Problem mit einer Abhängigkeit auftreten, kann dieses wie folgt gelöst werden:

```
# dpkg --ignore-depends=Paket1 ... -i Paketdatei.deb
# dpkg --force-depends -i Paketdatei.deb
# dpkg --force-depends --purge Paket
# dpkg --force-confmiss -i Paketdatei.deb # installiert fehlende
                                          # Konfigurationsdateien
```

6.3.4 Wiederherstellung von Paketauswahldaten

Wenn `/var/lib/dpkg/status` aus irgendeinem Grund fehlerhaft ist, so verliert das Debian-System die Paketauswahldaten und nimmt Schaden. Schauen Sie nach der alten `/var/lib/dpkg/status`-Datei `/var/lib/dpkg/status-old` oder `/var/backups/dpkg.status.*`.

Eine eigene Partition für `/var/backups/` zu erstellen, könnte eine gute Idee sein, insbesondere da sich dort wichtige Systemdaten befinden.

Wenn die alte `/var/lib/dpkg/status`-Datei nicht verfügbar ist, kann man dennoch Informationen von Verzeichnissen in `/usr/share/doc/` wiederherstellen.

```
# ls /usr/share/doc | \
  grep -v [A-Z] | \
  egrep -v '^(debian|texmf)$' | \
  awk '{print $1 " install"}' | \
  dpkg --set-selections
... neuinstallieren des Systems, abwählen je nach Bedarf:
# dselect --expert
```

6.3.5 Systemrettung nach Löschen von /var

Da das /var-Verzeichnis regelmäßig aktualisierte Daten wie Mails enthält, ist es für einen Ausfall anfällig. Das Auslagern auf eine unabhängige Partition reduziert die Risiken. Wenn ein Unglück geschieht, muss das /var-Verzeichnis wiederhergestellt werden, um das Debian-System zu retten.

Besorgen Sie sich den wichtigsten Inhalt des /var-Verzeichnisses von einem kleinen noch funktionsfähigem Debian-System, basierend auf der selben oder einer älteren Debian-Version, z.B. var.tar.gz (http://people.debian.org/~osamu/pub/) und kopieren Sie dies in das Stammverzeichnis des defekten Systems. Führen Sie nun Folgendes aus:

```
# cd /
# mv var var-alt      # wenn noch sinnvolle Daten vorhanden sind
# tar xvzf var.tar.gz # Woody's Minimaldatei verwenden
# aptitude            # oder dselect
```

Dies sollte zu einem funktionierenden System führen. Die Wiederherstellung der Paketauswahl kann beschleunigt werden, wenn man die in 'Wiederherstellung von Paketauswahldaten' auf der vorherigen Seite beschriebene Technik anwendet. (FIXME: Dieses Vorgehen benötigt mehr Versuche zur Bestätigung.)

6.3.6 Installation eines Pakets in ein nicht bootfähiges System

Booten Sie Linux von einer der Debian-Rettungs-Disketten/CDs oder von einer alternativen Partition in einem Multi-Boot-System. Vergleichen Sie 'Booten des Systems' auf Seite 95. Mounten Sie nun die nicht bootfähige Partition nach /target und verwenden Sie die chroot-Installationsmethode von dpkg.

```
# dpkg --root /target -i Paketdatei.deb
```

Anschließend sollten Sie alles nötige konfigurieren und Probleme beheben.

Übrigens, wenn lilo alles ist, was Sie davon abhält das System zu booten, können Sie auch ganz einfach von einer Debian-Rettungsdiskette booten. Am Boot-Prompt, vorausgesetzt Ihre root-Partition befindet sich auf /dev/hda12 und Sie möchten im Run-Level 3 starten, geben Sie Folgendes ein:

```
boot: rescue root=/dev/hda12 3
```

Daraufhin startet ein nahezu vollständiges System, mit dem Kernel von der Bootdiskette. (Möglicherweise werden nicht alle Hardwarekomponenten erkannt oder Module nicht geladen, dies liegt aber an dem Kernel von der Bootdiskette.)

6.3.7 Der dpkg-Befehl funktioniert nicht mehr – was nun?

Ein funktionsgestörtes `dpkg` kann es verständlicherweise unmöglich machen, beliebige `.deb`-Dateien zu installieren. Ein Vorgehen wie das unten aufgeführte wird Ihnen dabei helfen, diese Situation wieder in den Griff zu bekommen. (Natürlich können Sie `links` durch Ihrem favorisierten Browser ersetzen.)

```
$ links http://http.de.debian.org/debian/pool/main/d/dpkg/
  ... herunterladen von dpkg_version_arch.deb
$ su
password: *****
$ ar x dpkg_version_arch.deb
# mv data.tar.gz /data.tar.gz
# cd /
# tar xzfv data.tar.gz
```

Für die `i386`-Architektur kann auch `http://packages.debian.org/dpkg` als URL genutzt werden.

6.4 Debian-Nirwana-Befehle

Das **Verständnis** der folgenden Befehle ist eine essentielle Voraussetzung, um zum einem eventuellen Upgrade-Problemen gewachsen zu sein und zum anderen um das Debian-**Nirwana** zu erreichen. :-)

6.4.1 Informationen über eine Datei

Sie möchten den Namen des Pakets in Erfahrung bringen, welchem eine bestimmte Datei angehört:

```
... suche nach Muster in den installierten Paketdateien:
$ dpkg {-S|--search} Muster
... finde das Dateinamen-Muster in den Debian-Archivdateien:
$ wget http://ftp.us.debian.org/debian/dists/sarge/Contents-i386.gz
$ zgrep -e pattern Contents-i386.gz
```

Sie können auch spezialisierte Paketkommandos verwenden:

```
... dlocate:
# apt-get install dlocate
... schnellere Alternative zu dpkg -L and dpkg -S:
$ dlocate filename
```

```
... installiert Pakete automatisch auf Anfrage:
# apt-get install auto-apt
... erzeugt eine Datenbank für auto-apt:
# auto-apt update
... sucht nach einem Muster in allen Paketen, ob sie installiert
... sind oder nicht:
$ auto-apt search Muster
```

6.4.2 Informationen über ein Paket

Die Debian-Paketarchive können durchsucht und angezeigt werden. Stellen Sie jedoch vorher sicher, dass APT die richtigen Archive in `/etc/apt/sources.list` findet. Wenn Sie sehen wollen, wie sich Pakete in `testing/unstable` gegen die aktuell installierten verhalten, so verwenden Sie `apt-cache policy`. Dies ist sehr nett.

```
# apt-get check           # aktualisiert den Cache und schaut nach
                          # fehlerhaften Paketen
$ apt-cache search Muster # sucht Pakete anhand Paketbeschreibung
$ apt-cache policy Paket  # Paketprioritäten/Distributions-Infos
$ apt-cache show -a Paket # zeigt die Paketbeschreibung in allen
                          # Distributionen
$ apt-cache showsrc Paket # zeigt die Paketbeschreibung des
                          # Quellpakets
$ apt-cache showpkg Paket # zeigt Paketinformationen für Debugging
# dpkg --audit|-C         # sucht nach nicht vollständig
                          # installierten Paketen
$ dpkg {-s|--status} Paket ... # Beschreibung eines installierten
                          # Pakets
$ dpkg -l Paket ...       # Status des installierten Pakets
$ dpkg -L Paket ...       # zeigt die Dateien an, die durch das
                          # Paket installiert wurden
```

`apt-cache showsrc` ist in Woody noch nicht dokumentiert, funktioniert aber :)

Sie können ebenfalls anstelle der oben genannten Befehle Paketinformationen in folgenden Dateien finden (verwenden Sie beispielsweise `mc` zum Suchen):

```
/var/lib/apt/lists/*
/var/lib/dpkg/available
```

Ein Vergleich folgender Dateien kann Aufschluss darüber geben, was genau bei der letzten Installation passiert ist.

```
/var/lib/dpkg/status
/var/backups/dpkg.status*
```

6.4.3 Nicht-Interaktive Installation mit APT

Um Pakete nicht-interaktiv zu installieren, fügen Sie folgende Zeile zu `/etc/apt/apt.conf` hinzu:

```
Dpkg::Options {"--force-confold";}
```

Dies ist äquivalent zu `apt-get -q -y Paket`. Da dies alle Fragen automatisch mit "Ja" beantwortet, kann dies zu Problemen führen. Nutzen Sie diese Option deshalb mit Vorsicht. Lesen Sie `apt.conf(5)` und `dpkg(1)`.

Sie können jedes einzelne Paket auch später wie in 'Installierte Pakete erneut konfigurieren' auf dieser Seite beschrieben konfigurieren.

6.4.4 Installierte Pakete erneut konfigurieren

Verwenden Sie folgende Befehle, um bereits installierte Pakete erneut zu konfigurieren.

```
# dpkg-reconfigure --priority=medium Paket [...]
# dpkg-reconfigure --all   # Neukonfiguration aller Pakete
# dpkg-reconfigure locales # generiere zusätzliche locales
# dpkg-reconfigure --p=low xserver-xfree86 # konfiguriere den
                                           # X-Server neu
```

Führen Sie dies für `debconf` aus, wenn Sie den Dialogmodus von `debconf` permanent ändern möchten.

Einige Programme sind mit speziellen Konfigurationsskripten ausgestattet. [2]

```
apt-setup    - erstellt die Datei /etc/apt/sources.list
install-mbr  - installiert einen Master-Boot-Record-Manager
tzconfig     - setzt die lokale Zeitzone
gpmconfig    - konfiguriert den gpm-Maus-Daemon
eximconfig   - konfiguriert Exim (MTA)
texconfig    - konfiguriert teTeX
apacheconfig - konfiguriert Apache (httpd)
cvsconfig    - konfiguriert CVS
sndconfig    - konfiguriert das Soundsystem
...
update-alternatives - definiert Standardbefehle, z.B. vi für vim
update-rc.d         - System-V-init-Skript-Management
update-menus        - das Debian-Menüsystem
...
```

[2] Einige `*config`-Skripte verschwinden in der neueren Sarge-Ausgabe und die Paketkonfigurationsfunktionalitäten wurden nach `debconf` verschoben.

6.4.5 Deinstallation von Paketen

Entfernen eines Pakets, die Konfiguration wird nicht gelöscht:

```
# apt-get remove Paket ...
# dpkg  --remove Paket ...
```

Entfernen eines Pakets und dessen Konfiguration:

```
# apt-get remove --purge Paket ...
# dpkg    --purge        Paket ...
```

6.4.6 "Halten" älterer Pakete

Zunächst ein Beispiel: Möchte man, dass die Pakete `libc6` und `libc6-dev` beim Ausführen von `dselect` und `apt-get -u upgrade` nicht durch neuere Versionen ersetzt werden, so kann dies folgendermaßen erreicht werden:

```
# echo -e "libc6 hold\nlibc6-dev hold" | dpkg --set-selections
```

`apt-get -u install` *`Paket`* bleibt davon jedoch unbeeinflusst. Um diese Pakete andererseits bei einem automatischen Systemdowngrade mittels `apt-get -u upgrade` oder `apt-get -u dist-upgrade` zurück zu halten, ist die Datei `/etc/apt/preferences` wie folgt zu erweitern:

```
Package: libc6
Pin: release a=stable
Pin-Priority: 2000
```

An dieser Stelle kann der Eintrag "`Package:`" keine Paketnamen wie z.B. "`libc6*`" enthalten. Wenn Sie jedoch alle Pakete, die im Zusammenhang mit dem `glibc`-Quellpaket stehen, auf dem selben Stand halten möchten, müssen Sie diese alle explizit auflisten.

Folgender Befehl zeigt alle Pakete an, die den Status "hold" haben:

```
dpkg --get-selections "*" | grep -e "hold$"
```

6.4.7 Gemischtes `stable`/`testing`/`unstable`-System

`apt-show-versions` kann alle verfügbaren Paketversionen und deren Zugehörigkeit zu den einzelnen Distributionen anzeigen.

```
$ apt-show-versions | fgrep /testing | wc
... Anzahl der installierten Pakete aus testing
$ apt-show-versions -u
... Liste aller Pakete für die es aktuellere Versionen gibt
$ apt-get install `apt-show-versions -u -b | fgrep /unstable`
... aktualisiert alle Paket aus unstable auf die aktuellste Version
```

6.4.8 Löschen von zwischengespeicherten Paketdateien

APT speichert alle heruntergeladenen Paketdateien in `/var/cache/apt/archives/`. Diese können folgendermaßen gelöscht werden:

```
# apt-get autoclean # löscht nur nicht benötigte Paketdateien
# apt-get clean     # löscht alle zwischengespeicherten Paketdateien
```

6.4.9 Speichern/Kopieren der Systemkonfiguration

Erstellen einer lokalen Kopie der aktuellen Paketauswahl:

```
# debconf-get-selections > debconfsel.txt
# dpkg --get-selections "*" >meineAuswahl   # oder \*
```

`"*"` veranlasst, dass auch Pakete mit einem Status von "purge" in *`meineAuswahl`* gespeichert werden.

Diese Datei kann auf einen anderen Computer kopiert werden, um dort die gleichen Pakete zu installieren:

```
# dselect update
# debconf-set-selections  < debconfsel.txt
# dpkg --set-selections <meineAuswahl
# apt-get -u dselect-upgrade    # oder dselect install
```

6.4.10 Portierung eines Pakets auf die `stable`-Distribution

Um selektive Upgrades in der stabilen Distribution zu machen, ist es möglich, ein Quellpaket in dieser Umgebung zu kompilieren. Dies verhindert viele Paketaktualisierungen, die durch Abhängigkeiten verursacht würden. Zuerst müssen folgende Zeilen der `/etc/apt/sources.list`-Datei hinzugefügt werden:

```
deb-src http://http.us.debian.org/debian testing \
 main contrib non-free
deb-src http://http.us.debian.org/debian unstable \
 main contrib non-free
```

Aus Formatierungsgründen sind die Zeilen hier in je zwei Zeilen umgebrochen. Natürlich muss aber jede mit "deb-src" beginnende Zeile für sich alleine in `sources.list` stehen.

Nun ist der Quellcode zu beziehen und ein lokales Paket zu erstellen:

```
$ apt-get update  # aktualisiere die Quellpaketsuchliste
$ apt-get source Paket
$ dpkg-source -x Paket.dsc
$ cd paket-version
  ... schauen Sie nach benötigten Paketen (Build-depends in der
  ... .dsc-Datei) und installieren Sie diese ebenso wie fakeroot

$ dpkg-buildpackage -rfakeroot

  ... oder (keine Signatur erstellen)
  ... verwenden Sie später "debsign" wenn nötig
$ dpkg-buildpackage -rfakeroot -us -uc

  ... Anschließend die Installation des Pakets
$ su -c "dpkg -i Paketdatei.deb"
```

Gewöhnlich ist es nötig, einige Pakete mit der "-dev"-Erweiterung zu installieren, um Paketabhängigkeiten auflösen zu können. `debsign` ist im Paket `devscripts` enthalten. `auto-apt` kann die Auflösung von Abhängigkeiten erleichtern. Mittels `fakeroot` lässt sich unnötige Benutzung des root-Accounts vermeiden.

In der Woody-Distribution lassen sich diese Abhängigkeitsprobleme vereinfachen. Als ein Beispiel kompilieren wir `pine`, das nur im Quellcode vorliegt.

```
# apt-get build-dep pine
# apt-get source -b pine
```

6.4.11 Lokale Paketarchive

Möchte man ein lokales Paketarchiv erzeugen, welches mit APT und `dselect` kompatibel ist, so muss die Datei `Packages` erstellt werden und Paketdateien müssen in einem bestimmten Verzeichnis erstellt werden.

Ein lokales `deb`-Archiv (Repository) das ähnlich zum offiziellen Debian-Archiv ist, kann wie folgt erstellt werden:

```
# apt-get install dpkg-dev
# cd /usr/local
# install -d pool # die Paketdateien werden hier gespeichert
# install -d dists/unstable/main/binary-i386
```

```
# ls -1 pool | sed 's/_.*$/ priority section/' | uniq > override
# editor override # anpassen von priority und section
# dpkg-scanpackages pool override /usr/local/ \
   > dists/unstable/main/binary-i386/Packages
# cat > dists/unstable/main/Release << EOF
Archive: unstable
Version: 3.0
Component: main
Origin: Local
Label: Local
Architecture: i386
EOF
# echo "deb file:/usr/local unstable main" \
   >> /etc/apt/sources.list
```

Alternativ kann auch folgende, schnellere aber unsaubere, Methode verwendet werden:

```
# apt-get install dpkg-dev
# mkdir /usr/local/debian
# mv /irgend/ein/Paket.deb /usr/local/debian
# dpkg-scanpackages /usr/local/debian /dev/null | \
  gzip - > /usr/local/debian/Packages.gz
# echo "deb file:/usr/local/debian ./" >> /etc/apt/sources.list
```

Auf ein solches Archiv kann auch von entfernten Computern zugegriffen werden, indem der Zugriff auf diese Verzeichnisse durch HTTP oder FTP ermöglicht wird. Die Datei `/etc/apt/sources.list` muss dementsprechend angepasst werden.

6.4.12 Installation und Konvertierung eines fremden Binärpakets

Mit Hilfe des `alien`-Pakets lassen sich binäre Pakete, wie sie in vielen Distributionen wie Red Hat `rpm`, Stampede `slp`, Slackware `tgz` und Solaris `pkg` verwendet werden, in das Debian `deb`-Paketformat konvertieren. Wenn Sie ein Paket von einer anderen Distribution verwenden möchten, können Sie es mit `alien` in Ihr bevorzugtes Paketformat konvertieren und anschließend installieren. `alien` unterstützt ebenfalls LSB-Pakete.

6.4.13 Verifizierung installierter Paketdateien

`debsums` verifiziert installierte Paketdateien anhand deren MD5 Prüfsummen. Jedoch gibt es auch Pakete die keine MD5-Prüfsummen enthalten. Eine mögliche temporäre Lösung für Systemadministratoren ist folgende:

```
# cat >>/etc/apt/apt.conf.d/90debsums
DPkg::Post-Install-Pkgs {"xargs /usr/bin/debsums -sg";};
^D
```

von Joerg Wendland <`joergland@debian.org`> (nicht getestet).

6.4.14 Optimierte `sources.list`

Um es kurz zu machen, raffinierte Anstrengungen um `sources.list` optimal zu erzeugen führte nicht zu einer signifikanten Verbesserungen für mich in den USA. Ich wähle manuell einen nahe gelegenen Server mittels `apt-setup`.

`apt-spy` erzeugt `sources.list` automatisch, basierend auf der Wartezeit und der Bandbreite. `netselect-apt` erzeugt eine komplexere `sources.list`, nutzt aber eine einfachere Methode zur Bestimmung des besten Spiegels (ping Zeitvergleich).

```
# apt-get install apt-spy
# cd /etc/apt; mv sources.list sources.list.org
# apt-spy -d testing -l sources.apt
```

6.5 Weitere Debian-Besonderheiten

6.5.1 Der `dpkg-divert`-Befehl

Datei**umleitungen** ist eine Methode, um `dpkg` dazu zu bringen, Dateien nicht in ihre Standardverzeichnisse zu installieren, sondern in **umgeleitete** Verzeichnisse. **Umleitungen** können durch Debian-Skripte dazu genutzt werden, Dateien zu verschieben, wenn sie einen Konflikt verursachen. Systemadministratoren können auch eine Umleitung verwenden, um die Konfigurationsdatei eines Pakets zu überschreiben, oder wann immer einige Dateien (welche nicht als **conffiles** markiert sind) von `dpkg` nicht überschrieben werden sollen, wenn eine neuere Version eines Pakets installiert wird, das solche Dateien enthält (siehe 'Bewahren der lokalen Konfiguration' auf Seite 14).

```
# dpkg-divert [--add]  Datei # erstellen der "Umleitung"
# dpkg-divert --remove Datei # entfernen der "Umleitung"
```

Für gewöhnlich ist es ratsam `dpkg-divert` nicht zu verwenden wenn es nicht absolut notwendig ist.

6.5.2 Das `equivs`-Paket

Wenn Sie ein Quellpaket kompilieren ist es empfehlenswert, es zu einem echten lokalen Debianpaket (`*.deb`) zu machen. Benutzen Sie `equivs` als letzten Ausweg.

```
Package: equivs
Priority: extra
```

```
Section: admin
Description: Circumventing Debian package dependencies
 This is a dummy package which can be used to create Debian
 packages, which only contain dependency information.
```

6.5.3 Alternative Befehle

Möchte man dass der Befehl `vi` `vim` ausführt, benutzen Sie `update-alternatives`:

```
# update-alternatives --display vi
...
# update-alternatives --config vi
  Selection    Command
-----------------------------------------------
      1        /usr/bin/elvis-tiny
      2        /usr/bin/vim
*+    3        /usr/bin/nvi

Enter to keep the default[*], or type selection number: 2
```

Einstellungen im Debian-Alternativsystem werden in `/etc/alternatives/` als symbolische Links gespeichert.

Um Ihre bevorzugte X-Window-Umgebung einzustellen, führen sie `update-alternatives` für `/usr/bin/x-session-manager` und `/usr/bin/x-window-manager` aus. Für weitere Informationen, siehe 'Angepasste X-Sitzung' auf Seite 141.

`/bin/sh` ist ein symbolischer Link auf `/bin/bash` oder `/bin/dash`. Aus Kompatibilitätsgründen zu alten Bash-verunreinigten Skripten ist es sicherer, `/bin/bash` zu verwenden. Um die POSIX-Konformität einzuhalten sollte jedoch `/bin/dash` verwendet werden. Mit dem Upgrade auf den 2.4er-Linux-Kernel geht die Tendenz zugunsten zu `/bin/dash`.

6.5.4 `init` und Runlevel

Der Standardrunlevel in dem gebootet wird, lässt sich in `/etc/inittab` festlegen.

In Debian ist es dem Systemadministrator möglich, einzustellen, welche Dienste in welchem Runlevel laufen sollen. Das `update-rc.d`-Werkzeug kann verwendet werden, um diese Einstellungen zu ändern, unabhängig davon, ob die `sysv-rc`- oder die `file-rc`-Methode zum Kontrollieren von Diensten verwendet wird.

Um zum Beispiel den Dienst *name* (mit dem Init-Skript `/etc/init.d/name`) mit der Sequenznummer 20 in den Runlevel 2, 3 und 4 zu starten und in Runlevel 5 mit der Sequenznummer 80 zu stoppen, ist Folgendes auszuführen:

```
# update-rc.d name start 20 2 3 4 . stop 80 5 .
```

Wenn Sie das `sysv-rc` Paket benutzen, gibt es einen anderen Weg, solche Änderungen durchzuführen: Benennen Sie einfach die symbolischen Links in den `/etc/rc?.d` Verzeichnissen um.

```
# mv S99xdm K01xdm # xdm deaktivieren (X-Display-Manager)
```

6.5.5 Deaktivierung von Daemon-Diensten

Die Debian-Distribution nimmt die Systemsicherheit sehr ernst und erwartet, dass der Systemadministrator kompetent ist. Eine einfache Administration steht so oft nur an zweiter Stelle und viele Daemon-Dienste sind in der Standard-Installation mit dem höchsten Sicherheitslevel vorkonfiguriert oder mit den wenigsten Diensten (oder keinen) ausgestattet.

Führen Sie `ps aux` aus oder überprüfen Sie die Inhalte von `/etc/init.d/*` und `/etc/inetd.conf`, wenn Sie irgenwelche Zweifel (zu Exim, DHCP, . . .) haben. Überprüfen Sie auch `/etc/hosts.deny` wie in 'Zugriffskontrolle mit PAM und login' auf Seite 127. Der `pidof`-Befehl ist ebenfalls hilfreich (siehe `pidof(8)`).

X11 erlaubt standardmäßig keine (entfernten) TCP/IP-Verbindungen in neueren Debian-Versionen. Schauen Sie in 'TCP/IP-Verbindungen mit X' auf Seite 144. X-Weiterleitung mittels SSH ist ebenfalls deaktiviert. Vergleichen Sie mit 'X-Fernverbindungen mit `ssh`' auf Seite 144.

Kapitel 7

Der Linux-Kernel unter Debian

Debian hat eine eigene Methode einen Kernel und dazugehörige Module zu kompilieren. Weitere Infos sind in 'Debian und der Kernel' auf Seite 25 zu finden.

7.1 Neukompilieren des Kernels

Die Tools `gcc`, `binutils` und `modutils` aus Debian `unstable` sind sehr hilfreich, wenn man einen aktuellen Linux-Kernel kompilieren möchte. Siehe `/usr/share/doc/kernel-package/README.gz` um offizielle Informationen zu erhalten, besonders der untere Teil ist interessant.

Einen eigenen Kernel zu kompilieren ist eine nicht ganz einfache Sache, womit selbst begnadete Entwickler Schwierigkeiten haben könnten, gerade auch weil diese Thematik eine sich ständig weiter entwickelnde ist.

Manoj Srivastava schrieb: `--initrd` benötigt einen Debian-eigenen cramfs-Patch.

Herbert Xu schrieb: Nein, das tut es nicht. Das einzige was nötig ist, um ein anderes Dateisystem als CRAMFS zu nutzen, ist, MKIMAGE in `/etc/mkinitrd/mkinitrd.conf` zu setzen.

Seien Sie vorsichtig und ziehen Sie `/usr/share/doc/kernel-package/README.gz` von Manjo und Kent stets zu Hilfe. Des Weiteren stellen Sie sicher, dass Sie immer die aktuellste Version des unstable `kernel-package`-Paketes haben, wenn Sie einen instabilen Kernel kompilieren möchten.

Für einen Kernel, der nur für eine Maschine kompiliert ist, wird `initrd` nicht benötigt. Weil ich aber möchte, dass mein Kernel nahezu der gleiche ist, wie der, der von den kernel-image-Paketen bereitgestellt wird, nutze ich es trotzdem. Wenn Sie `initrd` nutzen möchten, lesen Sie bitte `mkinitrd(8)` und `mkinitrd.conf(5)`. Siehe auch `http://bugs.debian.org/149236`.

7.1.1 Die Debian-Standardmethode

Beachten Sie Fehlermeldungen für `kernel-package`, `gcc`, `binutils` und `modutils`. Benutzen Sie neue Versionen von diesen Programmen wenn nötig.

Einen angepassten Kernel unter Debian zu kompilieren, bedarf ein wenig Sorgfalt. Verwenden Sie die Option `--append_to_version` zusammen mit `make-kpkg`, um mehrere Kernel zu verwalten.

```
# apt-get install debhelper modutils kernel-package libncurses5-dev
# apt-get install kernel-source-2.4.18 # installiere die aktuellste Version
# apt-get install fakeroot
# vi /etc/kernel-pkg.conf            # fügen Sie Namen und E-Mail-Adresse hinzu
$ cd /usr/src                        # Arbeitsverzeichnis
$ tar --bzip2 -xvf kernel-source-2.4.18.tar.bz2
$ cd kernel-source-2.4.18            # wenn das die Kernelquellen sind
$ cp /boot/config-2.4.18-386 .config # verwende angegebene Konfiguration
$ make menuconfig                    # den Anforderungen entsprechend zusammenstellen
$ make-kpkg clean                    # nötig laut 'man make-kpkg'
$ fakeroot make-kpkg --append_to_version -486 --initrd \
      --revision=rev.01 kernel_image \
      modules_image # modules_image ist für pcmcia-cs* und weitere ...
$ cd ..
# dpkg -i kernel-image*.deb pcmcia-cs*.deb # installiert den Kernel
```

`make-kpkg kernel_image` führt `make oldconfig` und `make dep` aus. Verwenden Sie `--initrd` nicht, wenn initrd nicht ausgewählt wurde.

Wenn Kernel-Module aus pcmcia-cs verwendet werden sollen oder aber keine Module der pcmcia-Kernelquellen benötigt werden, sollte im `make menuconfig`-Dialog unter “General setup” der Eintrag “PCMCIA/CardBus support” ausgewählt und die Konfiguration auf “**< >** PCMCIA/CardBus support” gesetzt werden, beispielsweise durch Deaktivieren der Box.

Auf einer SMP-Maschine, setzen Sie CONCURRENCY_LEVEL der `kernel-pkg.conf(5)` entsprechend.

7.1.2 Klassische Methode

Laden Sie den “reinen” Quellcode von
- Linux: `http://www.kernel.org/`
- pcmcia-cs: `http://pcmcia-cs.sourceforge.net/`

herunter oder verwenden Sie einen äquivalenten Quellcode in Debian und führen Sie folgende Schritte aus:

```
# cd /usr/src
# tar xfvz linux-Version.tar.gz
```

```
# rm -rf linux
# ln -s linux-Version linux
# tar xfvz pcmcia-cs-Version.tar.gz
# ln -s pcmcia-cs-Version pcmcia
# cd linux
# make menuconfig
... den Anforderungen entsprechend konfigurieren ...
# make dep
# make bzImage
... Konfiguration von lilo / grub ...
... kopieren von /usr/src/linux/arch/i386/boot/bzImage nach /boot ...
... /sbin/lilo oder grub ausführen
# make modules; make modules_install
# cd ../pcmcia
# make config
# make all
# make install
... Hinzufügen von benötigten Modulen zu /etc/modules
# shutdown -r now
... nun den neuen Kernel booten ...
```

7.1.3 Kernelheader

Die meisten Programme brauchen die Kernelheader-Dateien nicht, und einige Programme lassen sich sogar nicht erfolgreich kompilieren, wenn die Header-Dateien genutzt werden. Stattdessen sollten die Programme beim Kompilieren gegen die Header-Dateien gelinkt werden, gegen die auch **`glibc` gelinkt wurde**. Auf einem Debian-System sind diese Kernelheader-Dateien in `/usr/include/linux` und `/usr/include/asm` zu finden.

Entgegen einigen veralteten Dokumentationen, sollten keine symbolischen Links in `/usr/src/linux` von `/usr/include/linux` und `/usr/include/asm` erstellt werden.

Es sollte, wenn ein Programm bestimmte Kernelheader-Dateien für Kernelspezifische Programme benötigt, stattdessen das Makefile so angepasst werden, dass der Pfad der einzubindenden Header-Dateien folgende Pfade enthält: `Verzeichnis-bestimmter-Kernelheader/include/linux` und `Verzeichnis-bestimmter-Kernelheader/include/asm`.

7.2 Der modularisierte Kernel 2.4

Die neuen Debian-2.4er-Kernel, welche von `kernel-image-2.4.NN` zur Verfügung gestellt werden, sind sehr stark modularisiert. Es muss sichergestellt sein, dass Module die verwendet werden sollen auch aktiviert sind, damit der Kernel wie gewünscht arbeitet.

Obwohl ich viele Beispiele für `/etc/modules` im folgenden Abschnitt als schnellen Fix habe, höre ich immer wieder, dass der richtige Weg, diese Modul abhängigen Probleme zu lösen, ist, Aliase für die Geräte in Dateien unterhalb von /etc/modutils/ anzulegen, da es genug davon in aktuellen Kernel gibt. Einige Module könnten automatisch von Hardwareerkennungsprogrammen wie discover aktiviert werden. Vergleichen Sie auch 'Hardware-Erkennung für X' auf Seite 137.

Siehe 'Spezielle Regeln für den Umgang mit Modulen' auf Seite 26 und Documentation /*.txt im Linux-Quellcode, um genauere Informationen zu erhalten.

7.2.1 PCMCIA

/etc/modules muss eventuell Folgendes enthalten, damit alte PCMCIA-Hardware korrekt funktioniert:

```
# ISA PnP Treiber
isa-pnp
# neuer lowlevel PCMCIA-Treiber
# yenta_socket # in meinem Fall jedoch nicht nötig
```

Für den Rest von PCMCIA sorgen die Skripte (aus dem pcmcia-cs Paket) depmod und kmod. isa-pnp brauchte ich auch, weil mein Laptop ein altes ISA-PCMCIA Modell ist. Neuere Laptops mit Cardbus/PCMCIA sollten dies nicht mehr brauchen.

Ein Zitat vom genialen Miquel van Smoorenburg <miquels@cistron.nl>:

"Ich habe einfach alles was mit PCMCIA zu tun hat, inklusive cardmgr, von meinem Laptop gelöscht und einen 2.4er-Kernel mit CardBus-Unterstützung installiert, sowie das neue hotplug-Paket aus Woody.

Solange man nur 32-bit Karten hat, wird das pcmcia-Paket nicht benötigt; 2.4 stellt die nötigen Karten-Dienste von sich aus zur Verfügung. Und der Standard-tulip-Treiber sollte wunderbar mit der Dlink-Karte arbeiten.

– Mike."

Sehen Sie dazu auch das Linux PCMCIA-HOWTO (http://www.tldp.org/HOWTO/PCMCIA-HOWTO.html) und 'Netzwerk-Konfiguration und PCMCIA' auf Seite 186.

7.2.2 SCSI

[NICHT GETESTET] /etc/modules braucht Folgendes damit SCSI funktioniert:

```
# SCSI core
scsi_mod
# SCSI generic driver
sg
# SCSI disk
```

```
sd_mod
# Alle anderen HW Module
...
```

`depmod` sollte auf die oben genannten Module achten.

7.2.3 Netzwerk-Funktionalität

`/etc/modules` sollte für zusätzliche Netzwerk-Funktionalität Folgendes enthalten:

```
# net/ipv-4
ip_gre
ipip

# net/ipv-4/netfilter
# iptable (in Reihenfolge)
ip_tables
ip_conntrack
ip_conntrack_ftp
iptable_nat
iptable_filter
iptable_mangle
#
ip_nat_ftp
ip_queue
#
ipt_LOG
ipt_MARK
ipt_MASQUERADE
ipt_MIRROR
ipt_REDIRECT
ipt_REJECT
ipt_TCPMSS
ipt_TOS
ipt_limit
ipt_mac
ipt_mark
ipt_multiport
ipt_owner
ipt_state
ipt_tcpmss
ipt_tos
ipt_unclean
#
```

```
#ipchains
#ipfwadm
```

Oben genanntes ist nicht optimiert. `depmod` sollte auf die oben genannten Module achten.

7.2.4 EXT3-Dateisystem (> 2.4.17)

Die Journaling Funktionalität für das EXT3-Dateisystem mit einem vorkompilierten Kernel-Image (> 2.4.17) von Debian zu aktivieren, setzt die folgenden Schritte voraus:

```
# cd /etc; mv fstab fstab.old
# sed 's/ext2/ext3,ext2/g' <fstab.old >fstab
# vi /etc/fstab
... setzen Sie den root-Dateisystemtyp auf "auto" anstelle von "ext3,ext2"
# cd /etc/mkinitrd
# echo jbd >>modules
# echo ext3 >>modules
# echo ext2 >>modules
# cd /
# apt-get update; apt-get install kernel-image-2.4.17-686-smp
... installieren Sie den aktuellsten Kernel und konfigurieren Sie LILO
dementsprechend
# tune2fs -j -i 0 /dev/hda1
# tune2fs -j -i 0 /dev/hda2
... tune2fs für alle EXT2 Dateisysteme wiederholen
# shutdown -r now
```

Von nun an ist EXT3-Journaling aktiviert. Für den Fall, dass der Kernel EXT3 für root-Partitionen nicht unterstützt, erweist sich `ext3,ext2` als Dateisystem-"Typ" in der `fstab` als sehr hilfreich, da so sicher ein "Fall-Back" gemacht werden kann.

Wenn Sie schon zuvor einen 2.4er-Kernel installiert haben, und ihn nicht erneut installieren möchten, dann führen Sie die oben genannten Befehle bis zu dem `apt-get` Kommando aus, und dann folgende:

```
# mkinitrd -o /boot/initrd.img-2.4.17-686-smp
/lib/modules/2.4.17-686-smp
# lilo
# tune2fs -j -i 0 /dev/hda1
# tune2fs -j -i 0 /dev/hda2
... tune2fs für alle EXT2-Dateisysteme wiederholen
# shutdown -r now
```

Von nun an ist EXT3-Journaling aktiviert.

Wenn `/etc/mkinitrd/modules` nicht gesetzt war als `mkinitrd` ausgeführt wurde, und Sie während der Boot-Phase zusätzliche Module hinzufügen möchten:

```
... um eine Shell (5 sec.) an dem initrd Prompt zu erhalten, RETURN drücken
# insmod jbd
# insmod ext3 # modprobe ext3 sollte alles nötige berücksichtigen
# insmod ext2
# ^D
... weiter gehts mit dem Boot-Vorgang
```

Die Meldung "cramfs: wrong magic" könnte in den Systemmeldungen beim Boot-Vorgang (`dmesg`) auftreten, diese wird jedoch als harmlos eingestuft. Dieses Problem wurde in Sarge (2002/10) gelöst. Siehe `http://bugs.debian.org/135537` und das EXT3 File System mini-HOWTO (`http://www.zip.com.au/~akpm/linux/ext3/ext3-usage.html`) oder `/usr/share/doc/HOWTO/en-txt/mini/extra/ext3-mini-HOWTO.gz` um weitere Informationen zu erhalten.

Es wurde berichtet, dass auf einigen Systemen einige Kernel lock-ups auftraten wenn EXT3 aktiviert war, aber was meinen 2.4.17 Kernel angeht, hatte ich keine Probleme.

7.2.5 Realtek RTL-8139 Unterstützung in Kernel 2.4

Aus irgendwelchen Gründen heißt das RTL-8139 Modul ab dem 2.4er-Kernel 8139too und nicht mehr rtl8139. Wenn Sie also von einem 2.2er Kernel auf einen 2.4er upgraden, ändern Sie einfach die `/etc/modules` dementsprechend ab.

7.2.6 Parallel-Port-Unterstützung

Von `kernel-image-2.4.*` wird Parallel-Port-Unterstützung als Modul angeboten. Aktivieren Sie es wie folgt:

```
# modprobe lp
# echo lp >> /etc/modules
```

Siehe auch `Documentation/parport.txt` im Linux-Quellcode.

7.3 Tunen des Kernels mittels des proc-Dateisystems

Das Verhalten des Linux-Kernels kann dynamisch durch das proc-Dateisystem geändert werden.

Für grundlegende Informationen über das Ändern von Kernelparametern durch das `/proc`-Dateisystem, lesen Sie im Linux-Quellcodeverzeichnis `Documentation/sysctl/*`.

Einige Beispiele der Kernelparametermanipulation können in /etc/init.d/networking und 'Eigenartige Probleme beim Zugriff auf einige Webseiten' auf Seite 46 gefunden werden.

Sehen Sie in sysctl.conf(5) wie man den Kernel zur Bootzeit mittels des /proc-Dateisystems konfigurieren kann. Dazu wird das Skript /etc/init.d/procps.sh genutzt, das in der Regel aus /etc/rcS.d/S30procps.sh gestartet wird.

7.3.1 Zu viele geöffnete Dateien

Der Linux-Kernel könnte unter Umständen die Meldung "Too many open files" ("Zu viele geöffnete Dateien") ausgeben. Der Grund hierfür ist, dass der Standardwert (8096) für file-max sehr klein gewählt wurde. Um dieses Problem zu lösen, führen Sie folgende Schritte als root aus:

```
# echo "65536"  > /proc/sys/fs/file-max  # für 2.2er- und 2.4er-Kernel
# echo "131072" > /proc/sys/fs/inode-max # nur für 2.2er-Kernel
```

Alternativ können Sie Folgendes in /etc/sysctl.conf eintragen, um diese Änderungen permanent zu aktivieren:

```
file-max=65536   # für 2.2er- und 2.4er-Kernel
inode-max=131072 # nur für 2.2er-Kernel
```

7.3.2 Disk-Flush-Intervalle

Sie können Disk-Flush-Intervalle durch das proc-Dateisystem ändern. Das Folgende kürzt das Intervall von standardmäßig fünf Sekunden auf eine Sekunde.

```
# echo "40 0 0 0 100 30000 60 0 0"  > /proc/sys/vm/bdflush
```

Dies kann die Ein-/Ausgabe-Performance ein klein wenig negativ beeinflussen. Aber dies sichert den Dateiinhalt mit Ausnahme der letzten Sekunde, was kürzer ist als die standardmäßigen fünf Sekunden. Dies ist selbst für die Journaling-Dateisysteme wahr.

7.3.3 Trägheit alter Maschinen mit wenig Speicher

Für einige alte Systeme mit wenig Speicher kann es immer noch sinnvoll sein, over-commit von Speicher durch das proc-Dateisystem zu aktivieren:

```
# echo 1 > /proc/sys/vm/overcommit_memory
```

7.4 Der 2.6er Kernel mit udev

udev ist ein dynamischer Ersatz für `/dev/`. Gerätenamen können sehr kurz gewählt werden. devfs, das in Kernel 2.4 verwendet wurde, ist nun überholt.

Installation des neuen Debian 2.6er Kernels, der von `kernel-image-2.6.`*`NN`* bereitgestellt wird, zusammen mit dem `udev`-Paket aktiviert es.

Kapitel 8

Debian-Tipps

8.1 Booten des Systems

Detaillierte Informationen über den Boot-Prompt (also die Eingabeaufforderung beim Systemstart) gibt es im BootPrompt-HOWTO (`http://www.tldp.org/HOWTO/BootPrompt-HOWTO.html`) des LDP.

8.1.1 "Ich habe das root-Passwort vergessen!" (1)

Sobald man Zugriff auf die Tastatur eines Rechners hat, kann das System gestartet werden und ein Login stattfinden, auch ohne das root-Passwort zu kennen. (Dies setzt allerdings voraus, dass keine zusätzlichen Schutzmaßnahmen ergriffen wurden, wie z.B. ein BIOS-Passwort oder eine Passwortabfrage durch `lilo`, was ein Booten des Systems verhindern würde.)

Dies ist ein Vorgehen, das keine externen Boot-Medien und Änderungen der BIOS-Einstellungen benötigt. Die Bezeichnung für die voreingestellte Boot-Option sei "Linux".

Sobald die `lilo`-Boot-Meldung `boot:` erscheint (auf einigen Systemen muss die Shift-/Umschalt-Taste gedrückt werden, um einen automatischen Bootvorgang zu unterbrechen und wenn `lilo` den Framebuffer verwendet, muss TAB gedrückt werden, um die eingegebenen Optionen zu sehen), ist Folgendes einzugeben:

```
boot: Linux init=/bin/sh
```

Das System startet nun den Kernel und daraufhin das Programm `/bin/sh` anstelle des üblichen `init`-Prozesses. Die nun erlangte Shell hat root-Privilegien. Da das Hauptverzeichnis `/` nur les- aber nicht schreibbar ist und viele während des üblichen Boot-Prozesses eingeklinkten Verzeichnisse nicht erreichbar sind, müssen diese nachträglich gemountet werden, um ein vernünftiges Arbeiten zu ermöglichen:

```
init-2.03# mount -n -o remount,rw /
```

```
init-2.03# mount -avt nonfs,noproc,nosmbfs
init-2.03# cd /etc
init-2.03# vi passwd
init-2.03# vi shadow
```

(Wenn in der Datei `/etc/passwd` das zweite Datenfeld bei allen Einträgen ein "x" ist, dann benutzt das System shadow-Passwörter. In diesem Fall muss statt `/etc/passwd` die Datei `/etc/shadow` editiert werden.) Um das root-Passwort zurückzusetzen, muss das zweite Feld des root-Eintrags gelöscht werden. Nach einem Neustart ist das root-Passwort leer und kann neu vergeben werden. Wenn das System in den Runlevel 1 fährt, wird (zumindest bei Debian-Versionen nach Potato) ein Passwort benötigt.

Ein Editor im `/bin/`-Verzeichnis ist absolut empfehlenswert, für den Fall, dass der `/usr/`-Baum nicht erreichbar ist (siehe dazu auch 'Rettungseditoren' auf Seite 195).

Für Notfälle hat sich außerdem `sash` (stand alone shell) bewährt. Wenn das System nicht mehr gebootet werden kann, geben Sie am `lilo`-Prompt Folgendes ein:

```
boot: Linux init=/bin/sash
```

Das Programm `sash` dient als Ersatz für `sh`, selbst wenn `/bin/sh` nicht nutzbar ist. Es ist statisch gelinkt und hat viele Standardbefehle eingebaut. (Wenn `sash` gestartet ist, wird "help" eine kurze Referenz anzeigen.)

8.1.2 "Ich habe das root-Passwort vergessen!" (2)

Starten Sie das System mit einer Rettungsdiskette/-CD. Sei */dev/hda3* die Root-Partition. Dann kann die Passwortdatei editiert werden durch:

```
# mkdir fixit
# mount /dev/hda3 fixit
# cd fixit/etc
# vi shadow
# vi passwd
```

Der Vorteil dieser Variante ist, dass ein eventuell vergebenes `lilo`-Passwort umgangen werden kann. Erforderlich ist allerdings, dass im BIOS die Einstellung für das Boot-Laufwerk manipuliert werden kann (falls es nicht schon entsprechend eingestellt ist).

8.1.3 Das System kann nicht gestartet werden

Dies ist kein Problem, selbst wenn Sie keine Boot-Diskette während der Installation angelegt haben. Wenn `lilo` defekt ist, können Sie die Boot-Disk des Debian-Installationssets verwenden und davon Ihr System booten. Wenn Ihre Root-Partition */dev/hda12* ist und im Runlevel 3 gestartet werden soll, geben Sie Folgendes am Boot-Prompt ein:

```
boot: rescue root=/dev/hda12 3
```

Danach haben Sie ein fast vollständig funktionsfähiges System mit dem Kernel des Installationsmediums gestartet. (Es kann einige kleinere Probleme aufgrund fehlender Kerneleigenschaften oder Module geben.)

Vergleichen Sie 'Installation eines Pakets in ein nicht bootfähiges System' auf Seite 73 wenn Sie ein defektes System haben.

Benötigen Sie eine angepasste Boot-Diskette, folgen Sie den Anweisungen in `readme.txt` auf der Rettungsdiskette.

8.1.4 Abschalten von X beim Booten

Dem Nachjagen von `unstable/sid` ist lustig, aber ein fehlerhaftes `xdm`, `gdm`, `kdm` oder `wdm`, das während des Boot-Prozesses gestartet wird, kann sehr unangenehm sein.

Zuerst sollten Sie Zugang zu einer root-Shell erlangen, indem Sie Folgendes am Boot-Prompt eingeben:

```
boot: Linux vga=normal s
```

Hier steht *Linux* für das Kernel-Image, das Sie starten; "vga=normal" stellt sicher, dass `lilo` im normalen VGA-Modus startet und "s" (oder "S") ist der Parameter der an `init` weitergegeben wird und für Single-User-Modus steht. Geben Sie das root-Passwort am Prompt ein.

Es gibt mehrere Möglichkeiten alle Daemonen die X starten zu deaktivieren:

- Starten von `update-rc.d -f ?dm remove; update-rc.d ?dm stop 99 1 2 3 4 5 6 .`
- Einfügen von "exit 0" am Anfang der `/etc/init.d/?dm`-Dateien.
- Umbenennen aller `/etc/rc2.d/S99?dm`-Dateien in `/etc/rc2.d/K99?dm`.
- Entfernen aller `/etc/rc2.d/S99?dm`-Dateien.
- Starten von `:>/etc/X11/default-display-manager`

Hier muss die Nummer in `rc2.d` dem Runlevel entsprechen, das in der Datei `/etc/inittab` angegeben wurde. `?dm` bedeutet, dass das Kommando mehrfach ausgeführt wird, wobei es jeweils durch `xdm`, `gdm`, `kdm` und `wdm` ersetzt wird.

Nur der erste Weg ist "der eine wahre Weg" in Debian. Der letzte ist einfach, funktioniert aber nur in Debian und erfordert, dass der Display-Manager später mit `dpkg-reconfigure` gesetzt wird. Alles andere sind allgemeine Methoden, um Daemonen zu deaktivieren.

Sie können immer noch X mit dem Kommando `startx` von einer beliebigen Konsole starten.

8.1.5 Weitere Tricks mit dem Boot-Prompt

Mittels des `lilo`-Boot-Prompts kann das System in einem beliebigen Runlevel gestartet werden. Details dazu finden sich im BootPrompt-HOWTO (`http://www.tldp.org/HOWTO/BootPrompt-HOWTO.html`) des LDP.

Soll das System im Runlevel 4 starten, verwenden Sie folgende Eingabe am `lilo`-Boot-Prompt.

```
boot: Linux 4
```

Wenn das System im normal funktionierendem Einzelnutzer-Modus gestartet werden soll und Sie das root-Passwort kennen, so kann eins der folgenden Beispiele am `lilo`-Boot-Prompt verwendet werden.

```
boot: Linux S
boot: Linux 1
boot: Linux -s
```

Auch der benutzte Speicherbereich kann eingeschränkt werden. Das folgende Beispiel weist das System an, nur 48MB des verfügbaren Speichers zu belegen.

```
boot: Linux mem=48M
```

Achtung: Wird an dieser Stelle mehr Speicher angegeben als vorhanden, wird der Kernel abstürzen. Wenn man mehr als 64 MB Speicher hat, kann es bei alten Kernels und/oder Mainboards passieren, dass das System nur 64MB nutzt. In diesem Fall kann man versuchen den ungenutzten Speicher mit `mem=128M` zu aktivieren. Äquivalent dazu ist ein Eintrag in der Datei `/etc/lilo.conf`.

8.1.6 Setzen von GRUB-Boot-Parametern

GRUB ist ein neuer Boot-Manager des GNU Hurd-Projekts und ist viel flexibler als Lilo, handhabt aber die Boot-Parameter etwas anders.

```
grub> find /vmlinuz
grub> root (hd0,0)
grub> kernel /vmlinuz root=/dev/hda1
grub> initrd /initrd
grub> boot
```

Sie müssen die Hurd-Gerätenamen kennen:

```
Hurd/GRUB           Linux               MSDOS/Windows
 (fd0)               /dev/fd0            A:
 (hd0,0)             /dev/hda1           C: (normalerweise)
 (hd0,3)             /dev/hda4           F: (normalerweise)
 (hd1,3)             /dev/hdb4           ?
```

Vergleichen Sie `/usr/share/doc/grub/README.Debian.gz` und `/usr/share/doc/grub-doc/html/` für Details.

8.2 Aufzeichnung von Aktivitäten

8.2.1 Aufzeichnen von Shell-Eingaben

Um ein Unix-artiges System gekonnt zu administrieren, sind mitunter knifflige oder raffinierte Aufgaben zu erledigen. Machen Sie sich deshalb mit den grundlegenden Konfigurationsaufgaben vertraut, so dass Sie im Notfall wissen, wo Sie Hand anlegen müssen. Fensterbasierte Konfigurations-Werkzeuge sind nett und bequem, nutzen aber nichts, wenn man versucht eine defekte X-Window-Konfiguration wieder herzustellen.

Die Aufzeichnung von Tastatur-Eingaben ist, insbesondere als root, hin und wieder nützlich.

Emacs: M-x `shell` startet die Aufzeichnung. ("M" steht hier für die Meta-Taste, meist Alt oder auch Esc.) C-x C-w schreibt die aufgezeichnete Sequenz in eine Datei.

Shell: Verwenden Sie das `screen`-Kommando mit "^A H", wie in 'Konsolenumschaltung mit `screen`' auf Seite 116 beschrieben oder das `script`-Kommando.

```
$ script
Script wurde gestartet, die Datei ist typescript
 ... irgendwelche Eingaben ...
 Strg-D
$ col -bx <typescript >Sicherung
$ vi Sicherung
```

Falls das `script`-Programm fehlt, kann diese Funktion mit der Shell simuliert werden:

```
$ bash -i 2>&1 | tee typescript
```

8.2.2 X-Aktivitäten aufzeichnen

Um den grafischen Bildschirm aufzunehmen, auch X-Terminals und andere beliebige Fenster, kann `gimp` benutzt werden. Alternativen sind `xwd` (`xbase-clients`), `import` (`imagemagick`) und `scrot` (`scrot`).

8.3 Kopieren und Archivieren eines Unterverzeichnis

Diese Kopier- und Archivierungskommandos bieten die Grundlagen für eine Sicherung des Systems und der Daten. Ein Beispiel eines einfachen Backup-Skripts findet sich unter dem Namen `backup` in den Beispielskripten (`http://www.debian.org/doc/manuals/debian-reference/examples/`).

8.3.1 Grundlegende Kommandos zum Kopieren eines Unterverzeichnis

Möchte man seine Verzeichnisstruktur neu arrangieren, verschiebt man den Inhalt inklusive Datei-Links wie folgt:

```
Standardmethode:
# cp -a /source/directory /dest/directory # erfordert GNU cp
# (cd /source/directory && tar cf - . ) | \
        (cd /dest/directory && tar xvfp - )
Wenn ein harter Link beteiligt ist, benötigt man eine pedantische
Methode:
# cd /Pfad/zum/alten/Verzeichnis
# find . -depth -print0 | afio -p -xv -0a /neues/Verzeichnis
Vom entfernten Rechner:
# (cd /Quell/Verzeichnis && tar cf - . ) | \
        ssh user@host.dom (cd /Ziel/Verzeichnis && tar xvfp - )
Wenn es keine Links gibt:
# scp -pr user1@host1.dom:/Quell/Verzeichnis \
          user2@host2.dom:/Ziel/Verzeichnis
```

Hier gilt `scp` <==> `rcp` und `ssh` <==> `rsh`.

Die folgenden vergleichenden Informationen zum Kopieren eines kompletten Unterverzeichnis wurden von Manoj Srivastava `<srivasta@debian.org>` auf debian-user@lists.debian.org aufgeführt.

8.3.2 cp

Ursprünglich war das Programm `cp` für diesen Zweck nicht gut geeignet, da es keine symbolischen Links auflösen oder harte Links beibehalten konnte. Ein anderer Punkt waren spärliche (sparse) Dateien (Dateien mit Löchern).

Die GNU-Version von `cp` überwand zwar diese Probleme, doch auf nicht-GNU-Systemen kann es noch zu Schwierigkeiten kommen. Des Weiteren kann `cp` keine kleinen, portablen Archive erzeugen.

```
% cp -a . neuesverzeichnis
```

8.3.3 **tar**

Das Archivierungsprogramm `tar` kann im Gegensatz zu `cp` mit symbolischen Links umgehen. Obwohl `cpio` mit speziellen Dateien umgehen kann, konnten ältere `tar`-Versionen dies nicht.

Wenn `tar` auf mehrere harte Links einer Datei stößt, wird nur einmal die Datei in das Archiv kopiert; die Datei kann dann *nur* unter dem Namen der ursprünglichen Datei zurückgewonnen werden. `cpio` dagegen kopiert für jeden harten Link die komplette Datei in das Archiv, so dass sie unter jedem der Link-Namen wiedergeholt werden kann.

Das Kommando `tar` hat die Option für das Packen mit `.bz2`-Dateien zwischen Potato und Woody umbenannt. Empfohlen wird deshalb in Skripten die Option `--bzip2` statt der Kurzform `-I` (Potato) oder `-j` (Woody).

8.3.4 **pax**

Hinter dem Namen `pax` (IEEE-Standard 1003.2-1992, Seiten 380–388 (Abschnitt 4.48) und Seiten 936–940 (Abschnitt E.4.48)) verbirgt sich ein multifunktionales POSIX-Werkzeug zum Austausch von portablen Archiven. `pax` liest und schreibt Archive, listet deren Inhalt auf und kopiert Verzeichnishierarchien. `pax` arbeitet unabhängig von einem spezifischen Archivformat und unterstützt eine große Bandbreite an Formaten.

`pax` ist noch neu und es können noch Kinderkrankheiten auftreten.

```
# apt-get install pax
$ pax -rw -p e . newdir
 oder
$ find . -depth  | pax -rw -p e  newdir
```

8.3.5 **cpio**

Das Kommando verwaltet die gleichnamigen Archive oder auch solche, die mit `tar` erstellt wurden. Das Archiv kann statt einer Datei auch eine Pipe oder ein Magnetband sein.

```
$ find . -depth -print0 | cpio --null --sparse -pvd new-dir
```

8.3.6 **afio**

`afio` ersetzt `cpio`. Es liest dessen Archive schneller, unterstützt mehr Laufwerksoptionen, hat eine bessere Fehlerkorrektur und kann das Archiv auf mehrere Medien verteilen. Die Kompression mit `afio` ist zuverlässiger als die von `tar` und `cpio`. Am besten benutzt man `afio` in Backup-Skripts als "Archiv-Maschinerie".

```
$ find . -depth -print0 | afio -px -0a new-dir
```

Ich mache alle meine Backups auf Band mit `afio`.

8.4 Differenzielles Backup und Datensynchronisation

Differenzielle Backups und Datensynchronisation kann mit verschiedenen Methoden implementiert werden:

- `rcs`: Backups, auch mehrerer älterer Versionen, nur für Texte
- `rdiff-backup`: Backups, auch mehrerer älterer Versionen. Symbolische Links werden unterstützt.
- `pdumpfs`: Backups, auch mehrerer älterer Versionen in einem Dateisystem. Symbolische Links werden unterstützt.
- `rsync`: 1-Weg-Synchronisation
- `unison`: 2-Wege-Synchronisation
- `cvs`: Mehr-Wege-Synchronisation mit Server-Backups, auch mehrerer älterer Versionen, nur für Texte, ausgereift. Vergleichen Sie mit 'Concurrent Versions System (CVS) – System für simultane Versionen' auf Seite 201.
- `arch`: Mehr-Wege-Synchronisation mit Server-Backups, auch mehrerer älterer Versionen. Es gibt keine Dinge wie "Arbeitsverzeichnis".
- `subversion`: Mehr-Wege-Synchronisation mit Server-Backups, auch mehrerer älterer Versionen, Apache.

Kombinationen einer dieser Methoden mit der Archivierungsmethode, die in 'Kopieren und Archivieren eines Unterverzeichnis' auf Seite 100 beschrieben ist, und den automatisierten Jobs in 'Terminplanung für Prozesse (`cron`, `at`)' auf Seite 115 bilden ein nettes Backup-System.

Ich werde drei einfach zu nutzende Hilfsmittel angeben.

8.4.1 Differenzielles Backup mit rdiff

`rdiff-backup` bietet ein nettes und einfaches Backup mit differenziellen Versionen für beliebige Dateitypen, inklusive symbolischen Links. Sichern des Großteils von `~/` nach `/mnt /backup`:

```
$ rdiff-backup --include ~/tmp/keep --exclude ~/tmp  ~/ /mnt/backup
```

Wiederherstellen von drei Tage alten Daten aus diesem Archiv nach `~/old`:

```
$ rdiff-backup -r 3D /mnt/backup ~/old
```

Vergleichen Sie `rdiff-backup(1)`.

8.4.2 Tägliches Backup mit `pdumpfs`

`pdumpfs` ist ein einfaches System zum täglichen Backup, ähnlich zu Plan9s `dumpfs`, das tägliche Schnappschüsse bewahrt. Man kann auf die letzten Schnappschüsse zu beliebiger Zeit zugreifen, um eine Datei eines bestimmten Tages wiederherzustellen. Führen Sie ein Backup Ihres Homeverzeichnisses mit `pdumpfs` und `cron` aus!

`pdumpfs` erstellt die Schnappschüsse `YYYY/MM/DD` im Zielverzeichnis. Alle Quelldateien werden in das Schnappschussverzeichnis kopiert, wenn `pdumpfs` das erste Mal gestartet wird. Beim zweiten und folgenden Male kopiert `pdumpfs` nur aktualisierte oder neu erstellte Dateien und speichert nicht geänderte Dateien als harte Links auf die Dateien vom Schnappschuss des letzten Tages, um Speicherplatz zu sparen.

```
$ pdumpfs src-dir dest-dir [dest-basename]
```

Vergleichen Sie `pdumpfs(8)`.

8.4.3 Regelmäßige differenzielle Backups mit RCS

`Changetrack` zeichnet Änderungen von textbasierten Konfigurationsdateien regelmäßig in RCS-Depots auf. Lesen Sie `changetrack(1)`.

```
# apt-get install changetrack
# vi changetrack.conf
```

8.5 Wiederherstellen eines eingefrorenen Systems

8.5.1 Einen Prozess killen/beenden

`top` hilft außer Kontrolle geratene Prozesse zu identifizieren. 'P' sortiert die Spalten nach CPU-Last, 'M' nach Speicherverbrauch und 'k' kann einen Prozess "abschießen". Alternativ kann das BSD-artige `ps aux | less` oder System V-artige `ps -efH | less` verwendet werden. Die System V-artige Syntax zeigt die IDs der Elternprozesse (`PPID`), die zum Killen von Zombie-Kindprozessen genutzt werden können.

Verwenden Sie `kill` zum Killen eines Prozesses mittels der Prozess-ID-Nummer (oder um ihm Signale zu senden). `killall` bewerkstelligt dasselbe über den Namen des Programms. Oft verwendete Signale sind

```
 1: HUP,  Daemon neustarten
15: TERM, normales Beenden
 9: KILL, erzwungenes Beenden
```

8.5.2 Alt-SysRq

Wenn der Kernel mit Unterstützung der "magischen SysRq-Taste" kompiliert wurde, kann das System mit etwas Glück auch aus dem totalen Nirwana geholt werden. Drücken von Alt-SysRq (SysRq ist oft mit "Druck" beschriftet) bei einem i386, gefolgt von einer der Tasten `r 0 k e i s u b`, aktiviert die Kernel-Notbremse.

'r' (un'r'aw) stellt die Tastatur wieder her, nachdem beispielsweise X abgestürzt ist. '0' setzt das Level, mit dem Fehlermeldungen auf der Konsole ausgegeben werden, herunter. 'k' (sa'k', system attention key) killt alle Prozesse auf der aktuellen virtuellen Konsole. 'e' (t'e'rminate) beendet alle Prozesse des aktuellen Terminals außer `init`. 'i' (k'i'll) killt alle Prozesse außer `init`.

'S'ync, 'u'mount und re'b'oot sind wirklich nur für den allerletzten Notfall.

Debian-Standardkernel sind, zur Zeit während dieses Dokument geschrieben wird, nicht mit dieser Option kompiliert. Es muss zur Nutzung dieser Taste ein neuer Kernel kompiliert werden. Detaillierte Informationen finden sich in `/usr/share/doc/kernel-doc-`*`version`*`/Documentation/sysrq.txt.gz` oder `/usr/src/`*`kernel-version`*`/Documentation/sysrq.txt.gz`.

8.6 Elegante kleine Kommandos zum Merken

8.6.1 Dateibetrachter

`less` ist der Standard-Dateibetrachter, der Textdateien seitenweise ausgibt. Hilfe bietet 'h'. `less` kann mehr als dessen Urahn `more`. Es kann mit `eval $(lesspipe)` oder `eval $(lessfile)` in einer Shell-Startdatei überladen (erweitert) werden. Sehen Sie hierzu `/usr/share/doc/lessf/LESSOPEN`. Die Option `-R` erlaubt die Ausgabe von Sonderzeichen (raw characters) und schaltet die ANSI-Farbsequenzen ein. Vergleichen Sie `less(1)`.

`w3m` kann für einige Code-Systeme (EUC) eine Alternative sein.

8.6.2 Freier Speicher

`free` und `top` informieren über freien Speicher und dessen Verbrauch. Sorgen Sie sich nicht um die Größe des "used"-Eintrags in der zweite Datenzeile, sondern lesen Sie den Wert darunter ab (hier: 38792)

```
$ free -k # für 256MB Hauptspeicher
             total       used       free     shared    buffers cached
Mem:        257136     230456      26680      45736     116136 75528
-/+ buffers/cache:      38792     218344
Swap:       264996          0     264996
```

Die präzise Menge an Hauptspeicher, die zur Verfügung steht, kann mit `grep '^Memory' /var/log/dmesg` bestätigt werden, was in diesem Fall "Memory: 256984k/262144k available (1652k kernel code, 412k reserved, 2944k data, 152k init)" ergibt.

```
Total         = 262144k = 256M (1k=1024, 1M=1024k)
Free to dmesg = 256984k = Total - kernel - reserved - data - init
Free to shell = 257136k = Total - kernel - reserved - data
```

Etwa 5MB können nicht vom System verwendet werden, da der Kernel sie in Beschlag nimmt.

8.6.3 Setzen der Uhrzeit (BIOS)

```
# date MMDDhhmmCCYY
# hwclock --utc --systohc
# hwclock --show
```

Dies setzt die Hardware-Uhr auf MM/DD hh:mm, CCYY. Dabei stehen die Kürzel für DD=Tag, MM=Monat, hh=Stunde, mm=Minute, CCYY=Jahr. Die Zeiten werden in lokaler Zeit ausgegeben, die Hardware-Uhr verwendet jedoch UTC als Zeitzone.

Falls die Hardware-Zeit (BIOS) auf GMT gesetzt ist, muss `UTC=yes` in `/etc/default/rcS` eingetragen sein.

8.6.4 Setzen der Uhrzeit (NTP)

Referenz: Managing Accurate Date and Time HOWTO (`http://www.tldp.org/HOWTO/TimePrecision-HOWTO/index.html`).

Setzen der Zeit bei permanenter Internet-Verbindung

Die Uhrzeit kann automatisch mit Hilfe eines Zeit-Servers korrekt gesetzt werden:

```
# ntpdate server
```

Dies sollte in `/etc/cron.daily/` verwendet werden, wenn man eine permanente Internet-Verbindung hat.

Setzen der Uhrzeit bei sporadischer Internet-Verbindung

Das Paket `chrony` hilft hier weiter.

8.6.5 Konsole-Eigenschaften wie den Bildschirmschoner steuern

Zum Deaktivieren des Bildschirmschoners sind folgende Kommandos geeignet:

Auf der Konsole:

```
# setterm -powersave off
```

Start der kon2 (kanji) Konsole mit:

```
# kon -SaveTime 0
```

Während X-Windows läuft:

```
# xset s off
 oder
# xset -dpms
 oder
# xscreensaver-command -prefs
```

Lesen Sie die entsprechenden Handbuchseiten, um zu erfahren, wie man andere Eigenschaften der Konsole steuert. Vergleichen Sie auch `stty(1)` zum Ändern und zur Ausgabe von Terminal-Line-Einstellungen.

8.6.6 Durchsuchen von Datenbanken zur Systemverwaltung

Die glibc-Bibliothek ermöglicht das Durchsuchen von System-Datenbanken, nach beispielsweise passwd, group, hosts, services, protocols oder networks mittels `getent(1)`.

```
getent database [key ...]
```

8.6.7 Sound abstellen (beep)

Im Zweifelsfall kann man immer den Stecker des Lautsprechers herausziehen ;-) Für die Bash-Shell gilt:

```
echo "set bell-style none" >> ~/.inputrc
```

8.6.8 Fehlermeldungen auf der Konsole

Wenn übermäßig viele Fehlermeldungen die Konsole unbrauchbar werden lassen, sollte man zuerst in `/etc/init.d/klogd` nachsehen. Um das Warnlevel zu ändern, kann man `KLOGD="-c 3"` setzen. Neustart des Daemons mit `/etc/init.d/klogdrestart` aktiviert die neuen Werte. Alternativ kann `dmesg -n3` benutzt werden.

Das Warnlevel schlüsselt sich wie folgt auf:

- 0: KERN_EMERG, System ist unbenutzbar
- 1: KERN_ALERT, sofortiger Eingriff nötig
- 2: KERN_CRIT, kritischer Zustand
- 3: KERN_ERR, Fehler
- 4: KERN_WARNING, Warnung
- 5: KERN_NOTICE, normale aber bedeutende Nachricht
- 6: KERN_INFO, Information
- 7: KERN_DEBUG, reine Debug-Nachricht

Wenn eine bestimmte unkritische Fehlermeldung oft auftritt, kann diese auch durch einen trivialen Kernelpatch unterbunden werden. (Siehe dazu das Beispiel `shutup-abit-bp6` im Beispielverzeichnis (`http://www.debian.org/doc/manuals/debian-reference/examples/`).)

Auch ein Blick in `/etc/syslog.conf` kann hilfreich sein, um zu verstehen, welche Nachrichten auf der Konsole ausgegeben werden.

8.6.9 Setzen des korrekten Konsole-Typs

Die Konsole wird in Unix-artigen Systemen üblicherweise mit den Routinen aus der (n)curses-Bibliothek angesteuert. Dies erlaubt eine im Wesentlichen von der Terminal-Art unabhängige Ausgabe mit vernünftiger Update-Strategie. Siehe `ncurses(3X)` und `terminfo(5)`.

Debian bietet eine ganze Reihe von Voreinstellungen:

```
$ toe | less                  # alle Einträge
$ toe /etc/terminfo/ | less   # durch Benutzer einstellbare Einträge
```

Die Auswahl kann durch Export der Umgebungsvariablen `TERM` aktiviert werden.

Wenn der terminfo-Eintrag für `xterm` mit einem `xterm`, das remote aufgerufen wird und das nicht auf Debian läuft, nicht funktioniert, kann der Terminaltyp von "xterm" auch auf eine der primitiveren Varianten, wie "xterm-r6", umgestellt werden. Sehen Sie `/usr/share/doc/libncurses5/FAQ` für weitere Informationen. Der kleinste gemeinsame Nenner für terminfo ist "dumb".

8.6.10 Die Konsole wiederherstellen

Ist der Bildschirm auf Grund von `cat` *`Binärdatei`* unleserlich (und sogar eingetippte Kommandos sind nicht sichtbar), so hilft:

```
$ reset
```

8.6.11 Konvertieren einer Textdatei von DOS nach Unix

Eine DOS-Textdatei (Zeilenende von `^M^J`), kann mit einem einzigen Kommando in eine Unix-Textdatei (Zeilenende = `^J`) umgewandelt werden:

```
# apt-get install sysutils
$ dos2unix DOS-Datei
```

8.6.12 Konvertierung einer Textdatei mit `recode`

Folgendes wird Textdateien zwischen DOS-, Mac- und Unix-Zeilenende-Stilen umwandeln:

```
$ recode /cl../cr <dos.txt >mac.txt
$ recode /cr.. <mac.txt >unix.txt
$ recode ../cl <unix.txt >dos.txt
```

Das freie `recode` konvertiert Dateien zwischen verschiedenen Zeichensätzen und Surfaces mittels:

```
$ recode Zeichensatz1/Surface1..Zeichensatz2/Surface2 \
  <Eingabe.txt >Ausgabe.txt
```

Üblicherweise verwendete Zeichensätze sind (siehe auch 'Introduction to locales' auf Seite 156)[1]:

- `us` — ASCII (7 Bit)
- `l1` — ISO Latin-1 (ISO-8859-1, westliches Europa, 8 Bit)
- `KOI8-RU` — KOI8-RU für Russisch (Unix)
- `SJIS` — Shift-JIS für Japanisch (Microsoft)
- `ISO2022JP` — E-Mail-Kodierung für Japanisch (7 Bit)
- `u2` — UCS-2 (Universal Character Set, 2 Byte)
- `u8` — UTF-8 (Universal Transformation Format, 8 Bit)

Gebräuchliche Surfaces sind[2]:

- `/cr` — Wagenrücklauf als Zeilenende (Mac-Text)
- `/cl` — Wagenrücklauf und Zeilenumbruch als Zeilenende (DOS-Text)
- `/` — Zeilenumbruch als Zeilenende (Unix-Text)

[1] `recode` erlaubt gebräuchlichere Bezeichnungen als `iconv`.

[2] Zeilenenden:

- Wagenrücklauf (carriage return) bedeutet ASCII 13, ASCII 0xD, ^M und `\r`.
- Zeilenumbruch (line feed) bedeutet ASCII 10, ASCII 0xA, ^J und `\n`.

- `/d1` — Menschenlesbare Byte-weise Dezimalausgabe
- `/x1` — Menschenlesbare Byte-weise Hexadezimalausgabe
- `/64` — Base64-kodierter Text
- `/QP` — Quoted-Printable-kodierter Text

Weitere Informationen finden sich in der entsprechenden Beschreibung in `info recode`.

Es gibt auch spezialisiertere Konvertierungswerkzeuge:

- Zeichensatzkonvertierung:
  - `iconv` — Locale-Kodierungskonvertierungen
  - `konwert` — raffinierte Kodierungskonvertierungen
- Binärdateikonvertierung:
  - `uuencode` und `uudecode` — für Unix.
  - `mimencode` — für E-Mails.

8.6.13 Ersetzung regulärer Ausdrücke

Alle Vorkommen des regulären Ausdrucks *REGEX* können durch *TEXT* in allen Dateien *DATEIEN* ersetzt werden durch:

```
$ perl -i -p -e 's/REGEX/TEXT/g;' DATEIEN ...
```

`-i` zeigt an, dass die Orginaldateien bearbeitet werden, `-p` sorgt ausdrücklich für die Iteration über die Dateinamen. Wenn der reguläre Ausdruck kompliziert ist, kann man sich versichern, indem man die Originaldateien behält: Durch `-i.bak` anstelle von `-i` bleiben die Originale erhalten und bekommen die Endung `.bak`.

8.6.14 Bearbeiten einer Datei mittels eines Skripts

Das folgende Skript entfernt die Zeilen 5–10 und 16–20 ohne Umweg über eine temporäre Datei.

```
#!/bin/bash
ed $1 <<EOF
16,20d
5,10d
w
q
EOF
```

Die `ed`-Kommandos sind die selben wie beim `vi` im Kommandomodus. Das Bearbeiten der Datei von hinten erleichtert die Arbeit.

8.6.15 Extrahieren von Unterschieden und Einbringen von Updates für Quelldateien

Die folgenden Kommandos bestimmen die Unterschiede zwischen zwei Quelldateien und erzeugen diff-Dateien *Datei.patch1* und *Datei.patch2* im “unified”-Stil:

```
$ diff -u Datei.alt Datei.neu > Datei.patch1
$ diff -u alt/Datei neu/Datei > Datei.patch2
```

Die Diff-Datei (alternativ wird sie auch Patch-Datei genannt) wird verwendet, um Veränderungen zu beschreiben. Jeder der diese Datei erhält, kann diese Änderungen auf eine andere Datei wie folgt anwenden:

```
$ patch -p0 Datei < Datei.patch1
$ patch -p1 Datei < Datei.patch2
```

Wenn drei Versionen des Quellcodes vorliegen, können diese einfacher mit `diff3` vermengt werden:

```
$ diff3 -m Datei.meine Datei.alt Datei.deine > Datei
```

8.6.16 Eine große Datei in kleine zerlegen

```
$ split -b 650m Datei   # in 650MB große Stücke trennen
$ cat x* >großeDatei    # wieder zusammenfügen
```

8.6.17 Extrahieren von Daten aus Tabellen in Textdateien

Sei `DPL` der Name einer Textdatei, in welcher alle vorherigen Debian-Projektleiter mit Namen und Einführungsdatum, durch Freizeichen getrennt, aufgeführt sind.

```
Ian     Murdock   August  1993
Bruce   Perens    April   1996
Ian     Jackson   Januar  1998
Wichert Akkerman  Januar  1999
Ben     Collins   April   2001
Bdale   Garbee    April   2002
Martin  Michlmayr März    2003
```

Awk wird oft benutzt, um Daten aus dieser Art von Datei zu extrahieren.

```
$ awk '{ print $3 }' <DPL                    # Monat des Beginns
August
April
Januar
Januar
April
April
März
$ awk '($1=="Ian") { print }' <DPL          # Vorname Ian
Ian     Murdock   August  1993
Ian     Jackson   Januar  1998
$ awk '($2=="Perens") { print $3,$4 }' <DPL # wann fing Perens an
April 1996
```

Shells wie Bash sind ebenfalls in der Lage, Dateien dieser Art auszulesen:

```
$ while read Vorname Nachname Monat Jahr; do
    echo $Monat
  done <DPL
... selbe Ausgabe wie beim ersten Awk-Beispiel
```

Das eingebaute Kommando `read` verwendet die Zeichen in $IFS (interne Feld-Separatoren), um Zeilen in Wörter aufzuteilen.

Wenn IFS auf ":" gesetzt wird, kann `/etc/passwd` leicht mit der Shell ausgelesen werden:

```
$ altIFS="$IFS"   # alten Wert sichern
$ IFS=":"
$ while read Benutzer Passwort uid gid Zeilenrest; do
    if [ "$Benutzer" = "osamu" ]; then
      echo "$Benutzer's ID ist $uid"
    fi
  done < /etc/passwd
osamu's ID ist 1001
$ IFS="$altIFS"   # Wert zurücksetzen
```

(Wenn Awk dazu verwendet wird, so wird der Spaltentrenner mit `FS=":"` angegeben.)

IFS wird auch von der Shell benutzt, um die Ergebnisse von Parameterauswertungen, Kommando-Substitutionen und arithmetischen Auswertungen aufzuteilen. Dies geschieht nicht innerhalb von doppelten oder einfachen Anführungszeichen. Der Standardwert von IFS sind die Werte <Leerzeichen>, <Tabulator> und <neue Zeile>.

Man muss bei der Verwendung von IFS-Tricks vorsichtig sein. Eigenartige Dinge können geschehen, wenn die Shell Teile eines Skripts als **Eingabe** interpretiert.

```
$ IFS=":,"                          # ":" und "," seien Feldtrenner
$ echo IFS=$IFS,   IFS="$IFS"       # echo ist ein Bash Kommando
IFS=  , IFS=:,
$ date -R                           # nur eine Kommandoausgabe
Sat, 23 Aug 2003 08:30:15 +0200
$ echo $(date -R)                   # Untershell --> Eingabe der Haupt-Shell
Sat  23 Aug 2003 08 30 36 +0200
$ unset IFS                         # zurücksetzen von IFS auf den Standardwert
$ echo $(date -R)
Sat, 23 Aug 2003 08:30:50 +0200
```

8.6.18 Skript-Auszüge für Pipe-Kommandos

Hier folgen einige kleine lehrreiche Beispiele zur Verwendung von Pipes:

```
find /usr | egrep -v "/usr/var|/usr/tmp|/usr/local"
                     # Finde alle Dateien unterhalb /usr
                     # mit Ausnahme bestimmter Pfade
xargs -n 1 Kommando  # Starte Kommando für alle Eingaben von stdin
xargs -n 1 echo |    # Beliebige Leerzeichen werden zu Zeilenumbrüchen
xargs echo      |    # alle Zeilen zu einer zusammenfassen
grep -e Muster|     # Gebe Zeilen aus, die Muster
                     # enthalten
cut -d: -f3 -|       # gebe das dritte Feld aus, : sei Trenner
                     # (z.B. für die Passwortdatei passwd)
awk '{ print $3 }' | # extrahiere das dritte Feld, Freizeichen sei Trenner
awk -F'\t' '{ print $3 }' |
                     # gib drittes Feld aus mit Tab als Trenner
col -bx |            # Entferne Backspace und expandiere Tabs zu Leerzeichen
expand -|            # expandiere Tabs zu Leerzeichen
sort -u|             # Sortiere und entferne doppelte Einträge

tr '\n' ' '|         # mehrere Zeilen zu einer zusammenfügen
tr '\r' ''|          # CR entfernen
tr 'A-Z' 'a-z'|      # Großbuchstaben in Kleinbuchstaben umwandeln
sed 's/^/# /'|       # aus der Zeile einen Kommentar machen
sed 's/\.ext//g'|    # Entferne .ext
sed  -n -e 2p|       # zeige die zweite Zeile
head -n 2 -|         # zeige die ersten beiden Zeilen
tail -n 2 -|         # zeige die letzten beiden Zeilen
```

8.6.19 Skript-Auszüge zum Durchlaufen mehrerer Dateien

Die folgenden Möglichkeiten zum Durchlaufen aller auf `*.ext` passenden Dateien sind auch dann geeignet, wenn die Dateinamen Sonderzeichen wie Leerzeichen enthalten und bewirken alle dasselbe:

- Shell-Schleife (Dieses Beispiel verwendet mehrere Zeilen mit `PS2="> "`. Um dasselbe in einer Zeile zu erreichen, muss ein Semikolon für jeden Zeilenumbruch eingefügt werden.):

```
for x in *.ext; do
  if test -f "$x"; then
    Kommando "$x"
  fi
done
```

- `find` und `xargs` Kombination:

```
find . -type f -maxdepth 1 -name '*.ext' -print0 | \
 xargs -0 -n 1 Kommando
```

- `find` mit `-exec` Option mit einem Kommando:

```
find . -type f -maxdepth 1 -name '*.ext' \
 -exec Kommando '{}' \;
```

- `find` mit `-exec` Option mit einem kurzen Shell-Skript:

```
find . -type f -maxdepth 1 -name '*.ext' \
 -exec sh -c "Kommando '{}' && echo 'erfolgreich'" \;
```

8.6.20 Kurze Perl-Skripte

Obwohl jedes Awk-Skript automatisch in ein Perl-Skript mittels `a2p(1)` umgeschrieben werden kann, sollten einzeilige Awk-Skripte am besten manuell nach Perl konvertiert werden. Zum Beispiel ist

```
awk '($2=="1957") { print $3 }' |
```

äquivalent zu allen der folgenden Zeilen:

```
perl -ne '@f=split; if ($f[1] eq "1957") { print "$f[2]\n"}' |
perl -ne 'if ((@f=split)[1] eq "1957") { print "$f[2]\n"}' |
perl -ne '@f=split; print $f[2] if ( $f[1]==1957 )' |
perl -lane 'print $F[2] if $F[1] eq "1957"' |
```

Da alle Leerräume im `perl`-Argument in der obigen Zeile entfernt werden können und unter Ausnutzung der automatischen Umwandlung zwischen Zahlen und Zeichenketten in Perl kann auch Folgendes verwendet werden:

```
perl -lane 'print$F[2]if$F[1]eq+1957' |
```

Man vergleiche `perlrun(1)` für die Kommandozeilenoptionen. Für noch verrücktere Perl-Skripte wird auf `http://perlgolf.sourceforge.net` verwiesen.

8.6.21 Text oder ein Mailinglistenarchiv aus einer Webseite extrahieren

Das Folgende liest eine Webseite aus und schreibt die Ausgabe in eine Textdatei. Dies ist sehr nützlich, wenn man Konfigurationen aus dem Web kopieren will.

```
$ lynx -dump http://www.adresse.de/info.html >Textdatei
```

`links` und `w3m` können hier auch verwendet werden, die Ergebnisse sich aber eventuell leicht unterschiedlich.

Falls ein Archiv einer Mailingliste geladen wird, kann `munpack` benutzt werden, um den MIME-Inhalt zu extrahieren.

8.6.22 Formatierte Ausgabe von Webseiten

Um eine Webseite zu drucken, kann diese in das PostScript-Format umgewandelt werden:

```
$ apt-get install html2ps
$ html2ps URL | lpr
```

Man vergleiche '`lpr/lpd`' auf Seite 42. Alternativ können auch das `a2ps` oder `mpage` Paket zum Erzeugen von PostScript-Dateien verwendet werden.

8.6.23 Formatierte Ausgabe einer Handbuchseite

Das Folgende druckt eine Handbuchseite in eine PostScript-Datei/Drucker.

```
$ man -Tps some-manpage | lpr
$ man -Tps some-manpage | mpage -2 | lpr
```

8.6.24 Vermengen zweier PostScript- oder PDF-Dateien

Zwei PostScript- oder PDF-Dateien können wie folgt zu einer zusammengefasst werden:

```
$ gs -q -dNOPAUSE -dBATCH -sDEVICE=pswrite \
  -sOutputFile=bla.ps -f foo1.ps foo2.ps
$ gs -q -dNOPAUSE -dBATCH -sDEVICE=pdfwrite \
  -sOutputFile=bla.pdf -f foo1.pdf foo2.pdf
```

8.6.25 Ausführungsdauer messen

Um die verschiedenen Zeiten, die mit einem Prozessablauf verbunden sind, anzuzeigen, kann `time` verwendet werden.

```
# time some-command >/dev/null
real    0m0.035s       # "Echte" Zeit
user    0m0.000s       # Zeit in Benutzermodus
sys     0m0.020s       # Zeit in Kernelmodus
```

8.6.26 `nice`-Befehl

Der `nice`-Befehl dient dazu, die Priorität eines Prozesses zu setzen. Dessen Verwandter `renice` (Paket `bsdutils`) ändert die Priorität eines bereits laufenden Prozesses. Letzteres kann auch aus `top` heraus geschehen. Ein `nice`-Wert von 19 bedeutet niedrigste Priorität – der Prozess ist langsam –, absteigend wird die Priorität erhöht. Kleinere Werte als 0 kann nur der Administrator setzen, -20 ist die höchste Priorität.

```
# nice  -19 top                                        # sehr nett
# nice --20 cdrecord -v -eject speed=2 dev=0,0 disk.img # sehr schnell
```

Manchmal kann ein sehr langsamer Prozess dem System mehr schaden als nützen, also ist Vorsicht geboten.

8.6.27 Terminplanung für Prozesse (`cron`, `at`)

Mit `cron` und `at` können zu bestimmten Terminen Befehle ausgeführt werden. Siehe `at(1)`, `crontab(5)`,`crontab(8)`.

Die Tabelle für wiederkehrende Ausführung kann mit `crontab -e` bearbeitet werden. Beispiele einer crontab-Tabelle:

```
# Verwende /bin/sh zur Ausführung, egal was passwd sagt
SHELL=/bin/sh
# Standardausgabe wird an ‘paul’ geschickt.
MAILTO=paul
# Min Stunde TagDesMonats Monat TagDerWoche Befehl (Komma wird ’und’)
# starte um 00:05, jeden Tag
5  0  *  * *   $HOME/bin/daily.job >> $HOME/tmp/out 2>&1
# starte unm 14:15 am Monatsersten -- Ausgabe per E-Mail an Paul
15 14 1  * *   $HOME/bin/monthly
# starte um 22:00 an Werktagen (1-5), schicke E-Mail an Joe. % für Zeilenumbruch, letztes % für cc:
0 22 *   * 1-5 mail -s "Es ist 10 Uhr" joe%Joe,%%Gute Nacht?%.%%
23 */2 1 2 *   echo "Startet 23 Minuten nach 0 Uhr, 2 Uhr, 4 Uhr ..., am 1. Feb."
5  4 *   * sun echo "Startet um 04:05 jeden Sonntag"
# Startet um 03:40 an jedem ersten Montag des Monats
40 3 1-7 * *   [ "$(date +%a)" == "Mon" ] && command -args
```

Um einen Befehl zur einmaligen Ausführung zu terminieren dient `at`

```
$ echo 'command -args'| at 3:40 monday
```

8.6.28 Konsolenumschaltung mit screen

Mit `screen` kann man mehrere virtuelle Terminals auf einer einzigen Konsole starten, jedes mit einer eigenen interaktiven Shell. Selbst wenn man immer zwischen virtuellen Terminals umschalten kann oder mehrere `xterm`-Fenster offen hält, sollte man sich `screen` wegen seiner vielfältigen Möglichkeiten anschauen. Dazu gehören

- Kommando-Rückholung,
- Kopieren-Einfügen,
- protokollieren,
- Eingabe von Umlauten und
- die Möglichkeit eine komplette Sitzung an einem Terminal anzuhalten und später fortzusetzen.

Beispiel für einen Fernzugriff

Wenn man sich häufig mit einem VT100-Terminal-Programm über ein Netzwerk auf einem PC einwählt, wird man das Sitzungsmanagement mit `screen` sicherlich nützlich finden.

1 Angenommen man ist über eine Telefonverbindung eingeloggt, eine `screen`-Sitzung ist gestartet und man hat einen längeren Text mit einem Editor geschrieben. Aus irgendwelchen Gründen, muss man jetzt die Verbindung unterbrechen.
2 Durch Drücken von `^A d` wird die aktuelle Sitzung gesichert. (Noch schneller geht es mit `^A DD`, dann wird sogar gleich der Logout für einen vorgenommen.)
3 Wenn man sich später wieder einwählt kann man mit `screen -r` die Sitzung holen, und `screen` wird die Sitzung so restaurieren, wie sie verlassen wurde.

Typische screen-Befehle

Wenn der `screen`-Prozess gestartet ist, reicht dieser alle Tastatureingaben an das aktuelle Fenster, bis auf die Kommandosequenz, die auf `^A` voreingestellt ist. Alle `screen`-Befehle beginnen mit `^A` und werden dann von einem einzelnen Buchstaben gefolgt (ggf. folgen dann die Parameter). Beispiele für Befehle:

```
^A ?     Hilfe (Tastaturbelegung)
^A c     Neues Fenster und wechsele dorthin
^A n     Gehe zum nächsten Fenster
^A p     Gehe zum vorherigen Fenster
^A 0     Gehe zum Fenster Nummer 0
^A w     Zeige eine Liste der Fenster
^A a     Gebe das ^A an die Applikation weiter
^A h     Hardcopy des Fensters in Datei
^A H     Starte/Beende Protokoll des aktuellen Fensters in eine Datei
^A ^X    Sichere das Terminal mit einem Passwort
^A d     Aktuelle Sitzung vom Terminal abmelden
^A DD    Vom Terminal abmelden und Logout
```

Dies ist nur eine kleine Auswahl der Befehle, die das mächtige `screen` unterstützt. Siehe `screen(1)` für weitere Details.

Backspace und/oder Strg-H in einer screen Sitzung

Wenn die Backspace- oder Strg-H-Taste während einer `screen`-Sitzung nicht funktionieren, muss die Zeile

```
bindkey -k kb stuff "\177"
```

in der Datei `/etc/screenrc` auskommentiert ("#") werden.

screen-Äquivalent unter X

Siehe Paket `xmove` und dort `xmove(1)`.

8.6.29 Grundlagen – Prüfung des Netzwerks

Zu installierende Pakete, um die grundlegenden Funktionen des Netzwerks zu überprüfen: `netkit-ping`, `traceroute`, `dnsutils`, `ipchains` (für 2.2er Kernel), `iptables` (für 2.4er Kernel) und das `net-tools`-Paket. Nun sollte man folgende Sequenz ausführen:

```
$ ping yahoo.com               # teste Internetverbindung
$ traceroute yahoo.com         # tracen der IP-Pakete
$ ifconfig                     # testen der host-Konfiguration
$ route -n                     # testen der routing-Konfiguration
$ dig [@dns-server.com] host.dom [{a|mx|any}] |less
      # teste host.dom DNS-Einträge am dns-server.com
      # und suche dort nach {a|mx|any} Eintrag
$ ipchains -L -n |less         # teste packet-Filter (2.2 kernel)
$ iptables -L -n |less         # teste packet-Filter (2.4 kernel)
$ netstat -a                   # Finde alle offenen Ports
$ netstat -l --inet            # Finde alle auf Eingabe wartenden Ports
$ netstat -ln --tcp            # Ebenso (TCP, numerisch) Ports
```

8.6.30 E-Mail aus dem lokalen Spooler leiten (flush)

Um E-Mail aus dem lokalen Spooler weiterzuleiten:

```
# exim -q    # wartende E-Mail anstoßen
# exim -qf   # alle E-Mails weiterleiten
# exim -qff  # dasselbe, auch E-Mail mit Status 'frozen' wird weitergeleitet
```

`-qff` wäre wohl die bessere Wahl in dem Skript `/etc/ppp/ip-up.d/exim`. Für Sarge muss `exim` durch `exim4` ersetzt werden.

8.6.31 Eingefrorene E-Mails aus dem lokalen Spooler entfernen

Um nicht weiter vermittelbare E-Mails aus dem lokalen Spooler zu entfernen und eine Fehlermeldung zurückzugeben dient:

```
# exim -Mg `mailq | grep frozen | awk '{ print $3 }'`
```

Für Sarge muss `exim` durch `exim4` ersetzt werden.

8.6.32 Neuausliefern von `mbox`-Inhalten

Falls das Homeverzeichnis voll wurde und `procmail` fehlschlug, muss man E-Mails aus `/var/mail/`*`Benutzername`* erneut manuell in die sortierten Postfächer im Homeverzeichnis ausliefern. Nachdem Plattenplatz im Homeverzeichnis bereitgestellt wurde, ist Folgendes zu starten:

```
# /etc/init.d/exim stop
# formail -s procmail </var/mail/Benutzername
# /etc/init.d/exim start
```

Für Sarge muss `exim` durch `exim4` ersetzt werden.

8.6.33 Dateiinhalte zurücksetzen

Um eine Datei zurückzusetzen, z.B. eine Log-Datei, sollte nicht `rm` benutzt werden, da diese Dateien im Sekundenrhythmus geschrieben werden. Sicherer ist:

```
$ :>zu_leerende_Datei
```

8.6.34 Dummy-Dateien

Die folgenden Befehle erzeugen Dummy- oder leere Dateien beliebiger Größe:

```
$ dd if=/dev/zero    of=Dateiname bs=1k count=5 # 5KB große, aber leere Datei
$ dd if=/dev/urandom of=Dateiname bs=1M count=7 # 7MB mit Zufallsinhalt
$ touch Dateiname #erzeuge 0B Datei (wenn sie existiert, setze mtime neu)
```

Die folgenden Kommandos werden, wenn sie von der Shell der Debian-Boot-Diskette gestartet werden, beispielsweise den gesamten Inhalt der Festplatte `/dev/hda` für die meisten praktischen Gegebenheiten löschen.

```
# dd if=/dev/urandom of=/dev/hda; dd if=/dev/zero of=/dev/hda
```

8.6.35 `chroot`

Das Programm `chroot`, `chroot(8)`, ermöglicht es, verschiedene Instanzen der GNU/Linux-Umgebung in einem einfachen System simultan ohne Neustart laufen zu lassen.

Es ist auch möglich, ein ressourcenhungriges Programm wie `apt-get` oder `dselect` im Chroot eines schnellen Rechners laufen zu lassen, während das Dateisystem eines langsamen Systems per NFS les- und schreibbar gemountet wurde und der Chroot auf den Einhängepunkt verweist.

Betreiben einer anderen Debian-Distribution mit `chroot`

Ein auf `chroot` basierendes System kann leicht mit dem `debootstrap`-Kommando in Sarge erzeugt werden. Für Distributionen, die neuer als Sarge sind, kann stattdessen auch `cdebootstrap` mit entsprechender Option verwendet werden. Um zum Beispiel ein Sid-Chroot in */sid-root* zu erzeugen, wenn man einen schnellen Internet-Zugang hat:

```
main # cd /; mkdir /sid-root
main # debootstrap sid /sid-root http://ftp.debian.org/debian/
... beobachten, wie das gesamte System heruntergeladen wird
main # echo "proc-sid /sid-root/proc proc none 0 0" >> /etc/fstab
```

```
main # mount proc-sid /sid-root/proc -t proc
main # cp /etc/hosts /sid-root/etc/hosts
main # chroot /sid-root /bin/bash
chroot # cd /dev; /sbin/MAKEDEV generic; cd -
chroot # apt-setup # Einrichten von /etc/apt/sources.list
chroot # vi /etc/apt/sources.list # Quellen auf unstable setzen
chroot # dselect  # oder aptitude, installieren von mc und vim :-)
```

Zu diesem Punkt sollte ein voll funktionsfähiges Debian-System zur Verfügung stehen, in dem man ohne Angst, die Hauptinstallation zu beeinflussen, herumspielen kann.

Dieser `debootstrap`-Trick kann auch verwendet werden, um Debian auf einem System ohne ein Debian-Installationsmedium, aber dafür mit dem einer anderen GNU/Linux-Distribution, zu installieren. Siehe `http://www.debian.org/releases/stable/i386/apcs04`.

Einrichten eines Logins für **chroot**

Eingeben von `chroot /sid-root /bin/bash` ist einfach, behält aber alle Umgebungsvariablen bei, was man eventuell nicht möchte, und hat andere Probleme. Ein viel besserer Ansatz ist es, einen anderen Login-Prozess auf einem separaten virtuellen Terminal zu starten, mit dem man sich direkt am Chroot anmelden kann.

Da auf Standard-Debian-Systemen auf `tty1` bis `tty6` Linux-Konsolen und auf `tty7` das X Window System läuft, wird `tty8` als Beispiel für die Chroot-Konsole verwendet. Nach der Erzeugung eines Chroot-Systems wie in 'Betreiben einer anderen Debian-Distribution mit `chroot`' auf der vorherigen Seite beschrieben, ist Folgendes in einer root-Shell im Hauptsystem einzugeben:

```
main # echo "8:23:respawn:/usr/sbin/chroot /sid-root "\
       "/sbin/getty 38400 tty8" >> /etc/inittab
main # init q    # init neuladen
```

Einrichten von X für **chroot**

Die aktuellste Version von X und GNOME soll sicher in einem Chroot laufen? Das ist möglich! Das folgende Beispiel wird GDM auf dem virtuellen Terminal `vt9` starten.

Zuerst ist ein Chroot-System, wie unter 'Betreiben einer anderen Debian-Distribution mit `chroot`' auf der vorherigen Seite beschrieben, zu installieren. Aus dem Hauptsystem sind einige wichtige Konfigurationsdateien in das Chroot-System zu kopieren.

```
main # cp /etc/X11/XF86Config-4 /sid-root/etc/X11/XF86Config-4
main # chroot /sid-root # oder Verwendung der Chroot-Konsole
chroot # cd /dev; /sbin/MAKEDEV generic; cd -
chroot # apt-get install gdm gnome x-window-system
```

```
chroot # vi /etc/gdm/gdm.conf # in [servers] s/vt7/vt9/ ersetzen
chroot # /etc/init.d/gdm start
```

`/etc/gdm/gdm.conf` wurde editiert, um die erste virtuelle Konsole von `vt7` in `vt9` abzuändern.

Nun kann leicht zwischen der vollen X-Umgebung im Chroot und im Hauptsystem umgeschaltet werden, indem man die virtuellen Linux-Terminals wechselt. Zum Beispiel durch Drücken von Strg-Alt-F7 und Strg-Alt-F9. Haben Sie Spaß!

[FIXME] Einen Kommentar und Link zum Init-Skript des `gdm` im Chroot hinzufügen.

Andere Distributionen mit `chroot` laufen lassen

Eine Chroot-Umgebung für eine andere Linux-Distribution kann leicht erstellt werden. Sie installieren ein System in eine andere Partition unter Verwendung des Installationsprogramms der anderen Distribution. Falls dessen Root-Partition sich in */dev/hda9* befindet:

```
main # cd /; mkdir /andere-Dist
main # mount -t ext3 /dev/hda9 /andere-Dist
main # chroot /andere-Dist /bin/bash
```

Dann ist wie in 'Betreiben einer anderen Debian-Distribution mit `chroot`' auf Seite 119, 'Einrichten eines Logins für `chroot`' auf der vorherigen Seite und 'Einrichten von X für `chroot`' auf der vorherigen Seite zu verfahren.

Erstellen eines Pakets mit `chroot`

Es gibt ein ausgefeilteres `chroot`-Paket, `pbuilder`. Es erzeugt eine `chroot`-Umgebung und stellt ein Debian-Paket in dieser Sandbox zusammen. Es ist ideal zum Überprüfen von Bauabhängigkeiten und zum Sicherstellen, dass falsche Paketabhängigkeiten nicht in den erzeugten Paketen existieren.

8.6.36 Tests auf harte Links

Man kann wie folgt überprüfen, ob zwei Dateien die selbe Datei mit zwei harten Links sind:

```
$ ls -li Datei1 Datei2
```

8.6.37 Ein Festplatten-Image `mounten`

Wenn `Datei.img` ein Festplatten-Image enthält und die Quellfestplatte eine Konfiguration wie *xxxx* = (Byte/Sektor) * (Sektor/Zylinder) hat, dann kann dieses Image mit dem folgenden Befehl nach `/mnt` gemountet werden:

```
# mount -o loop,offset=xxxx file.img /mnt
```

Die meisten Festplatten haben 512 Byte/Sektor.

8.6.38 Samba

Grundlagen, um auf Windows-Dateien zuzugreifen:

```
# mount -t smbfs -o username=myname,uid=my_uid,gid=my_gid \
  //server/share /mnt/smb # freigegebene Windows-Laufwerke mounten
# smbmount //server/share /mnt/smb \
  -o "username=mein_name,uid=meine_uid,gid=meine_gid"
# smbclient -L 192.168.1.2 # Freigaben anzeigen
```

Die Samba-Netzwerk-Nachbarschaft kann angezeigt werden mit:

```
# smbclient -N -L eigene_IP_Adresse | less
# nmblookup -T "*"
```

8.6.39 Werkzeuge für fremde Dateisysteme

Viele fremde Dateisysteme werden vom Linux-Kernel unterstützt, so dass man durch einfaches Einbinden der Geräte auf diese zugreifen kann. Für bestimmte Dateisysteme, gibt es auch ein paar spezielle Werkzeuge zum Zugriff auf Dateisysteme ohne die Geräte einzubinden. Dies wird durch User-Space-Programme erreicht, so dass Kernel-Unterstützung für diese Dateisysteme nicht benötigt wird.

- `mtools`: für MSDOS-Dateisysteme (MS-DOS, Windows)
- `cpmtools`: für CP/M-Dateisysteme
- `hfsutils`: für HFS-Dateisysteme (nativer Macintosh)
- `hfsplus`: für HFS+-Dateisysteme (moderner Macintosh)

Um MS-DOS-FAT-Dateisysteme zu erzeugen und zu überprüfen, ist das Paket `dosfstools` nützlich.

8.7 Typische Fehler

Es werden einige Beispiele gefährlicher Aktionen aufgeführt. Die negativen Auswirkungen werden verstärkt, wenn das privilegierte Konto `root` verwendet wird.

8.7.1 **rm -rf .***

Die Verwendung von Jokerzeichen in Kommandozeilenargumenten wie in "`rm -rf .*`" kann gefährliche Auswirkungen haben, da "`.*`" beim Expandieren auch "`.`" und "`..`" enthält. Glücklicherweise prüft die aktuelle Version des "`rm`"-Kommandos in der Debian-Distribution die Dateinamen im Argument und verweigert die Entfernung von "`.`" und "`..`". Dies ist jedoch nicht immer der Fall. Man kann das Folgende versuchen, um zu testen, wie Joker in Dateinamen funktionieren.

- "`echo .`": listet das aktuelle Verzeichnis und alles darin auf.
- "`echo *`": listet jede Nicht-Punktdatei und Nicht-Punktverzeichnisse im aktuellen Verzeichnis auf.
- "`echo .[^.]*`": listet jede Punktdatei und alle Punktverzeichnisse im aktuellen Verzeichnis auf.
- "`echo .*`": listet das Elternverzeichnis und alles darin auf.

8.7.2 **rm /etc/passwd**

Der Verlust einiger wichtiger Dateien wie `/etc/passwd` durch eigenes Verschulden ist schlimm. Das Debian-System macht regelmäßig Sicherheitskopien dieser Dateien in `/var/backups/`. Wenn diese Dateien wiederhergestellt werden, muss man eventuell die Rechte manuell anpassen.

```
# cp /var/backups/passwd /etc/passwd
# chmod 644 /etc/passwd
```

Siehe auch 'Wiederherstellung von Paketauswahldaten' auf Seite 72.

Kapitel 9

Tunen eines Debian-Systems

Dieses Kapitel basiert auf älterer Version des englischen ursprünglichen Textes. Die englische Version bitte auch überprüfen.

Dieses Kapitel beschreibt nur die Grundlagen der Systemkonfiguration mittels textbasierter Schnittstellen. Sinnvolles Vorwissen liefert 'Debian-System-Installationshinweise' auf Seite 27.

Für diejenigen Leser, die besonders Wert auf die Sicherheit ihres Systems legen, sollte das Securing Debian Manual (`http://www.debian.org/doc/manuals/securing-debian-howto/`) Pflichtlektüre sein. Dieses Dokument findet sich auch im Paket `harden-doc`.

9.1 Hinweise zur System-Inititalisierung

Der Abschnitt 'Das `init`-Programm' auf Seite 22 beschreibt die Grundlagen der Debian init-Skriptdateien.

9.1.1 Anpassen der Init-Skripte

Die Debian-Distribution verwendet das sys-V System für die Init-Skripte. Da alle Dateien unterhalb von `/etc/init.d/*` als Konfigurationsdateien angesehen werden, können sie vom Systemadministrator beliebig verändert werden. Dennoch ist der empfohlene Ansatz, die Dateien in `/etc/default/*` anzupassen.

Die Datei `/etc/init.d/rcS` dient z.B. zum Einstellen von Vorgaben, die beim Systemstart aktiviert werden sollen. Hier können Programme wie `motd` oder `sulogin` konfiguriert bzw. gestartet werden.

9.1.2 Anpassen der System-Protokollierung

Die Art und Weise der System-Protokollierung kann in der Konfigurationsdatei des `syslogd` in `/etc/syslog.conf` bestimmt werden. Das `colorize` Paket kann die Ausgaben dieses

Daemons nach belieben färben, um die Übersicht zu erhöhen. Vergleichen Sie `syslogd(8)` und `syslog.conf(5)`.

9.1.3 Hardware-Optimierung

Einige Optimierungen der Hardware überlässt Debian dem Systemadministrator.

- `hdparm`
  - Optimierung des Festplattenzugriffs. Sehr effektiv.
  - Aber auch gefährlich. Die Handbuchseite `hdparm(8)` sollte auf jeden Fall zuvor gelesen werden.
  - `hdparm -tT /dev/hda` testet die Geschwindigkeit der Festplatte.
  - `hdparm -q -c3 -d1 -u1 -m16 /dev/hda` aktiviert einige der Optimierungen einer IDE-Festplatte (Dies kann gefährlich sein, nicht einfach übernehmen!)
- `setcd`
  - Optimierung des Zugriffs auf CD-Geräte.
  - `setcd -x 2` verringert die Geschwindigkeit auf 2x.
  - Vergleichen Sie `setcd(1)`.
- `setserial`
  - Dieses Paket enthält eine Sammlung von Programmen zur Einstellung der seriellen Schnittstelle.
- `scsitools`
  - Dieses Paket enthält eine Sammlung von Programmen zur Einstellung der SCSI-Schnittstelle.
- `memtest86`
  - Dieses Paket enthält eine Sammlung von Programmen zum Testen des Speichers.
- `hwtools`
  - Dieses Paket enthält eine Sammlung von Programmen, die den Low-Level-Zugriff auf Einstellungen der Hardware ermöglichen:
    - `irqtune`: Ändert die IRQ-Priorität von Geräten, um höhere Prioritäten und schnelleren Zugriff zu ermöglichen (z.B. für serielle Ports, Modems). Eine Beschleunigung um den Faktor 3 ist bei seriellen Ports und Modems möglich.
    - `scanport`: Durchsucht den I/O-Adressraum von 0x100 bis 0x3ff auf der Suche nach Geräten am ISA-Bus.
    - `inb`: Ein kleines Programm das den IO-Port ausliest und den Inhalt in hexadezimaler und binärer Darstellung ausgibt.
- `schedutils`
  - Linux Scheduler-Werkzeuge. (Der Scheduler verwaltet die Zeitscheibe, mit der Prozesse Rechenzeit zugewiesen bekommen.)
  - `taskset`, `irqset`, `lsrt` und `rt` sind in diesem Paket enthalten.

  – Zusammen mit den schon beschriebenen `nice` und `renice` Programmen (die aber nicht enthalten sind), können die Scheduler-Parameter bis ins Detail verwaltet werden.

Das Mounten eines Dateisystems mit der noatime Option ist auch sehr effektiv zur Beschleunigung von Lesezugriffen auf Dateien. Vergleichen Sie `fstab(5)` und `mount(8)`.

Bestimmte Hardware kann direkt durch den Linux Kernel und das proc Dateisystem getunt werden. Vergleichen Sie 'Tunen des Kernels mittels des proc-Dateisystems' auf Seite 91.

Es gibt viele hardwarespezifische Konfigurationshilfsmittel in Debian. Viele von diesen beziehen sich auf Laptops. Einige interessante in Debian verfügbare Pakete sind:
- `tpconfig` - ein Programm zur Konfiguration von Touchpads
- `apmd` - Programme für Advanced Power Management (APM)
- `acpi` - zeigt Informationen zu ACPI Geräten an
- `acpid` - Hilfsmittel zur Verwendung von ACPI
- `lphdisk` - bereitet eine Hibernation-Partition für Phoenix NoteBIOS vor
- `sleepd` - lässt ein Notebook während Inaktivität schlafen
- `noflushd` - erlaubt es inaktiven Festplatten herunterzufahren
- `big-cursor` - größerer Mauscursor für X
- `acme` - Aktiviert die "Multimedia Tasten" von Laptops
- `tpctl` - Konfigurationstools für IBM ThinkPad Hardware
- `mwavem` - Unterstützung von Mwave/ACP Modems
- `toshset` - erlaubt Zugriff auf die meisten Hardwareschnittstellen eines Toshiba Laptops
- `toshutils` - Hilfsmittel für Toshiba Laptops
- `sjog` - Programm zum Zugriff auf das "Jog Dial" (Drehrad) in Sony Vaio Laptops
- `spicctrl` - Programm für das Sony Vaio zum Einstellen der Helligkeit des LCD

ACPI ist ein moderneres Protokoll für das Power-Management als APM.

Einige dieser Pakete erfordern spezielle Kernel-Module. Diese sind in der Regel bereits in den neuesten Kernelquellen enthalten. Bei Problemen kann es notwendig sein, selbst den neuesten Patch in den Kernel einzuspielen.

9.2 Zugriffskontrolle

9.2.1 Zugriffskontrolle mit PAM und login

PAM (Pluggable Authentication Modules) erlaubt die Kontrolle von Logins.

```
/etc/pam.d/*            # PAM Konfiguration
/etc/pam.d/login        # PAM Konfiguration des Logins
/etc/security/*         # PAM Modulparameter
/etc/securetty          # Konfiguration des root-Login auf der Konsole
/etc/login.defs         # generelle Logins
```

Falls wirklich (auf eigene Gefahr) ein Login an der Konsole ohne Passwort möglich sein soll, kann die Datei /etc/pam.d/login wie folgt geändert werden:

```
#auth       required   pam_unix.so nullok
auth       required   pam_permit.so
```

Ähnlich funktioniert das auch mit xdm, gdm, ..., um sich ohne Passwort an der X-Konsole anzumelden.

Andererseits kann cracklib2 installiert werden und /etc/pam.d/passwd wie folgt angepasst werden, um einen guten Passwortschutz zu erhalten.

```
password required       pam_cracklib.so retry=3 minlen=6 difok=3
```

Ein einmaliges Nutzen eines Login Passworts zur Zugangsaktivierung kann auch hilfreich sein. Dafür muss das passwd Kommando mit der Option -e verwendet werden. Vergleichen Sie passwd(1).

Die maximale Anzahl an Prozessen kann mit ulimit -u 1000 in der Bash-Shell gesetzt werden oder auch in der Datei /etc/security/limits.conf mit PAM. Andere Parameter, wie z.B. core können so ebenfalls gesetzt werden. Die Anfangseinstellung von PATH kann in /etc /login.defs festgelegt werden, bevor die Shell Skripte startet.

Die Dokumentation von PAM befindet sich im Paket libpam-doc. Das Buch *Linux-PAM System Administrator's Guide* behandelt PAM, die verfügbaren Module, usw. Die Dokumentation enthält auch *The Linux-PAM Application Developers' Guide* und *The Linux-PAM Module Writers' Guide*.

9.2.2 "Warum GNU **su** nicht die **wheel** Gruppe unterstützt"

Dieser Satz entstammt dem Ende der alten info su Seite von Richard M. Stallman. Man sollte sich aber nicht verwirren lassen. Die aktuelle su Version in Debian benutzt PAM. Somit kann der Gebrauch von su leicht auf eine beliebige Gruppe eingeschränkt werden. Dazu dient das Modul pam_wheel.so, welches in /etc/pam.d/su konfiguriert wird. Im folgenden Beispiel wird die adm Gruppe äquivalent zu der BSD Gruppe wheel eingerichtet und erlaubt den Gruppenmitgliedern den Gebrauch des su Kommandos ohne Passwort.

```
# anti-RMS Konfiguration in /etc/pam.d/su
auth       required   pam_wheel.so group=adm

# wheel Mitglieder die su ohne Passwort verwenden können
auth       sufficient pam_wheel.so trust group=adm
```

9.2.3 Bedeutung verschiedener Gruppen

Interessante Gruppen sind:

- Die `root` Gruppe entspricht der Standard wheel Gruppe für `su`, wenn `pam_wheel.so` ohne das `group=` Argument verwendet wird.
- Die `adm` Gruppe darf Log-Dateien lesen.
- Die `cdrom` Gruppe darf auf das CD-ROM-Laufwerk zugreifen.
- Die `floppy` Gruppe darf auf das Diskettenlaufwerk zugreifen.
- Die `audio` Gruppe darf auf ein vorhandenes Audiogerät zugreifen.
- Die `src` Gruppe darf auf Quelltexte unterhalb von `/usr/src` zugreifen. Mit dieser Gruppe verwaltet man den Zugriff auf Quelltexte.
- Die `staff` Gruppe ist gedacht für Hilfsadministratoren. Sie sollen Dateien in `/usr/local` ablegen und Verzeichnisse in `/home` erzeugen dürfen.

Eine vollständige Liste bietet der "FAQ"-Abschnitt in Securing Debian Manual (`http://www.debian.org/doc/manuals/securing-debian-howto/`), dieses Dokument befindet sich im `harden-doc` Paket in Woody. Auch das neue `base-passwd` (>3.4.6) enthält eine maßgebliche Liste: `/usr/share/doc/base-passwd/users-and-groups.html`.

9.2.4 `sudo` – eine sicherere Arbeitsumgebung

Ich verwende `sudo` hauptsächlich als Selbstschutz. `sudo` ist die etwas sicherere Alternative zur Verwendung des Administrator-Accounts.

`sudo` wird wie üblich installiert und mittels der Datei `/etc/sudoers(http://www.debian.org/doc/manuals/debian-reference/examples/)` konfiguriert. Überprüfen Sie auch die Merkmale der `sudo` Gruppe in `/usr/share/doc/sudo/OPTIONS`.

Diese Beispielkonfiguration verwendet eine Gruppe "staff", welche es den Mitgliedern erlaubt, beliebige Befehle mit root-Rechten mittels `sudo` auszuführen. Des Weiteren wird dieses Recht einer Gruppe "src" mit Einschränkungen auf bestimmte Befehle gegeben.

Der Vorteil von `sudo` ist, dass nur ein normales Benutzerpasswort zum Einloggen benötigt wird. Die Aktivitäten eines `sudo` Benutzers werden aufgezeichnet. Dies ermöglicht es, einige Aufgaben an Hilfsadministratoren zu delegieren. Zum Beispiel:

```
$ sudo chown -R ich:meineGruppe .
```

Selbstverständlich kann jeder der das Administratorpasswortes kennt (wie es bei den meisten Heimanwendern der Fall ist) auch auf andere Weise root-Kommandos absetzen:

```
$ su -c "shutdown -h now"
Password:
```

(Mit `sudo` können feiner abgestufte Sicherheitsbarrieren eingebaut werden. Ich hätte dies tun können. Da aber nur ich Zugriff auf meinen Rechner habe, hielt ich das nicht für nötig.)

Ein anderes Programm mit ähnlicher Funktionalität ist im `super` Paket zu finden.

9.2.5 Zugriffskontrolle auf Daemonen

Der Internet-*Superserver* `inetd` wird beim Systemboot aus `/etc/rc2.d/S20inetd` (für das Runlevel 2) gestartet; dies wiederum ist ein symbolischer Link auf `/etc/init.d/inetd`. Durch die Verwaltung verschiedener Dienste/Daemonen in diesem einen Daemon wird die Systemlast verringert.

Wenn eine Anfrage via Netzwerk an den Rechner gestellt wird, sieht der Daemon in der Datei `/etc/protocols` und `/etc/services` nach, welches Protokoll und welcher Service nachgefragt werden. Bei einem normalen Internetservice schaut `inetd` dann in `/etc/inetd.conf` nach oder bei Sun-RPC-Anfragen in `/etc/rpc.conf`.

Die Sicherheit des Systems kann durch die Abschaltung nicht gebrauchter Services in `/etc/inetd.conf` immens gesteigert werden. Sun-RPC-Services werden gebraucht, wenn NFS oder andere RPB-basierte Protokolle verwendet werden.

Für einige Daemonen wird der gewünschte Server nicht direkt gestartet. Stattdessen wird der `tcpd` TCP/IP-Daemon-Wrapper mit dem gewünschten Server als Argument aus `/etc/inetd.conf` aufgerufen. In diesem Fall wird der Server von `tcpd` erst nach Log-Einträgen und Sicherheitsüberprüfungen gestartet (`/etc/hosts.deny` und `/etc/hosts.allow` werden ausgewertet).

Falls beim entfernten Zugriff in einem Debian-System Probleme auftreten, kann eine Ursache der Eintrag: "ALL: PARANOID" in `/etc/hosts.deny` sein, der dann auskommentiert werden muss.

Details dazu liefern: `inetd(8)`, `inetd.conf(5)`, `protocols(5)`, `services(5)`, `tcpd(8)`,`hosts_access(5)` und `hosts_options(5)`.

Weitere Informationen zu Sun-RPC liefern: `rpcinfo(8)`, `portmap(8)` und `/usr/share/doc/portmap/portmapper.txt.gz`.

9.2.6 LDAP – Leichtgewichtiges Verzeichniszugangsprotokoll

Literatur zu LDAP (Lightweight Directory Access Protocol):

- OpenLDAP (`http://www.openldap.org/`)
- OpenLDAP Admin Guide im `openldap-guide` Paket
- LDP: LDAP Linux HOWTO (`http://www.tldp.org/HOWTO/LDAP-HOWTO/index.html`)
- LDP: LDAP Implementation HOWTO (`http://www.tldp.org/HOWTO/LDAP-Implementation-HOWTO/index.html`)
- OpenLDAP, ausführliche Nutzungsberichte (`http://portal.aphroland.org/~aphro/ldap-docs/ldap.html`)
- Open LDAP mit Courier IMAP und Postfix (`http://alinux.washcoll.edu/docs/plc/postfix-courier-howto.html`)

9.3 CD-Brenner

CD-Brenner mit ATAPI/IDE-Schnittstelle sind in der letzten Zeit populär geworden, da sie gut geeignet für Archivierungs- und Backupzwecke kleiner Datenmengen (< 640 MB) sind. Das wichtigste Dokument zu diesem Thema ist das CD-Writing-HOWTO (`http://www.tldp.org/HOWTO/CD-Writing-HOWTO.html`).

9.3.1 Einführung

Der Abbruch des Datenstroms zum CD-Brenner während des Brennvorgangs wird unweigerlich dazu führen, dass der Rohling unbrauchbar wird. Deshalb sollte der Puffer-Speicher so groß wie möglich sein. Wenn Geld keine Rolle spielt, sollte man lieber einen SCSI-Brenner kaufen, hat man eine Wahl bzgl. des Bussystems, sollte man den Brenner unbedingt über den PCI-Bus (z.B. auf dem Motherboard), statt den ISA-Bus (wie bei der SB16) anschließen.

Ist ein CD-Brenner am IDE-Bus angeschlossen, wird der IDE-SCSI Treiber anstatt eines gewöhnlichen IDE CD-Treibers für Linux 2.2 und 2.4 Kernel verwendet. Auch der allgemeine SCSI-Treiber muss aktiviert sein. Es gibt zwei Möglichkeiten, wenn man einen Kernel, wie er mit modernen Distributionen (März 2001) ausgeliefert wird, hat:

9.3.2 Ansatz 1: Module + `lilo`

Die folgende Zeile muss in der Datei `/etc/lilo.conf` hinzugefügt werden, wenn ein Standard-Debian-Kernel verwendet wird. Falls dort schon Einträge vorhanden sind, können diese mit Leerzeichen getrennt werden.

```
append="hdx=ide-scsi ignore=hdx"
```

Der Anschlussname eines CD-Brenners, der durch den IDE-SCSI-Treiber angesprochen wird, ergibt sich dabei wie folgt: In `hdx` ist *x* zu ersetzen gemäß:

```
hda          für den Master am ersten IDE-Anschluss
hdb          für den Slave am ersten IDE-Anschluss
hdc          für den Master am zweiten IDE-Anschluss
hdd          für den Slave am zweiten IDE-Anschluss
hde ... hdh  für ein Laufwerk an einem externen oder ATA66/100
             IDE-Anschluss.
```

Die neue Lilo-Konfiguration wird mit root-Rechten aktiviert durch:

```
# lilo
# shutdown -h now
```

9.3.3 Ansatz 2: Neukompilieren des Kernels

Debian benutzt `make-kpkg` um einen Kernel zu erstellen. Die Option `--append_to_version` ist beim Erzeugen mehrerer Kernel sinnvoll. Siehe dazu 'Der Linux-Kernel unter Debian' auf Seite 85.

Nach Eingabe von `make menuconfig` geht man wie folgt vor:

- bzImage
- der IDE-CD-Treiber sollte entfernt werden
- ide-scsi und sg sollten angewählt werden (mindestens als Modul)

9.3.4 Weitere Schritte nach der Konfiguration

Die Kernel-Unterstützung für den CD-Brenner kann während des Boot-Prozesses aktiviert werden durch:

```
# echo ide-scsi >>/etc/modules
# echo sg       >>/etc/modules
# cd /dev; ln -sf scd0 cdrom
```

Manuelle Aktivierung geschieht durch:

```
# modprobe ide-scsi
# modprobe sg
```

Nach dem Neustart kann die Installation geprüft werden durch

```
$ dmesg|less
# apt-get install cdrecord
# cdrecord -scanbus
```

[Von Warren Dodge] Manchmal gibt es Konflikte zwischen den `ide-scsi`- und `ide-cd`-Treibern, wenn für beide Treiber Geräte installiert sind. Durch die folgende Zeile, die zu der Datei `/etc/modutils/aliases` hinzugefügt werden soll, kann das Problem beseitigt werden. Vor dem Neustart ist noch ein Lauf von `update-modules` nötig.

```
pre-install      ide-scsi      modprobe ide-cd
```

Durch diese Zeile wird der IDE-Treiber vor `ide-scsi` geladen. Der IDE-Treiber `ide-cd` übernimmt die Kontrolle über die ATAPI-CD-ROM Geräte – über alle die nicht ignoriert werden sollen. Die anderen bisher ignorierten Geräte übernimmt dann das `ide-scsi`-Modul.

9.3.5 Bootfähiges CD-Image

Um eine bootfähige CD-Image `cd-image.raw` mit den Dateien unterhalb vom `Zielverzeichnis` zu erzeugen, geht man wie folgt vor: Man legt eine bootfähige Diskette in das erste Diskettenlaufwerk ein und erzeugt davon ein Image:

```
# dd if=/dev/fd0 Zielverzeichnis/boot.img
# mkisofs -r -V volume_id -b boot.img -c bootcatalog -J -T \
          -o cd-image.raw Zielverzeichnis/
```

Die `-t` Option erzeugt die Übersetzungstabelle für lange Dateinamen, `-b` und `-c` steht für bootfähig.

Tatsächlich kann sogar eine bootfähige CD mit DOS erstellt werden. Wenn die Datei *`boot.img`* im obigen Beispiel eine DOS-Boot-Diskette beinhaltet, dann wird die CD gestartet, als ob diese Diskette im Diskettenlaufwerk stecken würde. Vielleicht möchte ja mal jemand das mit Freedos probieren.

Das CD-Image kann man mounten mit:

```
# mount -t iso9660 -o ro,loop cd-image.raw /cdrom
# cd /cdrom
# mc
# umount /cdrom
```

9.3.6 Brennen mit dem CD-Brenner (R, RW):

Ein erster Test mit einem Double-Speed-Laufwerk sieht so aus (dabei wird keine CD gebrannt, sondern nur simuliert):

```
# nice --10 cdrecord -dummy speed=2 dev=0,0 disk.img
```

Falls dies erfolgreich ist, kann die CD-R gebrannt werden:

```
# nice --10 cdrecord -v -eject speed=2 dev=0,0 disk.img
```

Eine wiederbeschreibbare CD-RW muss zudem erst gelöscht werden:

```
# nice --10 cdrecord -v -eject blank=fast speed=2 dev=0,0 disk.img
```

Manche CD-RW-Laufwerke funktionieren besser mit:

```
# nice --10 cdrecord -v blank=all speed=2 dev=0,0 disk.img
```

gefolgt von:

```
# nice --10 cdrecord -v -eject speed=2 dev=0,0 disk.img
```

Diese getrennten Schritte sollen SCSI-Timeouts während des Löschens vorbeugen, die den Brenn-Prozess stören oder sogar unterbrechen können. Der `nice`-Wert kann noch der Systemlast angepasst werden.

9.3.7 Eine Abbilddatei (Image) einer CD erzeugen

Manche CD-R oder auch kommerzielle CD's haben defekte Sektoren, die eine Kopie mit `dd` verhindern (die Windows 98 CD ist ein Beispiel). Das `cdrecord` Paket beinhaltet den `readcd` Befehl, der dann verwendet werden kann, um CD-Images zu erzeugen. Bei einer Daten-CD ist die Vorgehensweise wie folgt: Die CD wird gemountet, `mount` zeigt dann die Anzahl der Blöcke an (1 Block=1024 Bytes). Teilt man diese Zahl durch 2, so erhält man die Anzahl der CD-Blöcke, die 2048 Bytes groß sind. `readcd` liest also die CD z.B. mit folgenden Parametern:

```
# readcd dev=Ziel,lun,scsibusnr # wähle Funktion 11
```

Hier sind alle 3 Kommandozeilenparameter meistens 0. Gewöhnlich ist die Anzahl der Sektoren, die von `readcd` zurückgegeben wird, zu hoch! Dann sollte man die Sektorzahl, die man aus dem `mount` Befehl gewonnen hat, verwenden.

Es sollte bemerkt werden, dass die Verwendung von `dd` einige Probleme bei CD-ROMs hat. Der erste Lauf von `dd` kann eine Fehlermeldung erzeugen und ein kürzeres Image mit unvollständigem Ende zur Folge haben. Der zweite Lauf von `dd` kann zu einem größeren Image mit am Ende angehangenen falschen Daten führen. Dies ist bei einigen Systemen möglich, wenn die Größe des Images nicht angegeben wird. Nur der zweite `dd` Aufruf mit der korrekten Größe ohne vorherigen CD-Auswurf nach der Fehlermeldung scheint diese Probleme zu umgehen. Wenn die von `df` bestimmte Größe des Images z.B. `46301184` Blöcke ist, muss das folgende Kommando zweimal angewandt werden, um ein korrektes Image zu erhalten (dies ist eine rein empirische Information):

```
# dd if=/dev/cdrom of=cd.img bs=2048 count=$((46301184/2))
```

9.3.8 Debian-CD-Images

Um aktuelle Informationen zu Debian-CD's zu erhalten, sollte die Debian-CD Seite (`http://www.debian.org/CD/`) besucht werden.

Wenn man eine schnelle Internet-Verbindung hat, kann man auch eine Netz-Installation durchführen:

- hierzu die Disketten-Images (`http://www.debian.org/distrib/floppyinst`)
- und ein minimales bootbares CD-Image (`http://www.debian.org/CD/netinst/`).

Wenn man keine schnelle Internet-Verbindung hat, kann man auch einfach ein Komplettpaket von einem der bekannten CD Distributoren (`http://www.debian.org/CD/vendors/`) kaufen.

Bandbreite sollte nie verschwendet werden, deswegen sollte man die Wahl des Installationsmechanismus ruhig besonnen überlegen.

Ein bemerkenswertes Projekt ist KNOPPIX – Linux Live-Dateisystem auf CD (`http://www.knopper.net/knoppix/index-en.html`). Diese CD bootet ein vollständiges Debian-System völlig ohne Installation.

9.3.9 Sicherheitskopien auf CD-R

Um grundlegende Konfigurations- und Datendateien auf CD zu sichern, kann das "backup" Skript backup (`http://www.debian.org/doc/manuals/debian-reference/examples/`) verwendet werden. Siehe dazu auch 'Kopieren und Archivieren eines Unterverzeichnis' auf Seite 100 und 'Differenzielles Backup und Datensynchronisation' auf Seite 102.

9.3.10 Eine Musik-CD kopieren

Noch nicht ausreichend getestet ist Folgendes:

```
# apt-get install cdrecord cdparanoia
# cdparanoia -s -B
# cdrecord dev=0,0,0 speed=2 -v -dao -eject defpregap=1 -audio *.wav
```

oder

```
# apt-get install cdrdao # disk at once
# cdrdao read-cd --device /dev/cdrom --paranoia-mode 3 my_cd # CD auslesen
# cdrdao write --device /dev/cdrom --speed 8 my_cd    # auf neue CD schreiben
```

`cdrdao` erzeugt eine echte Kopie (ohne Sprünge, etc.).

9.4 Das X-Window Paket

Die graphische Arbeitsumgebung X wird von Xfree86 (`http://www.xfree86.org/`) bereitgestellt. Derzeit sind zwei Hauptvarianten bei Debian in Benutzung: Version 3.x und 4.x.

'X-Server' auf Seite 137 dies ist das Programm, das auf dem lokalen Monitor die X-Fenster und die Oberfläche zeichnet und die Tastatureingaben und Mausbewegungen auswertet.

'X-Client Programme' auf Seite 139 dies ist ein Programm, das (lokal auf einem eventuell fernen Rechner) per X-Window-Protokoll Applikationen laufen lässt.

Diese Definition kehrt den gewöhnlichen Gebrauch der Begriffe Client und Server um. Eine Erklärung hierfür liefert: `X(7)`, das LDP XWindow-User-HOWTO (`http://www.tldp.org/HOWTO/XWindow-User-HOWTO.html`) und die Remote X Apps mini-HOWTO (`http://www.tldp.org/HOWTO/mini/Remote-X-Apps.html`).

Es gibt verschiedene Wege, um X-basierte Programme, die lokal angezeigt werden sollen ("X-Server seitig"), auf einem fremden Rechner (einem "X-Client") zu starten.

- `xhost`
  - Host-List Mechanismus (sehr unsicher)
  - nicht-verschlüsseltes Protokoll (kann abgehört werden).
  - Diese Variante sollte möglichst nicht gewählt werden.
  - Siehe 'X-Fernverbindungen mit `xhost`' auf Seite 144 und `xhost(1x)`.
- `xauth`
  - Der MIT magic cookie-Mechanismus (unsicher, aber besser als `xhost`).
  - nicht-verschlüsseltes Protokoll (kann auch abgehört werden).
  - nur für lokal begrenzte Benutzung (weniger rechenintensiv als `ssh -X`).
  - Siehe 'Administratorrechte unter X' auf Seite 146 und `xauth(1x)`.
- `xdm`, `wdm`, `gdm`, `kdm`, ...
  - MIT magic cookie-Mechanismus (unsicher wie `xauth`).
  - Siehe `xdm(1x)` und `Xsecurity(7)` für die Erklärung der Grundlagen der X-Zugriffskontrolle
  - Siehe `wdm(1x)`, `gdm(8)`, und `kdm.options(5)` für weitere Informationen.
  - Siehe '`init` und Runlevel' auf Seite 82 um das graphische Login mit `xdm` abzuschalten, ohne das Paket zu deinstallieren.
- **`ssh -X`**
  - Port-Weiterleitung mittels secure shell (**sicher**).
  - verschlüsselt (Ressourcenverschwendung bei lokaler Anwendung)
  - nur für ferne Verbindung oder in unsicheren Netzen verwenden.
  - Siehe 'X-Fernverbindungen mit `ssh`' auf Seite 144.

Alle Netzzugänge zum X-System (außer `SSH`) erfordern TCP/IP Verbindungen. Siehe 'TCP/IP-Verbindungen mit X' auf Seite 144.

9.4.1 X-System Pakete

Es gibt einige Metapakete, um die Installation von X zu vereinfachen.

x-window-system-core Dieses Metapaket stellt die grundlegenden Komponenten des X-Window-Systems für eine Workstation bereit. Es beinhaltet Bibliotheken, einen X-Server (genauer das Paket `xserver-xfree86`), eine Reihe von Schriften und einige X-Client-Programme und Werkzeuge.

x-window-system Dieses Metapaket beinhaltet alle X-Window Komponenten, so wie sie vom XFree86-Projekt bereitgestellt wurden. Hinzu kommen einige Hilfsprogramme. Es hängt von `x-window-system-core`, `twm` und `xdm` ab. Das bedeutet, diese Abhängigkeiten werden automatisch aufgelöst und diese Pakete müssen nicht manuell zur Installation ausgewählt werden.

xserver-common-v3 In diesem Paket befinden sich die Dateien und Programme für das alte XFree86 Version 3.X (X3)

xserver-* Zusätzliche X3-Server Pakete zur Unterstützung von Hardware, die aus irgendwelchen Gründen nicht von XFree86 4.X unterstützt werden. Einige alte ATI mach64-Grafikkarten fallen darunter; einige andere Grafikkarten hängen sich mit der XFree Version in der Debian-Version woody auf, etc. (Alle Serverpakete findet man durch `apt-cache search xserver-|less`. Alle diese X3-Server hängen vom Paket `xserver-common-v3` ab.)

In den meisten Fällen allerdings sollte das Paket `x-window-system` installiert werden. (Möchte man keinen grafischen Login, sollte das Paket `xdm` entfernt werden. Eine genaue Beschreibung dieses Vorganges findet sich im Kapitel 'Abschalten von X beim Booten' auf Seite 97.)

9.4.2 Hardware-Erkennung für X

Um eine automatische Hardware-Erkennung durchzuführen, braucht man folgende Pakete, die vor der X-Installation vorhanden sein sollten:

- `discover` – Hardwareerkennungssystem.
- `mdetect` – Maus-Erkennung.
- `read-edid` – Informations-Werkzeug für VESA PnP Monitore.

9.4.3 X-Server

Siehe `XFree86(1x)` für weitere Informationen zum X-Server.

Der X-Server kann direkt von der Konsole gestartet werden:

```
$ startx -- :<display> vtXX
z.B.:
$ startx -- :1 vt8 -bpp 16
... startet auf vt8 am Rechner  localhost:1 mit 16 bpp-Farbmodus.
```

Argumente nach dem `--` werden an den X-Server weitergeleitet.

Wenn eine `~/.xserverrc` Konfigurationsdatei verwendet wird, um das Starten des X-Servers genauer im Griff zu haben, muss darauf geachtet werden, dass der Server auch explizit mit `exec` gestartet wird, sonst werden Start und Beenden des Servers sehr langwierig:

```
#!/bin/sh
exec /usr/bin/X11/X -dpi 100 -nolisten tcp
```

Konfiguration des X4-Servers

Um den X4-Server zu konfigurieren kann `dpkg` benutzt werden.

```
# dpkg-reconfigure --priority=low xserver-common
# dpkg-reconfigure --priority=low xserver-xfree86
```

Es wird eine `/etc/X11/XF86Config-4` Datei erzeugt, die mit dem Skript `dexconf` modifiziert werden kann.

Konfiguration des X3-Servers

Um den X3-Server zu konfigurieren kann auch `dpkg` benutzt werden.

```
# dpkg-reconfigure --priority=low xserver-common-v3
# dpkg-reconfigure --priority=low xserver-mach64
```

Die Konfigurationsdatei von X3 heißt `/etc/X11/XF86Config` und wird mit `xf86config-v3` bearbeitet.

Manuelle Konfiguration von X4

Um Benutzer spezifische Einstellungen vorzunehmen, darf nicht der Text verändert werden, der von debconf erstellt wurde, also ist der Bereich zwischen dem BEGIN und dem END tabu.

```
### BEGIN DEBCONF SECTION
[snip]
### END DEBCONF SECTION
```

Stattdessen können die Anpassungen vor dem BEGIN eingetragen werden, wie in dem folgenden Beispiel:

```
Section "Device"
  Identifier        "Custom Device"
  Driver            "ati"
  Option            "NoAccel"
EndSection

Section "Screen"
  Identifier  "Custom Screen"
  Device      "Custom Device"
  Monitor     "Generic Monitor"
  DefaultDepth 24
  Subsection "Display"
    Depth       8
    Modes       "1280x960" "1152x864" "1024x768" "800x600" "640x480"
  EndSubsection
  Subsection "Display"
    Depth       16
    Modes       "1280x960" "1152x864" "1024x768" "800x600" "640x480"
  EndSubsection
  Subsection "Display"
    Depth       24
    Modes       "1280x960" "1152x864" "1024x768" "800x600" "640x480"
  EndSubsection
EndSection

Section "ServerLayout"
    Identifier       "Custom"
    Screen           "Custom Screen"
    InputDevice      "Generic Keyboard" "CoreKeyboard"
    InputDevice      "Configured Mouse" "CorePointer"
EndSection
```

9.4.4 X-Client Programme

Die meisten X-Programme können einfach von der Kommandozeile aus gestartet werden:

```
client $ xterm -geometry 80x24+30+200 -fn 6x10 -display hostname:0 &
```

Die hier angeführten Parameter bedeuten:

- `-geometry` *BREITE*x*HÖHE*+*X-OFFSET*+*Y-OFFSET*: anfängliche Größe und Lage des neuen Fensters.
- `-fn` *FONTNAME*: die Schrift, die im Fenster verwendet wird. *FONTNAME* kann sein:

  - a14: Normale Größe
  - a24: Große Größe
  - ... (Weitere Fonts können mit `xlsfont` ermittelt werden.)

- `-display` *`displayname`*: der Name des X-Servers, der verwendet wird. *`displayname`* kann sein:
  - *`hostname`*`:`*`D.S`* bedeutet Anzeige *`S`* auf dem Bildschirm *`D`* auf dem Rechner *`hostname`*; der X-Server für dieses Display liegt auf dem TCP-Port 6000+D.
  - *`host`*`/unix:`*`D.S`* bedeutet Anzeige *`S`* auf dem Bildschirm *D* auf dem Rechner *`host`*; der X-Server für diese Anzeige lauscht auf dem Unix-Domain-Socket `/tmp/.X11-unix/XD` (so ist es nur auf dem Rechner *`host`* erreichbar).
  - *`:D.S`* bedeutet das gleiche wie *`host`*`/unix:`*`D.S`*, wobei *`host`* der Name des lokalen Rechners ist.

Der Standard *`displayname`* für die X-Client-Programme kann mit der DISPLAY Umgebungsvariablen gesetzt werden. Bevor man nun ein X-Programm startet kann die Umgebungsvariable z.B. wie folgt gesetzt werden:

```
$ export DISPLAY=:0
        # Die Standardeinstellung
$ export DISPLAY=hostname.fulldomain.name:0.2
$ export DISPLAY=localhost:0
```

Der Start kann in der Datei `~/.xinitrc` konfiguriert werden. Zum Beispiel:

```
xrdb -load $HOME/.Xresources
xsetroot -solid gray &
xclock -g 50x50-0+0 -bw 0 &
xload -g 50x50-50+0 -bw 0 &
xterm -g 80x24+0+0 &
xterm -g 80x24+0-0 &
twm
```

Wie in 'Angepasste X-Sitzung' auf der nächsten Seite beschrieben, haben diese Einstellungen Vorrang vor denen von Xsession, welches bei `startx` verwendet wird. Diese Methode sollte nicht verwendet werden.

9.4.5 X-Session

Eine X-Session kann auf folgende Weisen gestartet werden:

- `startx`: ein Skript, das `initx` und so den X-Server und Client auf der Konsole startet. Wenn die Datei `~/.xinitrc` nicht existiert, wird `/etc/X11/Xsession` via `/etc/X11/xinit/xinitrc` ausgeführt.

- xdm, gdm, kdm oder wdm: Dies sind Daemonprogramme, die den Login von einer bereits gestarteten graphischen Oberfläche ermöglichen. /etc/X11/Xsession: direktes Starten.

Die Konsole erhält man mittels 'Abschalten von X beim Booten' auf Seite 97.

Angepasste X-Sitzung

Im Endeffekt ist der Start durch /etc/X11/Xsession nur eine Kombination von /etc/X11/Xsession.d/50xfree86-common_determine-startup und /etc/X11/Xsession.d/99xfree86-common_start.

Die Ausführung von /etc/X11/Xsession wird beeinflusst durch /etc/X11/Xsession.options und ist im Wesentlichen bestimmt für die Ausführung des ersten Programms, das in folgenden Skripten durch 'exec' ausgeführt wird:

1. ~/.xsession oder ~/.Xsession, wenn vorhanden.
2. /usr/bin/x-session-manager, wenn vorhanden.
3. /usr/bin/x-window-manager, wenn vorhanden.
4. /usr/bin/x-terminal-emulator, wenn vorhanden.

Was nun genau gestartet wird, hängt von den Einstellungen des Debian-Alternativen-Systems ab, wie es in 'Alternative Befehle' auf Seite 82 erklärt wird. Zum Beispiel:

```
# update-alternatives --config x-session-manager
... oder
# update-alternatives --config x-window-manager
```

Um einen anderen X-Window-Manager einzustellen als die Voreinstellung und dennoch KDE oder Gnome als Sitzungsmanager beizubehalten, muss die Datei /etc/X11/Xsession.d/50xfree86-common_determine-startup durch die im zweiten Fehlerreport (siehe http://bugs.debian.org/168347) ersetzt werden. (Ich hoffe, dass dies bald mit aufgeführt wird.) Dann kann die Datei /etc/X11/Xsession.options wie folgt geändert werden, um den X-Sitzungsmanager zu umgehen:

```
# /etc/X11/Xsession.options
#
# configuration options for /etc/X11/Xsession
# See Xsession.options(5) for an explanation of the available options.
# Default enabled
allow-failsafe
allow-user-resources
allow-user-xsession
use-ssh-agent
# Default disabled (enable them by uncommenting)
do-not-use-x-session-manager
#do-not-use-x-window-manager
```

Ohne die gezeigten Veränderungen sind die Pakete `gnome-session` und `kdebase` dafür zuständig, einen X-Sitzungsmanager bereitzustellen. Das Entfernen dieser beiden Pakete ermöglicht die Einstellung eines beliebigen X-Window-Managers.

Wenn die Datei `/etc/X11/Xsession.options` eine Zeile wie `allow-user-xsession` (ohne führende Zeichen) enthält, kann der Benutzer die Datei `~/.xsession` oder `~/.Xsession` für eine beliebige benutzerseitige Erweiterung der Datei `/etc/X11/Xsession` verwendet werden.

Der letzte Befehl in `~/.xsession` sollte die Form `exec` *`some-window/session-manager`* haben, um den Window-Manager zu starten.

Ein gutes Beispiel für eine `~/.xsession` Datei findet sich hier: `/usr/share/doc/xfree86-common/examples/xsession.gz`.

Ich benutze dies, um den Window-Manager, Bildschirmzugang und die Sprache für jeden Benutzer einzeln zu setzen. Siehe 'Start eines gewählten X-Sitzungs-/Window-Managers' auf dieser Seite, 'Administratorrechte unter X' auf Seite 146 und 'Beispiel für ein zweisprachiges System (japanisches EUC und ISO-8859-1)' auf Seite 159.

Benutzerspezifische X-Einstellungen können in `~/.Xresources` gespeichert werden, wohingegen sich systemweite Einstellungen in `/etc/X11/Xresources/*` befinden. Siehe `xrdb(1x)`.

Benutzerspezifische Tastaturbelegungen und Maustastenbelegungen können in `~/.xmodmaprc` definiert werden. Siehe `xmodmap(1x)`.

Start eines gewählten X-Sitzungs-/Window-Managers

Eine benutzergewählte Sitzungs-/Window-Manager-Konfiguration kann also gemäß 'Angepasste X-Sitzung' auf der vorherigen Seite in `~/.xsession` durch das Ändern der letzten Zeile eingestellt werden. (Ich mag blackbox wegen dessen Geschwindigkeit und Einfachheit):

- Voreinstellung – X-Sitzungsmanager.
  - Vergleichen Sie 'Alternative Befehle' auf Seite 82.
  - `exec /usr/bin/x-session-manager`
- Voreinstellung – X-Window-Manager.
  - Vergleichen Sie 'Alternative Befehle' auf Seite 82.
  - `exec /usr/bin/x-window-manager`
- GNOME-Sitzungsmanager. (mächtiges Paket)
  - Installieren Sie das Paket: `gnome-session`
  - `exec /usr/bin/gnome-session`
- KDE-Sitzungsmanager. (ebenso mächtiges Paket)
  - Installieren Sie das Paket: `kdebase` (oder `kdebase3` für KDE3)
  - `exec /usr/bin/kde2`

- Blackbox-Window-Manager. (leichtgewichtiges Paket).
  - Installieren Sie das Paket: `blackbox`
  - `exec /usr/bin/blackbox`
- IceWM-Window-Manager. (ebenso leichtgewichtig)
  - Installiere das Paket: `icewm`
  - `exec /usr/bin/X11/icewm`
- FVWM2-Window-Manager.
  - Installieren Sie das Paket: `fvwm`
  - `exec /usr/bin/fvwm2`
- Windowmaker-Window-Manager.
  - Installieren Sie das Paket: `wmaker`
  - `exec /usr/bin/wmaker`
- Enlightenment-Window-Manager (überladen).
  - Installieren Sie das Paket: `enlightenment`
  - `exec /usr/bin/enlightenment`

Siehe X-Window-Manager (`http://www.xwinman.org`).

KDE/GNOME-Umgebung konfigurieren

Um eine komplette KDE/Gnome-Umgebung einzurichten empfiehlt sich die Installation der folgenden Metapakete:

- KDE: installiert das `kde` Paket
- GNOME: installiert das `gnome` Paket

Es sollte ein Installationswerkzeug gewählt werden, das die Installationsempfehlungen (`recommends`) des Paketes auflösen kann; dies sind u.a. `dselect` und `aptitude`. Sie sind menügesteuert und können auch durch die automatische Abhängigkeitserfassung die Installation im Vergleich zu `apt-get` beschleunigen und erleichtern.

Wer nur ein Konsolen-Login benötigt, sollte die Login-Manager `kdm`, `gdm`, `wdm` oder auch `xdm` nicht installieren (d.h. wieder markieren zur *nicht*-Installation), die bei einer Installation von X schnell automatisch mit ausgewählt werden. Erklärung hierzu in 'Abschalten von X beim Booten' auf Seite 97.

Wenn Gnome anstelle von KDE als Sitzungsmanager gewünscht wird, muss dies der `x-session-manager`, wie in 'Alternative Befehle' auf Seite 82 beschrieben, geändert werden.

9.4.6 TCP/IP-Verbindungen mit X

Da X-TCP/IP-Verbindungen nicht verschlüsselt werden, besteht bei Fernverbindungen die Gefahr des Abgehörtwerdens. Aus diesem Grund ist in Debian die Standardeinstellung so gewählt, dass TCP/IP-Socket deaktiviert ist. Eine TCP/IP-Verbindung auch zur Übertragung von graphischen Elementen kann aber sicher mit `ssh` erzeugt werden. Siehe dazu 'X-Fernverbindungen mit `ssh`' auf dieser Seite.

Die Methode, die hier im Folgenden beschrieben wird, ist also nur in Umgebungen zu empfehlen, die als sicher charakterisiert werden können, wie z.B. ein hinter einer gut gewarteten Firewall agierendes Intranet oder ein kleines Heimnetzwerk. Mit dem folgenden Befehl kann die momentane X-Server-Einstellung bzgl. TCP/IP herausgefunden werden:

```
# find /etc/X11 -type f -print0 | xargs -0 grep nolisten
/etc/X11/xinit/xserverrc:exec /usr/bin/X11/X -dpi 100 -nolisten tcp
```

Entfernen von `-nolisten` stellt das Lauschen auf dem X-Server wieder ein.

9.4.7 X-Fernverbindungen mit xhost

`xhost` erlaubt Zugriff auf den X-Server basierend auf dem Namen des Rechners. Dies ist sehr unsicher. Das folgende Beispiel unterdrückt den Test der Rechnernamen und erlaubt so Verbindungen von jedem Rechner aus, falls TCP/IP-Socket ebenfalls aktiviert sind. (Siehe auch 'TCP/IP-Verbindungen mit X' auf dieser Seite):

```
$ xhost +
```

Das Testen der Rechnernamen kann wieder aktiviert werden mit:

```
$ xhost -
```

`xhost` kann keine Unterscheidung auf Benutzerebene vornehmen. Des Weiteren können Rechnernamen natürlich gefälscht werden.

Diese Methode sollte nicht verwendet werden, wenn man sich in einer ungesicherten Netzumgebung befindet – dies betrifft auch Wählverbindungen mittels PPP. Siehe `xhost(1x)`.

9.4.8 X-Fernverbindungen mit ssh

Die Verwendung von `ssh` ermöglicht die sichere Übertragung von Daten von einem lokalen X-Server zu einem Applikations-Server.

- Setze auf dem entfernten Rechner in der Datei `/etc/ssh/sshd_config` die Einträge `X11Forwarding` und `AllowTcpForwarding` auf `yes`.

- Starte den X-Server auf dem lokalen Rechner.
- Öffne ein `xterm` auf dem lokalen Rechner.
- Starte `ssh` um eine Verbindung zu dem entfernten Rechner aufzubauen.

```
localname @ localhost $ ssh -q -X -l loginname remotehost.domain
Password:
...
```

- Starte eine X-basierte Applikation auf dem entfernten Rechner.

```
loginname @ remotehost $ gimp &
```

Diese Methode erlaubt es, die graphische Ausgabe eines X-Clients anzuzeigen, als liefe er über den lokalen Unix-Domain-Socket.

9.4.9 `xterm`

Alles was mit `xterm` zu tun hat, findet man hier: `http://dickey.his.com/xterm/xterm.faq.html`.

9.4.10 X-Ressourcen Datenbank

Viele ältere X-Programme, wie z.B. xterm, verwenden die X-Ressourcen, um ihr Erscheinungsbild zu konfigurieren. Die Datei `~/.Xresources` wird verwendet um derartige Daten benutzerabhängig zu speichern. Die Datei wird automatisch während des Logins an die systemweiten Einstellungen angehängt.

Es folgen einige nützliche Einstellungen, die man in `~/.Xresources` vornehmen kann.

```
! Die Schrift in den Terminals vergrößern:
XTerm*font: 9x15

! Scrollbalken anzeigen
XTerm*scrollBar: true

! Puffergröße einstellen
XTerm*saveLines: 1000
```

Diese Einstellungen können auch sofort aktiviert werden durch:

```
xrdb -merge ~/.Xresources
```

9.4.11 Administratorrechte unter X

Wenn ein graphisches Programm mit root-Rechten gestartet werden muss, müssen einige Vorkehrungen getroffen werden. Keinesfalls sollte man als Superuser einen X-Server starten, wegen der beträchtlichen Sicherheitsrisiken.

Man startet den X-Server als gewöhnlicher Benutzer und öffnet ein `xterm` Fenster. Dann folgen:

```
$ XAUTHORITY=$HOME/.Xauthority
$ export XAUTHORITY
$ su root
Password:*****
# printtool &
```

Wenn man diesen Trick verwendet und sich mit `su` als ein Nicht-root-Benutzer anmeldet, muss darauf geachtet werden, dass die Datei `~/.Xauthority` für diesen speziellen Benutzer Gruppen-lesbar ist.

Um diesen Vorgang zu automatisieren, kann der betreffende Benutzer eine Datei `~/.xsession` anlegen, die folgende Zeilen enthält.

```
# So funktioniert X auch wenn su benutzt wurde.
if [ -z "$XAUTHORITY" ]; then
        XAUTHORITY=$HOME/.Xauthority
        export XAUTHORITY
fi
unset XSTARTUP
# Wenn ein besonderer Window/Sitzungsmanager gewünscht wird
# kann der hier eingestellt werden.
#XSTARTUP=/usr/bin/blackbox
# This start x-window/session-manager program
if [ -z "$XSTARTUP" ]; then
  if [ -x /usr/bin/x-session-manager ]; then
    XSTARTUP=x-session-manager
  elif [ -x /usr/bin/x-window-manager ]; then
    XSTARTUP=x-window-manager
  elif [ -x /usr/bin/x-terminal-emulator ]; then
    XSTARTUP=x-terminal-emulator
  fi
fi
# Ausführung des Window/Sitzungsmanagers.
exec $XSTARTUP
```

Nun kann `su` (nicht `su -`) in einem `xterm` Fenster vom Benutzer gestartet werden. Auch graphische Programme, die von der Shell mit Superuser-Rechten gestartet wurden, können nun angezeigt werden. Dieser Trick funktioniert nur solange, wie die übliche `/etc`

`/X11/Xsession` ausgeführt wird. Wenn der Benutzer eine eigene Datei `~/.xinit` oder `~/.xsession` Datei anlegt, muss die oben erwähnte Umgebungsvariable `XAUTHORITY` an diese Skripte angepasst werden.

Alternativ, kann das Paket `sudo` verwendet werden, um diese Sequenz abzuarbeiten.

```
$ sudo xterm
... oder
$ sudo -H -s
```

Hier muss die `/root/.bashrc` Folgendes enthalten:

```
if [ $SUDO_USER ]; then
    sudo -H -u $SUDO_USER xauth extract - $DISPLAY | xauth merge -
fi
```

Dies funktioniert auch, wenn das Home-Verzeichnis des Benutzers mit NFS gemountet wird, da root die Datei `.Xauthority` nicht lesen muss.

There are also several specialized packages for this purpose: `kdesu`, `gksu`, `gksudo`, `gnome-sudo`, and `xsu`. Some other methods can be used to achieve similar results: creating a symlink from `/root/.Xauthority` to the user's corresponding one; use of the script sux (`http://fgouget.free.fr/sux/sux-readme.shtml`); or putting "`xauth merge ~`*`USER_RUNNING_X`*`/.Xauthority`" in the root initialization script.

See more on the debian-devel mailing list (`http://lists.debian.org/debian-devel/2002/debian-devel-200207/msg00259.html`).

9.4.12 TrueType-Schriften unter X

Der Standard-X-Font-Server `xfs` in Free86-4 kommt problemlos mit TrueType Schriften zurecht. Wenn man XFree86-3 verwendet, braucht man zusätzlich den Font-Server `xfs-xtt`.

Wenn man selbst Programme kompiliert, muss man darauf achten, diese gegen libXft oder libfreetype zu linken. Wenn man nur die Pakete aus der Distribution verwendet, gibt es weiter nichts zu beachten.

Zu den neuen Schriften müssen die Dateien `fonts.{scale,dir}` erzeugt werden, da diese nötige Verwaltungsinformationen für den X-Server bereitstellen.

Da **freie** Schriften oft eingeschränkt sind, kann das Installieren von Shareware- oder kommerziellen TrueType-Schriften eine Alternative sein. Um diesen Schritt zu vereinfachen, gibt es einige Pakete:

- `ttf-commercial`
- `msttcorefonts (>1.1.0)` (Dieses Paket funktioniert in Woody seit August 2002 nicht mehr, da Microsoft die von dem Paket geladenen Schriften vom Server entfernt hat.)

Man muss schon eine gute Auswahl an Schriften haben, um das besonders frei gestaltete System mit kommerziellen Schriften zu 'verunreinigen'.

9.4.13 Web-Browser (graphische)

Es gibt eine Handvoll graphischer Web-Browser in Debian
- `mozilla` Der Mozilla-Browser (neu)
- `galeon` Mozilla-basierter Browser mit Gnome Benutzeroberfläche (neu)
- `konqueror` KDE-Browser
- `dillo` GTK-Browser
- `amaya-gtk` W3C-Referenz-Browser
- `amaya-lesstif` W3C-Referenz-Browser
- `netscape-...` (viele, alt)
- `communicator-...` (viele, alt)
- ...

Die Version des `mozilla` muss zu der `galeon` Version passen, da zweiteres die Gecko-HTML-Rendering-Engine des ersteren verwendet.

Plugins für die Browser wie `mozilla` und `galeon` können installiert werden, indem man einfach die "`*.so`" in das entsprechende Plugin-Verzeichnis kopiert (dies findet man z.B. mit `locate libnullplugin`).

Erhältliche Plugins:
- Java Plugin: Download des "J2SE" von `http://java.sun.com`.
- Flash Plugin: Download des "Macromedia Flash Player 5" von `http://www.macromedia.com/software/flashplayer/`.
- `freewrl`: VRML-Browser und Netscape-Plugin
- ...

9.5 SSH

SSH (Secure SHell) ist ein sicheres Werkzeug, um Verbindungen über das Internet herzustellen. Eine freie Version von SSH namens OpenSSH ist als `openssh` Paket in Debian enthalten.

9.5.1 Grundlagen

Zunächst müssen der OpenSSH-Server und -Client installiert werden.

```
# apt-get update && apt-get install ssh
```

Es darf keine Datei `/etc/ssh/sshd_not_to_be_run` bestehen.

SSH beherrscht zwei Arten von Authentisierungsprotokollen:

- SSH-Protokoll Version 1:
  - In Debian-Potato wird nur die Version 1 unterstützt.
  - Diese Version unterstützt:

    * RSAAuthentication: RSA Identitäts-Schlüssel basierte Benutzererkennung
    * RhostsAuthentication: Auf .rhosts basierende Rechnererkennung (unsicher, daher nicht aktiviert.)
    * RhostsRSAAuthentication: Authentisierung mit `.rhosts` kombiniert mit RSA Host-Schlüssel (auch deaktiviert)
    * ChallengeResponseAuthentication: RSA Challenge-Response Authentisierung
    * PasswordAuthentication: Passwortbasierte Authentisierung

- SSH Protokoll Version 2:
  - Alle Versionen seit Woody verwenden bevorzugt diese Version.
  - Diese Version unterstützt:
    * PubkeyAuthentication: Public-Key basierte Authentisierung
    * HostbasedAuthentication: `.rhosts` oder `/etc/hosts.equiv` Authentisierung kombiniert mit Public-Key Rechner Authentisierung (disabled)
    * ChallengeResponseAuthentication: challenge-response Authentisierung
    * PasswordAuthentication: Passwort basierte Authentisierung

Die unterschiedlichen unterstützten Protokolle müssen beachtet werden, wenn man von oder nach Woody migriert, da man sonst vielleicht keine SSH-Verbindungen mehr zu den gewohnten Rechnern aufbauen kann.

Siehe dazu auch `/usr/share/doc/ssh/README.Debian.gz`, `ssh(1)`, `sshd(8)`, `ssh-agent(1)` und `ssh-keygen(1)` für Details.

Die wichtigsten Konfigurationsdateien für SSH sind:

- `/etc/ssh/ssh_config`: SSH-Client Voreinstellungen. Siehe `ssh(1)`. Wichtige Einträge sind:
  - `Host`: Beschränkt die folgenden Restriktionen auf diesen Rechner bzw. dieses Rechnermuster (bis zum nächsten 'Host'-Eintrag).
  - `Protocol`: Spezifiziert die Version des SSH-Protokolls. Die Voreinstellung ist "2,1".
  - `PreferredAuthentications`: Spezifiziert die Methode mit der SSH2 authentisieren soll. Die Voreinstellung ist "hostbased,publickey,keyboard-interactive,password".
  - `ForwardX11`: Diese Option ist per Voreinstellung abgeschaltet. Sie kann über die Kommandozeilenoption "`-X`" überstimmt werden, so dass auch graphische Ausgaben übertragen werden dürfen.

- `/etc/ssh/sshd_config`: Voreinstellungen des SSH-Servers. Siehe `sshd(8)`. Wichtige Einträge sind:
  - `ListenAddress`: Spezifiziert die lokalen Adressen, auf denen `sshd` lauscht. Hier sind mehrere möglich.
  - `AllowTcpForwarding`: Voreinstellung: aus
  - `X11Forwarding`: Voreinstellung: aus.

- `$HOME/.ssh/authorized_keys`: In dieser Datei liegt die Liste der voreingestellten öffentlichen Schlüssel ('public keys'), mit denen Clients sich mit dem System verbunden haben. Siehe `ssh-keygen(1)`.

- `$HOME/.ssh/identity`: Siehe `ssh-add(1)` und `ssh-agent(1)`.

Mit folgendem Kommando kann man eine `ssh`-Verbindung aufbauen:

```
$ ssh username@hostname.domain.ext
$ ssh -1 username@hostname.domain.ext # SSH version 1 erzwungen
```

`ssh` ersetzt so auf sichere und praktische Weise `telnet`.

9.5.2 Port-Umleitung – für das Tunneln mit SMTP/POP3

Um eine Weiterleitung ('pipe') einzurichten, die den Port 25 eines entfernten Servers (*remote-server*) auf den Port 4025 lokal und den Port 110 des entfernten Servers auf den Port 4110 auf dem lokalen Rechner einzurichten, kann `ssh` folgendermaßen benutzt werden:

```
# ssh -q -L 4025:remote-server:25 4110:remote-server:110 \
           benutzername@remote-server
```

Auf diese Weise können sichere SMTP/POP3-Übertragungen realisiert werden. Man setze die Option `AllowTcpForwarding` auf `yes` in der Datei `/etc/ssh/sshd_config` auf dem entfernten Rechner.

9.5.3 Mit weniger Passwörter agieren

Man kann die sich immer wiederholende Eingabe von Passwörtern umgehen, wenn man auf das RSAAuthentication (SSH1 Protokoll)-Verfahren, bzw. PubkeyAuthentication (SSH2 Protokoll)-Verfahren zurückgreift.

Auf dem entfernten System wird dazu in der Konfigurationsdatei `/etc/ssh/sshd_config` "RSAAuthentication yes" bzw. "PubkeyAuthentication yes" gesetzt.

Dann werden Schlüssel lokal erzeugt und der dazugehörige öffentliche Schlüssel auf dem entfernten System hinterlegt.

```
$ ssh-keygen          # RSAAuthentication: RSA1 Schlüssel für SSH1
$ cat .ssh/identity.pub | ssh benutzer1@remote \
        "cat - >>.ssh/authorized_keys"
...
$ ssh-keygen -t rsa   # PubkeyAuthentication: RSA Schlüssel für SSH2
$ cat .ssh/id_rsa.pub | ssh benutzer1@remote \
        "cat - >>.ssh/authorized_keys"
...
$ ssh-keygen -t dsa   # PubkeyAuthentication: DSA Schlüssel für SSH2
$ cat .ssh/id_dsa.pub | ssh benutzer1@remote \
        "cat - >>.ssh/authorized_keys"
```

Später kann das Passwort mit "`ssh-keygen -p`" neu gesetzt werden. Bei Problemen können vermehrt Statusinformationen ausgegeben werden mit: "`ssh -v`".

In der Datei `authorized_keys` können zu einem beliebigen Rechner Optionen hinzugefügt werden, um dort z.B. Kommandos auszuführen. Siehe `sshd(8)` für weitere Details.

SSH2 verfügt über `HostbasedAuthentication`. Damit dies funktioniert, muss auf dem Server in der Datei `/etc/ssh/sshd_config` `HostbasedAuthentication` auf `yes` gesetzt werden, desgleichen in `/etc/ssh/ssh_config` oder `$HOME/.ssh/config` auf dem Client.

9.5.4 Weitere SSH-Client-Programme

Es gibt einige freie SSH-Client-Programme auf nicht-Unix-artigen Plattformen.

Windows puTTY (`http://www.chiark.greenend.org.uk/~sgtatham/putty/`) (GPL)

Windows (cygwin) SSH in cygwin (`http://www.cygwin.com/`) (GPL)

Macintosh Classic macSSH (`http://www.macssh.com/`) (GPL) [Mac OS X beinhaltet OpenSSH; es wird gestartet mit `ssh` im Terminal]

Siehe auch SourceForge.net, site documentation (`http://www.sourceforge.net/docman/?group_id=1`), "6. CVS Instructions".

9.5.5 SSH-Agent

Der öffentliche Schlüssel kann einfach in der Datei `~/.ssh/authorized_keys` hinterlegt werden, worauf er von ssh anerkannt wird.

```
$ ssh-agent
$ # die Ausgabe in der Shell einfügen
$ ssh-add .ssh/identity
$ # oder ssh-add .ssh/id_dsa oder wie auch immer der private Schlüssel heißt
$ scp remote.host.with.public.key
```

Weitere Informationen liefern `ssh-agent(1)` und `ssh-add(1)`.

9.5.6 Problemlösungen

Wenn Probleme bei der Benutzung von `ssh` auftreten, sollte man zuerst einmal die Berechtigungen der Konfigurationsdateien mittels der `ssh` Option `-v` prüfen.

Wenn man root-Rechte hat, kann mit `-P` die Benutzung der Ports 1–1023 unterbunden werden, welche bei Firewalls Schwierigkeiten bereiten.

Wenn `ssh`-Verbindungen, die ursprünglich funktionierten plötzlich nicht gestartet werden können, hat der Systemadministrator wahrscheinlich die Datei `host_key` geändert. Nachdem man sich vergewissert hat, dass dies der Fall ist und es sich nicht um den Versuch handelt, eine Verbindung zu einem gehackten Rechner zu etablieren, kann einfach die zu dem Remote-Rechner gehörige `host_key` Zeile in der Datei `$HOME/.ssh/known_hosts` gelöscht werden. Bei der nächsten Verbindungsaufnahme wird die Zeile neu generiert.

9.6 E-Mail Programme

Die E-Mail Konfiguration eines Systems zerfällt in drei Kategorien:

- Mail-Transfer-Agenten (mail transfer agent, MTA): `exim`
- Mail-Hilfsprogramme: `procmail`, `fetchmail`, `mail`, ...
- Mail-Benutzer-Agenten (mail user agent, MUA): `mutt`

9.6.1 E-Mail Transport/Transfer Agenten (Exim)

Literatur:
- `exim-doc` und `exim-doc-html` Pakete
- `http://www.exim.org/`

Wenn `exim` als MTA eingesetzt wird, müssen die folgenden Dateien angepasst werden:

```
/etc/exim/exim.conf     "eximconfig" bearbeitet diese Datei
/etc/inetd.conf         durch Auskommentieren von smtp läuft exim als Daemon
/etc/email-addresses    Add spoofed source address lists
Die Filter können mittels exim -brw, -bf, -bF, -bV, ...  getestet werden.
```

Ein Fangnetz für nicht existierende E-Mail-Adressen (Exim)

In der Datei `/etc/exim/exim.conf` (Woody oder später), kann am Ende des Abschnittes DIRECTORS (gleich nach localuser: director) ein Ausdruck eingefügt werden, der auf alle E-Mail-Adressen passt, die vorher nicht aufgelöst werden konnten (nach Miquel van Smoorenburg):

```
catchall:
  driver = smartuser
    new_address = webmaster@mydomain.com
```

Wer eine komplexere Anordnung von virtuellen Domänen angelegt hat, wird an das Ende der Datei `/etc/exim/exim.conf` einen Eintrag wie folgt vornehmen: (von mir, nicht sehr ausführlich getestet):

```
*@yourdomain.com ${lookup{$1}lsearch*{/etc/email-addresses} \
        {$value}fail} T
```

Dazu benötigt man einen "*" Eintrag in der Datei /etc/email-addresses.

Selektive Adressen-Umschreibung für ausgehende E-Mail (Exim)

Mit exim kann auch die From: Zeile der ausgehenden E-Mails manipuliert werden. Auch dies kann in der exim Konfigurationsdatei /etc/exim/exim.conf erledigt werden:

```
*@host1.something.dyndns.org \
  "${if eq {${lookup{$1}lsearch{/etc/passwd}{1}{0}}} {1}  \
   {$0}{$1@somethig.dyndns.org}}"  frFs
```

Dies schreibt alle Adressen um, die auf den Ausdruck *@host1.something.dyndns.org passen.

1 Zuerst wird in /etc/password nachgesehen, ob der Ausdruck $1 auf einen lokalen Benutzer zutrifft.

2 Wenn dies der Fall ist, wird daraus das, was es vorher schon war ($0).

3 Wenn es den Benutzer lokal nicht gibt, wird der Domänenanteil ersetzt.

9.6.2 Mail-Hilfsprogramme (Fetchmail)

fetchmail wird im Daemon-Modus gestartet und ist für die Abholung von E-Mail von POP3-Accounts von einem ISP zuständig, um die Mails dann in das lokale E-Mail-System einzufügen. Die Konfiguration geschieht über die Dateien:

```
/etc/init.d/fetchmail
/etc/rc?.d/???fetchmail startet update-rc.d fetchmail Default Priorität 30
/etc/fetchmailrc        Konfigurationsdatei (chown 600, Besitzer ist fetchmail)
```

In der alten Debian-Version Potato war das Starten von fetchmail aus init.d heraus noch schwierig. Dies ist mit Woody behoben. Beispiele sind außerdem zu finden in den Dateien /etc/init.d/fetchmail und /etc/fetchmailrc unter der Adresse example scripts (http://www.debian.org/doc/manuals/debian-reference/examples/).

Wenn im E-Mail-Kopf ^M Zeichen vorkommen, sollte "stripcr" in den Optionen der Datei $HOME/.fetchmailrc hinzugefügt werden:

```
options fetchall no keep stripcr
```

9.6.3 Mail-Hilfsprogramme (Procmail)

`procmail` ist ein Programm zur lokalen Mail-Zustellung und kann auch als Filter verwendet werden. Die Datei `$HOME/.procmailrc` wird für jeden Benutzer benötigt. Beispiel: _procmailrc (`http://www.debian.org/doc/manuals/debian-reference/examples/`)

9.6.4 E-Mail-Programme (Mail user agent) am Beispiel Mutt

Ein beliebtes E-Mail-Programm für die Konsole ist `mutt` in Kombination mit `vim`. Einstellungen liegen in `~/.muttrc`; z.B.

```
# diese Einstellungen modifizieren
# das Verhalten des Editors.
set editor="vim -c 'set tw=72 et ft=mail'"
#
# header weeding taken from the manual (Sven's Draconian header weeding)
#
ignore *
unignore from: date subject to cc
unignore user-agent x-mailer
hdr_order from subject to cc date user-agent x-mailer
auto_view application/msword
...
```

Um HTML-E-Mail oder MS-Word-Dokumente als Text anzeigen zu können, muss `antiword` installiert sein und folgende Zeilen an entweder `/etc/mailcap` (global) oder `$HOME/.mailcap` (per User) angehängt werden.

```
text/html; lynx -force_html %s; needsterminal;
application/msword; /usr/bin/antiword '%s'; copiousoutput;
description="Microsoft Word Text"; nametemplate=%s.doc
```

9.7 Lokalisation und Sprachen

Debian ist eine internationale Distribution, die eine wachsende Anzahl an Sprachen und regionaler Besonderheiten berücksichtigt. Die folgenden Abschnitte zeigen, in welcher Form Debian dies bewerkstelligt und an welchen Stellen eine Lokalisierung eingestellt werden kann. Dies betrifft in erster Linie Ein- und Ausgaben in der jeweiligen Muttersprache, Daten, numerische und finanzielle Formate, die von System zu System unterschiedlich sein können.

9.7.1 Grundlagen

Es folgen die wichtigsten Anpassungen für die Lokalisierung.

Tastatur

Debian stellt Tastaturbelegungen für nahezu zwei Dutzend Tastaturvarianten bereit. In Woody, können diese eingestellt werden mit:

- `dpkg-reconfigure --priority=low console-data # Konsole`
- `dpkg-reconfigure --priority=low xserver-xfree86 # X4`
- `dpkg-reconfigure --priority=low xserver-common-v3 # X3`

Daten

Die große Mehrheit an Debian-Software unterstützt den Umgang mit nicht-US-ASCII-Zeichen durch die LC_CTYPE Umgebungsvariable, welche den **locale**-Mechanismus der glibc konfiguriert.

- 8-bit clean: praktisch alle Programme
- andere lateinische Buchstaben (z.B. ISO-8859-1 oder ISO-8859-2): viele Programme
- multi-byte Sprachen wie Chinesisch, Japanisch oder Koreanisch: viele neue Programm

Anzeige

X-Windows kann jede Zeichen-Kodierung anzeigen, inklusive UTF-8 und unterstützt viele Schrifttechnologien. Die Unterstützung reicht über 8-bit-Schriften über 16-bit-Schriften für Chinesisch, Japanisch oder Koreanisch. Multi-byte-Zeichen werden über den XIM-Mechanismus unterstützt. Siehe 'Beispiel für ein zweisprachiges System (japanisches EUC und ISO-8859-1)' auf Seite 159.

Japanischer EUC-Kode kann auch (mit dem Paket `kon2`) auf der (S)VGA-Konsole angezeigt werden. Eine neue Möglichkeit, welche die FB-Konsole verwendet ist `jfbterm`. In diesen Umgebungen muss allerdings die Applikation selbst die Unterstützung dieser Sprachen mitbringen. `egg` ist ein Paket für Emacs und `jvim` ein japanisiertes Vim.

Übersetzungen

Viele Text-Nachrichten und Dokumente für das Debian-System liegen übersetzt vor, z.B. Fehlermeldungen, Standardausgaben, Menüs und Handbuchseiten. Derzeit gibt es Projekte zur Übersetzung in: Deutsch, Spanisch, Finnisch, Französisch, Ungarisch, Italienisch, Japanisch, Koreanisch und Polnisch. Hierzu muss natürlich das entsprechende Paket in der Form `manpages-`*`LANG`* installiert sein. (*LANG* ist hier ein eindeutiger Sprachkennzeichner gemäß den zweibuchstabigen ISO-Länderkodes.) Um herauszufinden, welche Pakete zur Auswahl stehen, kann `apt-cache search manpages- | less` verwendet werden.

Um eine NLS Handbuchseite anzuzeigen, muss die Umgebungsvariable LC_MESSAGES gesetzt sein. Um z.B. deutsche Handbuchseiten zu sehen, muss `LC_MESSAGES=de` gesetzt sein. Das `man` Programm wird die Handbuchseite dann in `/usr/share/man/de/` suchen.

9.7.2 Lokalisierung mit locale

Debian unterstützt die **locale** Technologie. Locale ist ein System zur Ausgabe von Zeichen, Daten und Zeiten und Währungseinheiten entsprechend lokaler Gegebenheiten. Es verwendet Umgebungsvariablen, um die Region festzustellen. Wenn z.B. englische und deutsche locale Pakete installiert sind, werden die Fehlermeldungen mehrsprachig.

```
$ LANG="en_US" cat foo
cat: foo: No such file or directory
$ LANG="de_DE" cat foo
cat: foo: Datei oder Verzeichnis nicht gefunden
```

Diese Funktionalität wird von der glibc bereitgestellt. (`localeconf` erlaubt eine komfortable Konfiguration der locale-Einstellungen.) Siehe auch `locale(7)`.

9.7.3 Introduction to locales

Full locale description consists of 3 parts: `xx_YY.ZZZZ`.

- **`xx`**: ISO 639 language codes (lower case)
- **`YY`**: ISO 3166 country codes (upper case)
- **`ZZZZ`**: codeset, i.e., character set or encoding identifier.

For language codes and country codes, see pertinent description in the `info gettext`.

Please note this codeset part may be normalized internally to achieve cross platform compatibility by removing all - and by converting all characters into lower case. Typical codesets are:

- **UTF-8**: Unicode for all regions, mostly in 1-3 Octets (new de facto standard)
- **ISO-8859-1**: western Europe (de facto old standard)
- **ISO-8859-2**: eastern Europe (Bosnian, Croatian, Czech, Hungarian, Polish, Romanian, Serbian, Slovak, Slovenian)
- **ISO-8859-3**: Maltese
- **ISO-8859-5**: Macedonian, Serbian
- **ISO-8859-6**: Arabic
- **ISO-8859-7**: Greek
- **ISO-8859-8**: Hebrew
- **ISO-8859-9**: Turkish
- **ISO-8859-11**: Thai (=TIS-620)
- **ISO-8859-13**: Latvian, Lithuanian, Maori
- **ISO-8859-14**: Welsh
- **ISO-8859-15**: western Europe with euro
- **KOI8-R**: Russian
- **KOI8-U**: Ukrainian
- **CP1250**: Czech, Hungarian, Polish (MS Windows origin)
- **CP1251**: Bulgarian, Byelorussian (MS Windows origin)

- **eucJP**: Unix style Japanese (=ujis)
- **eucKR**: Unix style Korean
- **GB2312**: Unix style Simplified Chinese (=GB, =eucCN) for zh_CN
- **Big5**: Traditional Chinese for zh_TW
- **sjis**: Microsoft style Japanese (Shift-JIS)

As for the meaning of basic encoding system jargons:

- **ASCII**: 7 bits (0-0x7f)
- **ISO-8859-?**: 8 bits (0-0xff)
- **ISO-10646-1**: Universal Character Set (UCS) (31 bits, 0-0x7fffffff)
- **UCS-2**: First 16 bit of UCS as straight 2 Octets (Unicode: 0-0xffff)
- **UCS-4**: UCS as straight 4 Octets (UCS: 0-0x7fffffff)
- **UTF-8**: UCS encoded in 1-6 Octets (mostly in 3 Octets)
- **ISO-2022**: 7 bits (0-0xff) with the escape sequence. ISO-2022-JP is the most popular encoding for the Japanese e-mail.
- **EUC**: 8 bits + 16 bits combination (0-0xff), Unix style
- **Shift-JIS**: 8 bits + 16 bits combination (0-0xff), Microsoft style.

ISO-8859-?, EUC, ISO-10646-1, UCS-2, UCS-4, and UTF-8 share the same code with ASCII for the 7 bit characters. EUC or Shift-JIS uses high-bit characters (0x80-0xff) to indicate that part of encoding is 16 bit. UTF-8 also uses high-bit characters (0x80-0xff) to indicate non 7 bit character sequence bytes and this is the most sane encoding system to handle non-ASCII characters.

Please note the byte order difference of Unicode implementation:

- **Standard UCS-2, UCS-4**: big endian
- **Microsoft UCS-2, UCS-4**: little endian for ix86 (machine-dependent)

See 'Konvertierung einer Textdatei mit `recode`' auf Seite 108 for conversion between various character sets. For more see Introduction to i18n (`http://www.debian.org/doc/manuals/intro-i18n/`).

9.7.4 Aktivierung der Lokalisierung

Debian hat nicht alle verschiedenen Sprachen einkompiliert. Die Datei `/usr/lib/locale` zeigt, welche Sprachen sofort verfügbar sind (neben der Standardeinstellung "C"). Wenn die benötigte Sprache fehlt, gibt es zwei Möglichkeiten:

- Hinzufügen der gewünschten Locale zu `/etc/locale.gen`, anschließend `locale-gen` starten, um die Locale zu erzeugen. Siehe dazu `locale-gen(8)` und die Handbuchseiten im "SEE ALSO"-Abschnitt dieser Handbuchseite.
- Starte `dpkg-reconfigure locales`, um das `locales` Paket neu zu konfigurieren. Auch bei Neuinstallation des Paketes `locales` wird das Konfigurationstool gestartet, so dass alle Einstellungen vorgenommen werden können.

9.7.5 Eine Locale aktivieren

Die folgenden Umgebungsvariablen werden ausgewertet, um die lokalen Formate zu berücksichtigen.

1 LANGUAGE: Diese Umgebungsvariable enthält eine durch Doppelpunkte separierte Liste von Sprachen (mit absteigender Priorität). Sie wird nur verwendet, wenn die POSIX-locale nicht "C" ist [in Woody; in Potato hat LANGUAGE immer Priorität vor POSIX]. (GNU-Erweiterung)

2 LC_ALL: Wenn dieser Eintrag nicht leer ist, wird er für alle Kategorien verwendet. (POSIX.1) Voreinstellung ist "" (null).

3 LC_*: Wenn dieser Eintrag nicht leer ist, wird er für die entsprechende Kategorie gesetzt. (POSIX.1). Voreinstellung "C".

   Mögliche LC_*-Variablen sind:
   - LC_CTYPE: Buchstaben-Klassifikation und Groß-/Kleinumschaltung.
   - LC_COLLATE: Collation-Reihenfolge. (???)
   - LC_TIME: Datum- und Zeit-Format.
   - LC_NUMERIC: Zahlenformat.
   - LC_MONETARY: Geld-Format.
   - LC_MESSAGES: Format der Meldungen und Diagnostik-Anzeigen, auch für interaktive Eingaben.
   - LC_PAPER: Papier-Format.
   - LC_NAME: Namen-Format.
   - LC_ADDRESS: Adressen-Format und Orts-Information.
   - LC_TELEPHONE: Telefonnummer-Format.
   - LC_MEASUREMENT: Maßeinheiten (Metrisch oder andere).
   - LC_IDENTIFICATION: Metadaten über die locale-Einstellung.

4 LANG: Wenn diese nicht leer ist und LC_ALL nicht definiert ist, wird dieser Wert für alle bis dahin nicht definierten Variablen gesetzt. (POSIX.1) Voreinstellung "C".

Einige Programme ignorieren allerdings diese Variablen. In der Regel haben diese Programme ein eigenes Sprachen-Management (z.B. Netscape Navigator 4).

Das `locale`-Programm kann die aktiven locale-Einstellungen sowie andere installierte locale anzeigen. Siehe `locale(1)`. (NOTIZ: `locale -a` zeigt alle locales an, die das System kennt; das bedeutet *nicht*, dass sie verwendbar sind (weil unübersetzt). 'Aktivierung der Lokalisierung' auf der vorherigen Seite.)

9.7.6 ISO 8601 Datum-Format-locale

Die Unterstützung für die Darstellung des internationalen Datums wird durch die locale `en_DK` – **Englisch in Dänemark** – in Form des ISO 8601-Standards (`yyyy-mm-dd`) geleistet. Dies scheint aber nur für die Konsole mit `ls` zu funktionieren.

9.7.7 Beispiel für die USA (ISO-8859-1)

Zu der Datei `~/.bash_profile` Folgendes hinzufügen:

```
LC_CTYPE=en_US.ISO-8859-1
export LC_CTYPE
```

9.7.8 Beispiel für Frankreich mit Euro-Symbol (ISO-8859-15)

Zu der Datei `~/.bash_profile` Folgendes hinzufügen:

```
LANG=fr_FR@euro
export LANG
LC_CTYPE=fr_FR@euro
export LC_CTYPE
```

Die Tastatur kann auf das deutsche "QWERTZ" umgestellt werden wie beschrieben in 'Tastatur' auf Seite 155. Die deutschen Handbuchseiten befinden sich im Paket `manpages-de`. Die rechte Alt-Taste heißt in Europa Alt-Gr. Durch Drücken dieser Taste können diverse Sonderzeichen erreicht werden. So erzeugt z.B. AltGr-E das Euro-Symbol.

Die meisten westeuropäischen Sprachen können wie beschrieben eingestellt werden.

Siehe Debian Euro HOWTO (`http://www.debian.org/doc/manuals/debian-euro-support/`) für die Konfiguration des Euro-Symbols. Das Paket XXX für weitere Details zur Nutzung deutscher Anpassungen.

9.7.9 Beispiel für ein zweisprachiges System (japanisches EUC und ISO-8859-1)

In diesem Beispiel wird ein System aufgesetzt, das japanisch und englisch unterstützt. Die Umgebungsvariable für die japanische Sprache heißt in diesem Fall ja_JP.eucJP (japanisches EUC für traditionelle japanische Unix Umgebung). Die Unterstützung betrifft X mit gleichzeitig englischen Ausgaben und Datumsangaben nach ISO-Standard. Hinzu kommt Englisch als primäre Sprache der Text-Konsole mit der Umgebungseinstellung en_US.ISO-8859-1 (ASCII mit zusätzlichen Akzenten).

- zuerst wird eine japanische Lokale ja_JP.eucJP erzeugt, wie beschrieben in 'Lokalisation und Sprachen' auf Seite 154.
- es wird ein Kana-nach-Kanji-Konverter und ein Wörterbuch installiert:
  - `canna` – Lokaler Server (NEC free-license) oder
  - `freewnn-jserver` – Netzwerk-basierter erweiterbarer Server (Public Domain)
- Installation einer Eingabemethode für Japanisch:
  - `kinput2-canna` – für X oder
  - `kinput2-canna-wnn` – für X und
  - `egg` – funktioniert mit Emacsen, optional auch im Konsole-Modus
- Japanisch-kompatibles Terminal:

  – `kterm` – X (klassisch),
  – `mlterm` – X (sehr nett, verschiedene Schriftgrößen) und

- füge japanische Schriften hinzu.
- Konfiguriere X wie beschrieben in ‘Angepasste X-Sitzung’ auf Seite 141. Eine an den Benutzer angepasste X-Umgebung kann in Abhängigkeit der Start-Methode von X eingestellt werden. (`startx`, `xdm`, . . .)
- Letztlich wird eine `~/.xsession` Datei erzeugt, in der die Einstellungen vorgenommen werden:

```
#!/bin/sh
# This makes X work when I su to root.
if [ -z "$XAUTHORITY" ]; then
        XAUTHORITY=$HOME/.Xauthority
        export XAUTHORITY
fi

# Japanische locale als Default, C locale als Backup
LANG=ja_JP.eucJP
export LANG
# In der Konsole wollen wir en_US.ISO-8859-1
LC_CTYPE=ja_JP.eucJP
export LC_CTYPE
# menu-Nachrichten in Englisch und ASCII :-)
LC_MESSAGES=C
export LC_MESSAGES
# aktiviere Eingabe-Methode
kinput2 &
XMODIFIERS=@im=kinput2
export XMODIFIERS
# blackbox ist ein guter Window-Manager (lightweight)
exec /usr/bin/blackbox
```

- wir fügen folgende Zeilen zu `~/.bash_profile` hinzu:

```
LC_CTYPE=en_US.ISO-8859-1
export LC_CTYPE
# Für ISO yyyy-mm-dd Datumsanzeige kommt dem japanischen mehr entgegen
LC_TIME=en_DK.ISO-8859-1
export LC_TIME
```

- folgende Zeilen können, falls mutt als Mail-Programm verwendet wird, zu `~/.muttrc` hinzugefügt werden:

```
# UTF-8 Unterstützung ist in Japan bei EMACS nicht üblich
# 7 bit Kodierung mittels iso-2022-jp ist einfacher
# Default Reihenfolge = us-ascii --> iso-8859-1 --> utf-8
#set send_charset="us-ascii:iso-8859-1:utf-8"
#set allow_8bit=yes
set send_charset="us-ascii:iso-8859-1:iso-2022-jp"
set allow_8bit=no
```

- aktiviere XIM `kinput2` für X-Applikationen
  - füge `*inputMethod: kinput2` der .Xresources Datei hinzu: `~/.Xresources` (teilweise automatisch in Debian)
  - Einige Programme erlauben sogar die Änderung der Kodierung während der Laufzeit (`mlterm` kann diese dynamisch mit **Strg-Mausknopf-3** ändern.
- Nun wird X gestartet: `startx` oder durch einen Displaymanager (xdm, gdm, kdm, wdm, ...)
- In X wird ein japanisch-fähiges Programm gestartet: VIM6, (x)emacs21, mc-4.5, mutt-1.4, ... (Emacs ist populär. Aber ich verwende es nicht.)
- "**Shift+Space**" schaltet die japanische Unterstützung ein und aus.

Sieh auch die SuSE Seiten für CJK (`http://www.suse.de/~mfabian/suse-cjk/suse-cjk.html`).

9.7.10 Beispiel für UTF-8 in X

In Zukunft wird Unicode alle Schriftprobleme lösen. Siehe Das Unicode HOWTO (`http://www.tldp.org/HOWTO/Unicode-HOWTO.html`).

9.7.11 Beispiel für UTF-8 in der FB-Konsole

Die UTF-8 Unterstützung in der Konsole wird vom Paket `bterm` bereitgestellt. Der `debian-installer` verwendet dies.

9.7.12 Über locale hinaus

Wenn eine größere Anpassung an eine Sprache gewünscht wird, sollte man zunächst nachsehen, ob `tasksel` oder `aptitude` bereits eine Option für die betreffende Sprache hat. So kann eine Paketsammlung geladen werden, die bereits Dateien für die nationale Anpassung enthält, was auch für mehrsprachige Systeme Sinn macht. Wenn bei einem System mit feinjustierter Sprachunterstützung Paketkonflikte auftreten, sollte zunächst die Ursache genau festgestellt werden, damit die Spracheinstellungen nicht verloren gehen. Eventuell muss `update-alternative` verwendet werden, um den vorherigen Status wieder her zu stellen.

Die meisten größeren Programme nutzen bereits die glibc 2.2 und sind größtenteils internationalisiert. Aus diesem Grund sind lokalisierte Versionen eines speziellen Programms wie z.B. VIM überflüssig. So wird die Funktionalität von `jvim` für VIM von `vim` ab Version 6.0 in X automatisch bereitgestellt. Tatsächlich ist die Unterstützung noch etwas holprig. Da `jvim` die japanische Unterstützung eingebaut hat, sollte man sich das Programm zumindest doch einmal ansehen.

Es kann sein, dass Programme über die `locale` Konfiguration hinaus angepasst werden müssen. Das `language-env` Paket und dessen Kommando `set-language-env` erleichtern diesen Schritt.

Siehe auch die Internationalisierungsdokumente unter Einführung in i18n (`http://www.debian.org/doc/manuals/intro-i18n/`). Es richtet sich an Entwickler und Systemadministratoren.

Kapitel 10

Netzwerk-Konfiguration

Dieses Kapitel konzentriert sich auf die Netzwerk-Administration in Debian. Eine allgemeinere Einführung in GNU/Linux findet sich in Net-HOWTO (`http://www.tldp.org/HOWTO/Net-HOWTO/index.html`).

Um sich mit einem Debian-Rechner mit dem Internet zu verbinden, muss seine Netzwerk-Schnittstelle korrekt konfiguriert sein.

Die erste Voraussetzung ist Kernel-Unterstützung für die vorhandenen Geräte. Beispiele solcher Geräte sind: Ethernet-Karten, Wi-Fi-Karten und weitere. Um diese Unterstützung des Kernels zu erhalten, kann es nötig sein, den Kernel neu zu kompilieren oder Module hinzuzufügen, so wie es in 'Der Linux-Kernel unter Debian' auf Seite 85 beschrieben ist.

Die Konfiguration der Netzwerk-Geräte wird weiter unten beschrieben. Die Angaben in diesem Kapitel sind für Sarge aktualisiert worden. Vieles vom hier gesagten gilt nicht mehr für vorherige Versionen von Debian.

10.1 Netzwerk-Grundlagen mit IP

Ein Debian-Rechner kann mehrere Schnittstellen besitzen, die jeweils unterschiedliche IP (Internet Protokoll)-Adressen haben können. Diese Schnittstellen können auch unterschiedlichen Typs sein, darunter:

- Loopback: `lo`
- Ethernet: `eth0`, `eth1`, ...
- Wi-Fi: `wlan0`, `wlan1`, `wifi0`, ... [1]
- Token Ring: `tr0`, `tr1`, ...
- PPP: `ppp0`, `ppp1`, ...

Es gibt eine Vielzahl anderer Netzwerkgeräte, darunter SLIP, PLIP (IP über serielle und parallele Verbindungen), "shaper"-Geräte, die das Datenaufkommen auf bestimmten Geräten steuern, Frame-Relay, AX.25, X.25, ARCnet und LocalTalk.

[1] Einige Wi-Fi-Schnittstellen sind tatsächlich ein Alias für Ethernet-Schnittstellen, die Zugriff auf die Konfigurationsparameter erlauben, die speziell für Wi-Fi gelten. Diese Parameter können mit dem Programm `ifconfig` eingestellt werden.

Jede Netzwerkschnittstelle, die direkt mit dem Internet (oder IP-basierten Netzwerk) verbunden ist, wird durch eine eindeutige 32-Bit IP-Adresse identifiziert.[2] Die IP-Adresse können in einen Teil für Netzwerkadressierung und einen Teil zur Adressierung des Rechners geteilt werden. Wenn in einer IP-Adresse alle Bits, die Teil der Netzwerkadresse sind, auf Eins gesetzt werden und alle Bits, die Teil der Host-Adresse sind, auf Null gesetzt werden, dann erhält man die so genannte Netzmaske (netmask) des Netzwerks.

IP-Netzwerke sind traditionell in Klassen eingeteilt, deren Netzwerkadressen 8, 16 oder 24 Bit lang sind. Dieses System war nicht flexibel und verschwendete viele IP-Adressen, so dass IPv4-Netzwerke heute unterschiedlich lange Netzwerkadressen zugewiesen bekommen.

```
        IP-Adressen                     Netzmasken      Länge
Class A 1.0.0.0 - 126.255.255.255       255.0.0.0       = /8
Class B 128.0.0.0 - 191.255.255.255     255.255.0.0     = /16
Class C 192.0.0.0 - 223.255.255.255     255.255.255.0   = /24
```

IP-Adressen, die nicht darin liegen, sind für besondere Zwecke reserviert.

Einige Adressbereiche innerhalb jeder Klasse sind für lokale Netzwerke (LANs) reserviert. Diese Adressen kollidieren garantiert nicht mit irgendwelchen Adressen des Internet. (Aus diesem Grund können die Rechner, die eine solche Adresse zugewiesen bekommen, nicht direkt mit dem Internet verbunden sein, sondern müssen einen Gateway-Rechner als Zwischenschritt verwenden, der entweder die einzelnen Services anbietet oder Network-Adress-Translation (NAT) durchführt.) Diese Adressbereiche und die Anzahl der Bereiche in jeder Klasse sind in der folgenden Tabelle dargestellt.

```
        Netzwerkadressen                Länge   Anzahl
Class A 10.x.x.x                        /8      1
Class B 172.16.x.x - 172.31.x.x         /16     16
Class C 192.168.0.x - 192.168.255.x     /24     256
```

Die erste Adresse in einem IP-Netzwerk ist die Adresse des Netzwerks selbst. Die letzte Adresse ist die Broadcast-Adresse des Netzwerks.[3] Alle anderen Adressen können einzelnen Rechnern des Netzwerks zugewiesen werden. Üblicherweise wird die erste oder letzte Adresse eines Netzwerks für den Gateway-Rechner verwendet.

Die Routing-Tabelle enthält vom Kernel bereitgestellte Informationen darüber, wie IP-Pakete an ihre Ziele verschickt werden. Hier eine Beispieltabelle eines Debian-Rechners in einem lokalen Netzwerk mit der IP-Adresse 192.168.50.x/24. Der Rechner 192.168.50.1 (ebenfalls im LAN) ist ein Router für das Firmennetzwerk, 172.20.x.x/16 und 192.168.50.254 (ebenfalls im LAN) sind Router für das Internet.

[2] Dies gilt für IP-Version 4. IP-Adressen in der Version 6 bestehen aus 128 Bit. Siehe `http://www.ipv6.org/`.

[3] Die Adresse des Netzwerks kann berechnet werden, indem die Adresse bitweise mit der Netzmaske via UND verknüpft wird. Die Broadcast-Adresse kann berechnet werden, indem die Netzwerkadresse bitweise mit dem Komplement der Einsen geODERt wird.

```
# route
Kernel IP routing table
Destination   Gateway         Genmask         Flags Metric Ref Use Iface
127.0.0.0     *               255.0.0.0       U     0      0     2 lo
192.168.50.0  *               255.255.255.0   U     0      0   137 eth0
172.20.0.0    192.168.50.1    255.255.0.0     UG    1      0     7 eth0
default       192.168.50.254  0.0.0.0         UG    1      0    36 eth0
```

- Die erste Zeile nach der Kopfzeile bestimmt, dass Daten für das Netzwerk `127.x.x.x` durch das Loopback-Device `lo` geroutet werden.
- Die zweite Zeile bestimmt, dass Daten für Hosts im LAN über die `eth0`-Schnittstelle versendet werden.
- Die dritte Zeile bestimmt, dass Daten für das Firmennetzwerk ebenfalls über `eth0` an das Gateway `192.168.50.1` versendet werden.
- Die vierte Zeile bestimmt, dass Daten für das Internet ebenfalls über `eth0` an das Gateway `192.168.50.254` gesendet werden.

Die IP-Adressen in dieser Tabelle können auch also Namen dargestellt werden, so wie sie in der Datei `/etc/networks` hinterlegt sind oder wie sie von der C-Bibliothek aufgelöst werden.

Der Kernel ist neben seiner Aufgabe als Router auch in der Lage, Network Address Translation (NAT), Traffic Shaping und Filtering durchzuführen.

Siehe Net-HOWTO (`http://www.tldp.org/HOWTO/Net-HOWTO/index.html`) und other networking HOWTOs (`http://www.tldp.org/HOWTO/Networking-Overview-HOWTO.html`) für weitere Informationen zu diesem Thema.

10.2 Netzwerk-Konfiguration auf niederer Ebene

Die traditionellen Werkzeuge, um unter GNU/Linux die Netzwerkkonfiguration auf niederer Ebene vorzunehmen, sind `ifconfig` und `route`, welche im Paket `net-tools` enthalten sind. Diese Werkzeuge wurden offiziel von `ip` abgelöst, welches sich im Paket `iproute` befindet. Das Programm `ip` funktioniert ab Linux 2.2 und leistet mehr als die alten Werkzeuge. Die alten Werkzeuge funktionieren jedoch noch immer und sind mehr Benutzern bekannt.

10.2.1 Netzwerkkonfiguration auf niederer Ebene – `ifconfig` und `route`

An einem Beispiel soll gezeigt werden, wie die IP-Adresse der Schnittstelle `eth0` von der Adresse `192.168.0.3` auf `192.168.0.111` zu ändern und `eth0` als Route für das Netzwerk `10.0.0.0` via `192.168.0.1` einzurichten ist. Wir starten `ifconfig` und `route` ohne Argumente, um den derzeitigen Status aller Netzwerkschnittstellen und des Routing darzustellen.

```
# ifconfig
eth0    Link encap:Ethernet HWaddr 08:00:46:7A:02:B0
        inet addr:192.168.0.3 Bcast:192.168.255.255 Mask:255.255.0.0
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:23363 errors:0 dropped:0 overruns:0 frame:0
        TX packets:21798 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:100
        RX bytes:13479541 (12.8 MiB) TX bytes:20262643 (19.3 MiB)
        Interrupt:9

lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        UP LOOPBACK RUNNING MTU:16436 Metric:1
        RX packets:230172 errors:0 dropped:0 overruns:0 frame:0
        TX packets:230172 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:22685256 (21.6 MiB) TX bytes:22685256 (21.6 MiB)

# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags   Metric  Ref     Use     Iface
192.168.0.0     *               255.255.0.0     U       0       0       0       eth0
default         192.168.0.1     255.255.255.255 UG      0       0       0       eth0
```

Zunächst schalten wir die Schnittstelle ab.

```
# ifconfig eth0 inet down
# ifconfig
lo Link encap:Local Loopback
        ... (keine weiteren eth0-Einträge)
# route
        ... (keine weiteren Routing-Einträge)
```

Dann schalten wir sie mit einer neuen IP-Adresse und neuem Routing wieder ein.

```
# ifconfig eth0 inet up 192.168.0.111 \
        netmask 255.255.255.0 broadcast 192.168.0.255
# route add -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.0.1 dev eth0
```

Das Ergebnis:

```
# ifconfig
eth0 Link encap:Ethernet HWaddr 08:00:46:7A:02:B0
        inet addr:192.168.0.111 Bcast:192.168.0.255 Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        ...
lo   Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0 ...

# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags   Metric  Ref     Use     Iface
192.168.0.0     *               255.255.255.0   U       0       0       0       eth0
10.0.0.0        192.168.0.1     255.0.0.0       UG      0       0       0       eth0
```

Weitere Informationen sind in `ifconfig(8)` und `route(8)` zu finden.

10.2.2 Netzwerkkonfiguration auf niederer Ebene – `ip`

Das Programm `ip` erledigt folgendermaßen das gleiche wie die obigen Aufrufe von `ifconfig` und `route`

- `ip link show`
- `ip route list`
- `ip link set eth0 down`
- `ip addr del dev eth0 local 192.168.0.3`
- `ip addr add dev eth0 local 192.168.0.111/24 broadcast 192.168.0.255`
- `ip link set eth0 up`
- `ip route add dev eth0 to 10.0.0.0/8 src 192.168.0.111 via 192.168.0.1`

Eine Befehlsübersicht des Programms `ip` kann mittels des Arguments `help` aufgerufen werden. Zum Beispiel druckt `ip link help`:

```
Usage: ip link set DEVICE { up | down | arp { on | off } |
                        dynamic { on | off } |
                        multicast { on | off } | txqueuelen PACKETS |
                        name NEWNAME |
                        address LLADDR | broadcast LLADDR |
                        mtu MTU}
        ip link show [ DEVICE ]
```

Siehe auch `ip(8)`.

10.2.3 Konfiguration der Wi-Fi-Schnittstelle

Die Wi-Fi-Schnittstelle wird - zusätzlich zu `ifconfig` oder `ip` - mit `iwconfig` program konfiguriert und befindet sich im Paket `wireless-tools`.

Siehe `iwconfig(8)`.

10.2.4 Konfiguration der PPP-Schnittstelle

Wenn mit einem Modem über eine Telefonleitung auf das Internet zugegriffen werden soll, dann wird die Verbindung mit Hilfe des Point-to-Point-Protokolls (PPP) aufgebaut. Derartige Verbindungen werden als Netzwerkschnittstellen `ppp0`, `ppp1`, usw. dargestellt.

Die PPP-Schnittstelle wird vom PPP-Dämon `pppd` verwaltet, der sich im Paket `ppp` befindet. Möchte der Benutzer die PPP-Schnittstelle konfigurieren, so bedeutet dies `pppd` zu konfigurieren.

Manuelle `pppd`-Konfiguration

Um eine Netzwerkverbindung herzustellen, muss ein Kommunikationskanal geöffnet werden (üblicherweise eine serielle Schnittstelle), Befehle müssen an das Kommunikationsgerät (meist ein Modem) gesendet werden, eine Telefonnummer muss gewählt werden, die Identität des Benutzers muss dem fremden PPP-Dämon bestätigt werden, der Kernel muss eine PPP-Schnittstelle erzeugen, die Routing-Tabelle muss geändert werden, so dass der Datenverkehr über diese Schnittstelle abgewickelt wird. `pppd` kann all dieses leisten und hat daher eine sehr lange Liste von Optionen. Diese Optionen sind in `pppd(8)` beschrieben.

Auf einem Debian-System können globale Einstellungen in der Datei `/etc/ppp/options` abgelegt werden. Benutzerspezifische Optionen können in `~/.ppprc` gespeichert werden. Optionen, die vom jeweiligen Port abhängen werden in `/etc/ppp/options.`*`portname`* gespeichert. Zum Beispiel seien zwei Modems vorhanden: Ein eingebautes Lucent MT-Modem, auf das via /dev/LT-modem zugegriffen wird, und ein externes Modem, auf das via /dev/ttyS0 zugegriffen wird. Zur Konfiguration dienen zwei Dateien:

```
# cat > /etc/ppp/options.LT-modem <<EOF
115200
init "/usr/sbin/chat -f /etc/chatscripts/setup-LT-modem"
EOF
# cat > /etc/ppp/options.ttyS0 <<EOF
115200 init "/usr/sbin/chat -f /etc/chatscripts/setup-ttyS0"
EOF
```

Diese verweisen auf die folgenden `chat`-Skripte. Erstens `/etc/chatscripts /setup-LT-modem`.

```
ABORT ERROR
'' ATZ
OK 'ATW2X2 S7=70 S11=55'
OK AT
```

Zweitens: `/etc/chatscripts/setup-ttyS0`.

```
ABORT ERROR
'' ATZ
OK 'ATL1M1Q0V1W2X4&C1&D2 S6=4 S7=70 S11=55 S95=63 S109=1 +FCLASS=0'
OK AT
```

Der Inhalt dieser Dateien muss selbstverständlich an die jeweilige Hardware angepasst werden.

Optionen können dem `pppd`-Dämon auch als Argumente übergeben werden.

In Debian wird `pppd` üblicherweise durch das Kommando `pon` gestartet. Das erste Argument von `pon` bezeichnet die Optionen-Datei in `/etc/pppp/peers/`, welche dann von `pppd` gelesen wird. [4] Hier werden Optionen gesetzt, die für einen bestimmten Verbindungspartner gedacht sind, zum Beispiel einen Internet-Service-Provider (ISP).

Nehmen wir an, man pendele zwischen Amsterdam und Den Haag. In jeder Stadt ist der Internetzugang über die beiden ISP-Anbieter Planet und KPN möglich. Zunächst wird ein eine Optionen-Datei für jeden ISP angelegt.

```
# cat > /etc/ppp/peers/KPN <<EOF
remotename KPN
noauth
user kpn
noipdefault
ipparam KPN
EOF
# cat > /etc/ppp/peers/Planet <<EOF
remotename Planet
auth
user user3579@planet.nl
noipdefault
mru 1000
mtu 1000
ipparam Planet
EOF
```

[4] Diese Datei wird durch die Option `call` eingebunden.

Diese Dateien setzen die Optionen, die sich zwischen den beiden ISPs unterscheiden. Optionen, die für beide ISPs gleich sind, können in `/etc/ppp/options` oder in einer der Dateien für die Schnittstellenkonfiguration abgelegt werden.

Nun werden Dateien für jeden ISP angelegt. In unserem Beispiel liegt der einzige Unterschied zwischen den beiden Anbietern in den verschiedenen Chat-Skripten. (Das Chatscript unterscheiden sich, weil die lokale Telefonnummer für den Internet-Zugriff verschiedenist.)

```
# cat > /etc/ppp/peers/KPN-Amsterdam <<EOF
connect "/usr/sbin/chat -v -f /etc/chatscripts/KPN-Amsterdam"
file /etc/ppp/peers/KPN
EOF

# cat > /etc/ppp/peers/KPN-DenHaag <<EOF
connect "/usr/sbin/chat -v -f /etc/chatscripts/KPN-DenHaag"
file /etc/ppp/peers/KPN
EOF

# cat > /etc/ppp/peers/Planet-Amsterdam <<EOF
connect "/usr/sbin/chat -v -f /etc/chatscripts/Planet-Amsterdam"
file /etc/ppp/peers/Planet
EOF

# cat > /etc/ppp/peers/Planet-DenHaag <<EOF
connect "/usr/sbin/chat -v -f /etc/chatscripts/Planet-DenHaag"
file /etc/ppp/peers/Planet
EOF
```

Der Aufruf von `file` fügt eine der oben angegebenen Optionen-Dateien ein. Die `connect`-Anweisung übergibt den Befehl, den `pppd` benutzt, um die Verbindung aufzubauen. Normalerweise wird hierfür `chat` verwendet, wobei das chat-Skript an den ISP angepaßt werden muss. Hier sind die chat-Skripte für Den Haag; das chat-Skript für Amsterdam ist bis auf die Telefonnummer gleich.

```
# cat > /etc/chatscripts/KPN-DenHaag <<EOF
ABORT BUSY
ABORT 'NO CARRIER'
ABORT VOICE
ABORT 'NO DIALTONE'
ABORT 'NO DIAL TONE'
ABORT 'NO ANSWER'
ABORT ERROR
```

```
OK-AT-OK ATDT 0676012321
CONNECT \d\c
EOF
# cat > /etc/chatscripts/Planet-DenHaag <<EOF
ABORT BUSY
ABORT 'NO CARRIER'
ABORT VOICE
ABORT 'NO DIALTONE'
ABORT 'NO DIAL TONE'
ABORT 'NO ANSWER'
ABORT ERROR
OK-AT-OK ATDT 0676002505
CONNECT \d\c
EOF
```

Um sich mit den ISPs verbinden zu können, benötigt man einen Benutzernamen und das zugehörige Password, die `pppd` der Gegenstelle auf Anfrage zuschickt. Diese Informationen werden entweder in `/etc/ppp/pap-secrets` (falls das PAP-Protokolk benutzt wird) oder in `/etc/ppp/chap-secrets` (falls das CHAP-Protokol benutzt wird) gespeichert. Obwohl CHAP sicherer ist, ist PAP noch weit verbreitet. Da diese Dateien geheime Daten enthalten, sollten die Gruppen und die Welt keine Schreib- und Leserechte besitzen. Das FOrmat dieser Dateien wird in `pppd(8)` erklärt. Das Geheimnis "secret" (drittes Feld) wird durch Suche nach dem Benutzernamen (erstes Feld) oder dem Servernamen (zweites Feld) gefunden. Wenn eine Verbindung zu einem ISP aufgenommen wird, ist der Servername in der Regel nciht bekannt, weswegen der Benutzername verwendet wird; dies geschieht in den `user`Zeilen in `peers/KPN` und `peers/Planet` (siehe oben).

```
# client name          server name     secret
kpn                    *               kpn
user3579@planet.nl     *               myfavoritepet
```

Siehe `/usr/share/doc/ppp/README.Debian.gz` für weitere Informationen.

Konfiguration von `pppd` mit `pppconfig`

`pppd` ist mit `pppconfig` einfach zu konfigurieren; es befindet sich im gleichnamigen Paket. `pppconfig` fragt den Benutzer menügesteuert nach den wichtigen Daten und richtet die nötigen Dateien ein.

Konfiguration der PPP-Schnittstelle mit `wvdial`

Ein anderer Ansatz zur Benutzung des `pppd` liegt in `wvdial`, das im sich im Paket `wvdial` findet. Statt `pppd` mit `chat` aufzurufen, um zu wählen und die Verbindung auszuhandeln, kann wvdial dies tun und daraufhin selbständig `pppd` starten. Allein mit der Telefonnummer, dem Benutzernamen und dem Passwort kann `wvdial` in den meisten Fällen erfolgreich die Verbindung aufbauen.

10.3 Namensgebung

10.3.1 Hostname

Der Kernel verwaltet einen Namen des Systems, den **hostname**. Das Initialisierungs-Skript `/etc/init.d/hostname.sh` setzt den System-Hostnamen, der sich in der Datei `/etc /hostname` befindet, während des Bootens mit dem Befehl `hostname`. Diese Datei solle **nur** den System-Hostnamen beinhalten, keinen voll qualifizierten Domänennamen (FQDN).

Um den derzeitigen Hostnamen auszugeben, wird das Kommando `hostname` ohne Argumente aufgerufen.

10.3.2 Mailname

Der **mailname** eines Rechners ist der Name, den er im Zusammenhang mit E-Mail-Zustellung verwendet. Die Datein `/etc/mailname` enthält diesen Namen, der von einem Zeilenwechsel (newline) gefolgt sein muss. Der Mailname ist normalerweise ein voll qualifizierter Domänenname, der sich auch zu einer der IP-Adressen des Rechners auflösen lässt. Siehe `mailname(5)`.

Was der Empfänger von E-Mail in der `Von:`-Zeile der E-Mails von einem Debian-Rechner sieht, hängt von der Konfiguration der E-Mail-Programme (MUA) und dem Mail-Transfer-System (MTA) ab. Angenommen ein lokaler Benutzer namens *`foo`* senset eine E-Mail von einem Rechner mit dem Mailnamen *`meinrechner.dom`*. Die `Von:` Kopfzeile der ausgehenden E-Mail wird folgendermaßen aussehen:

- `Von:` *`foo@meinrechner.dom`*" Falls der MUA keine `Von:` Kopfzeile gesetzt hat;
- `Von:` *`bar@meinrechner.dom`*" Falls der MUA "`Von:` *`bar`*" setzt;
- "`From:` *`bar@bogus.dom`*" Falls der MUA "`Von:` *`bar@bogus.dom`*" setzt.

Selbst wen der MUA eine `Von:`-Kopfzeile gesetzt hat, kann der MTA eine Zeile "`Sender:`*`foo@herman.dom`*" hinzufügen, um die wahre Herkunft anzuzeigen.

Da jeder beteiligte MTA die Adresse ändern kann, kann sie beim Empfänger völlig anders aussehen.

10.4 Domain Name Service (DNS)

Rechner werden durch ihren Domänennamen ebenso spezifiziert wie durch ihre IP-Adresse. DNS ist ein Client-Server-System, in dem Namensauflösung von Nameservern angeboten wird, die Domänennamen mit IP-Adressen verknüpft (neben weiteren Eigenschaften der Rechner). Die GNU C-Bibliothek `resolver(3)` kann ebenfalls IP-Adressen in Dateien nachschlagen oder den Network Information Service (NIS) konsultieren.

Manche Programme (z.B. GNOME) erwarten, dass der kanonische voll qualifizierte Domainname (FQDN) eines Hosts zu einer IP-Adresse aufgelöst werden kann. Dies ist unsauber, da der System-Hostname und der Domänenname zwei sehr verschiedene Dinge sind. Um diese Software zu unterstützen ist es nötig dafür zu sorgen, dass der System-Hostname aufgelöst werden kann. Meist geschieht dies, indem eine Zeile in der Datei hinzugefügt wird, die einige IP-Adressen und den System-Hostnamen enthält. Hat das System eine permanente IP-Adresse, dann wird diese dort eingetragen, in allen anderen Fällen wird kann Adresse 127.0.1.1 verwendet werden.

```
127.0.0.1       localhost
127.0.1.1       uranus
```

Ob der System-Hostname zu einer IP-Adresse mit einem vollen qualifizierten Domänennamen aufgelöst werden kann, wird der Befehl `hostname --fqdn` benutzt.

10.4.1 Resolver

Die Aufgabe des Resolvers (Auflösers) ist die Umsetzung von Domänennamen in IP-Adressen. Der meistgenutzte Resolver sind die Resolver-Funktionen (`resolver(3)`) in der GNU C-Bibliothek. Ein anderer Resolver ist FireDNS, der sich im Paket `libfiredns` befindet. Es gibt noch andere.

Die Konfigurationsdatei `/etc/nsswitch.conf` enthält eine `hosts`-Zeile, welche bestimmt auf welche Art und Weise der GNU-libC-Resolver Namen auflöst. Diese Zeile listet die Services auf, die zur Namensauflösung verwendet werden: z.B., `dns`, `files`, `nis`, `nisplus`. Siehe `nsswitch.conf(5)`. Wenn dort der Service `files` genutzt wird, bestimmt die Konfigurationsdatei `/etc/hosts` das Verhalten des Resolvers. Siehe `hosts(5)`.

Alle genannten Dateien sind statisch und können mit einem beliebigen Editor verändert werden.

Wird der Service `dns` verwendet, bestimmt die Konfigurationsdatei `/etc/resolv.conf` das Verhalten des Resolvers. Siehe `resolv.conf(5)`- Eine der wichtigen Funktionen von `resolv.conf` ist die Bereitstellung der Adressen der Nameserver, die zur Namensauflösung kontaktiert werden. Diese Liste hängt vom Netzwerk ab und kann sich zur Laufzeit ändern. Programme wie `pppd` oder `dhclient` können `resolv.conf` manipulieren und Zeilen hinzufügen oder entfernen, aber dies funktioniert nicht immer zuverlässig und konfliktfrei, Das Paket `resolvconf` löst dieses Problem durch Schaffung einer standardisierten Methode zur Aktualisierung dieser Datei. Siehe 'Verwaltung von Nameserver-Informationen – `resolvconf`' auf der nächsten Seite.

10.4.2 Verwaltung von Nameserver-Informationen – `resolvconf`

Das Paket `resolvconf` schafft einen Rahmen für die dynamische Verwaltung von Informationen über erreichbare Nameserver. Es löst das alte Problem der Organisation dynamischer Liste von Nameservern für den Resolver oder DNS-Caches. Resolvconf ist ein Vermittler zwischen den Programmen, die Netzwerkschnittstellen steuern oder Nameserver-Informationen bereitstellen und Programmen, die Nameserver-Dienste erfragen.

`resolvconf` funktioniert ohne manuelle Konfiguration. Das Paket ist jedoch relativ neu und benötigt vielleicht Anpassungen, bis es richtig läuft. Falls Pakete angepasst wurden, die ihrerseits `/etc/resolv.conf` verändern, so müssen diese Anpassungen allerdings rückgängig gemacht werden. Siehe `/usr/share/doc/resolvconf/README.gz` für weiter Details.

10.4.3 Zwischenspeicherung von Hostnamen – `nscd`, `dnsmasq`, `pdnsd`, `bind9`

Wenn der Nameserver langsam ist, kann `nscd` verwendet werden, um die Ergebnisse der Abfragen des `libc6`-Resolvers zwischenzuspeichern (Caching).

Wenn auch die Abfragen anderer Rechner im lokalen Netzwerk zwischengespeichert werden sollen, kann ein besonderer Nameserver (Caching Forwarding Nameserver) wie `dnsmasq` oder `pdnsd` genutzt werden.

Auch `named` aus dem Paket `bind9` kann als Caching Forwarding Nameserver eingesetzt werden. Dies ist ein umfangreiches Programm, dass nur verwendet werden sollte, wenn dessen weitreichenden Möglichkeiten auch tatsächlich genutzt werden, ansonsten sind die vorher genannten Pakete völlig ausreichend.

Alle diese Pakete arbeiten gut mit `resolvconf` zusammen.

10.4.4 Bereitstellung eines Domain Name Service - `bind`

Falls offizieller Name Service in einer Domäne angeboten werden soll, sollte ein ausgewachsener Nameserver wie `named` aus dem `bind9`-Paket verwendet werden.

Zusammen mit `bind9` sollte auch `dnsutils` installiert werden. Auch folgende Pakete enthalten nützliche Werkzeuge: `bind9-host`; `dns-browse`; `dnscvsutil`; `nslint`. Die Dokumentation befindet sich in einem eigenen Paket: `bind9-doc`. Die Entwicklerpakete befinden sich in diesen Paketen: `libbind-dev`; `libnet-dns-perl`. Falls Schnittstellen mit DHCP konfiguriert werden sollen, sind diese Pakete nützlich: `dhcp-dns`.

Die grundlegende Einrichtung geschieht bei Installation von `bind9` oder dem Aufruf von `dpkg-reconfigure` mit dem Paketnamen als Argument. Die Konfiguration wird in `named.conf` vorgenommen. In Debian befindet sich diese Datei in `/etc/bind/` und wird hauptsächlich genutzt, um die elementaren DSN-Zonen festzulegen; sie fügt mit `include` zwei weitere Dateien ein: `named.conf.local` definiert lokale Zonen, während `named.conf.options` Optionen festlegt. (Letztere wird von `resolvconf` verarbeitet, um daraus /var/run/bind/named.options zu erzeugen, welche sich vom Original darin unterscheidet, dass die forwarders-Spezifikation eine Liste der derzeit erreichbaren nicht-lokalen Nameserver beinhaltet. Um dieses dann zu nutzen, muss die include-Zeile in named.conf angepasst werden, so dass /var/run/bind/named.options eingebunden wird. Siehe 'Verwaltung von Nameserver-Informationen – resolvconf' auf der vorherigen Seite

Datenbankdateien der Form named.conf* ohne vollen Pfadnamen werden in /var/cache /bind/ gespeichert. An dieser Stelle sollten die Dateien abgelegt werden, die named erzeugt. zum Beispiel Datenbakdateien für Zonen, für die der Dämon Sekundant istXXX. Statische Datenbankdateien in /etc/bind/ müssen in named.conf mit vollem Pfadnamen angegeben werden. Siehe /usr/share/doc/bind9/README.Debian.gz für Details.

10.5 Konfiguration der Netzwerkschnittstelle mit DHCP

Die Low-Level-Konfiguration der Netzwerkschnittstelle kann mit dem Dynamic Configuration Protocol (DHCP) automatisiert werden. Auf diese Weise können Firewalls und Router oder Breitband-ISP ihre IP-Adressen und andere Parameter verteilen.

Dazu müssen folgende Pakete installiert werden:
- dhcp3-client (version 3, Internet Software Consortium)
- dhcpcd (Yoichi Hariguchi and Sergei Viznyuk)
- pump (Red Hat)

pump ist einfach und weit verbreitet. dhcp3-client ist komplexer, aber dafür umfassender konfigurierbar. [5]

10.6 High-Level-Netzworkkonfiguration in Debian

10.6.1 High-Level-Netzwerkkonfiguration mit ifupdown

Um die Netzwerkkonfiguration in Debian zu vereinfachen existieren die High-Level-Konfigurationswerkzeugeifup und ifdown, sowie die /etc/network/interfaces-Datei. [6] Falls das Paket ifupdown zur Netzwerkkonfiguration genutzt wird, dann sollten die Low-Level-Befehle **nicht** benutzt werden. Auch andere High-Level-Tools, wie whereami, divine, intuitively, etc., die wiederum Low-Level-Tools aufrufen, sollten gemieden werden. Das Paket ifupdown wurde entworfen, um allein für die Netzwerkschnittstellen zuständig zu sein.

[5] Seit April 2004 gibt es das Paket dhcp-client-Paket. Dieses Paket enthält die Version 2 des ISC DHCP-Clients. Aktuell ist derzeit Version 3 im Paket dhcp3-client. Nach der Veröffentlichung von Sarge soll das Paket dhcp3-client in dhcp-client umbenannt werden. Es sollte sichergestellt werden, dass keine experimentellen Versionen des Pakets dhcp-client installiert sind. ifupdown funktioniert damit nicht.

[6] Das Format der /etc/network/interfaces-Datei in der aktuellen Version des ifupdown unterscheidet sich vom Format der Versionen in Potato. Das ifupdown post-installation-Skript wird diese Datei falls nötig automatisch anpassen. Die konvertierte Datei sollte jedoch überprüft werden.

Um dies zu unterstützen, muss Folgendes getan werden:

```
# ifdown eth0
# editor /etc/network/interfaces # sollte angepasst werden
# ifup eth0
```

Weitere Informationen finden sich in `interfaces(5)`, `/usr/share/doc/ifupdown/examples/network-interfaces.gz` und `ifup(8)`.

Konfiguration einer Schnittstelle mit einer statischen IP-Adresse

Nehmen wir an eine Ethernet-Schnittstelle soll die feste IP-Adresse `192.168.0.111` zugewiesen bekommen. Diese Adresse beginnt mit `192.168.0`, weswegen es sich in einem LAN befinden muss. Das Gateway dieses LANs habe die Adresse `192.168.0.1`. Der Datei `/etc/network/interfaces` muss also diese Zeile hinzugefügt werden:

```
iface eth0 inet static
        address 192.168.0.111
        netmask 255.255.255.0
        gateway 192.168.0.1
```

In "up"- und "down"-Zeilen können andere Aspekte dieser Schnittstelle eingestellt oder Aktionen definiert werden, die beim Auf- oder Abschalten der Schnittstelle durchgeführt werden.

```
iface eth0 inet static
        address 192.168.0.111
        netmask 255.255.255.0
        gateway 192.168.0.1
        up route add -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.0.2 dev $IFACE
        down route del -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.0.2 dev $IFACE
        up echo Interface $IFACE going up | /usr/bin/logger -t ifup
        down echo Interface $IFACE Going down | /usr/bin/logger -t ifdown
```

Desweiteren können auszuführende Skripte in `/etc/network/if-up.d` und `/etc/network/if-down.d` plaziert werden. In diesen Skripten können auch weitergehende Optionen programmiert werden. Siehe `interfaces(5)` für Details. Zum Beispiel beinhaltet das Paket `resolvconf` Skripte, mit denen DNS-Informationen zu `/etc/resolv.conf` hinzugefügt werden können, während die Schnittstelle aktiv ist.

```
iface eth0 inet static
        address 192.168.0.111
        netmask 255.255.255.0
        gateway 192.168.0.1
        dns-search somedomain.org
        dns-nameservers 195.238.2.21 195.238.2.22
```

Der Parameter `somedomain.org` der `dns-search`-Option entspricht dem Parameterder `search` -Option in `resolv.conf(5)`. Die Parameter `195.238.2.21` und `195.238.2.22` der `dns-nameservers`-Parameter entspricht den `nameserver`-Parametern. Andere Optionen sind `dns-domain` und `dns-sortlist`. Siehe 'Verwaltung von Nameserver-Informationen – `resolvconf`' auf Seite 174.

Konfiguration einer Schnittstelle mit DHCP

Um eine Schnittstelle mit DHCP zu konfigurieren, muss in `/etc/network/interfaces` eine Zeile wie die folgende hinzugefügt werden:

```
iface eth0 inet dhcp
```

Damit dies funktioniert, muss einer der DHCP-Clients installiert sein (siehe 'Konfiguration der Netzwerkschnittstelle mit DHCP' auf Seite 175).

konfiguration einer Wi-Fi-Schnittstelle

Das `wireless-tools`-Paket beinhaltet ein Skript `/etc/network/if-pre-up.d/wireless-tools` welches Wi-Fi (802.11a/b/g) bevor die Schnittstelle aufgeschaltet wird. Die Konfiguration erledigt das Programm `iwconfig`; siehe `iwconfig(8)`. Für jeden `iwconfig`-Parameter kann eine Option in `/etc/network/interfaces` wie jene schon dort vorhandenen hinzugefügt werden, wobei die drahtlosen Varianten durch "wireless-" eingeleitet werden. Um zum Beispiel die ESSID von `eth0` auf `myessid` und den Schlüssel auf `123456789e` zu setzen, bevor `eth0` mit DHCP aufgeschaltet wird, muss `/etc/network/interfaces` folgendermaßen erweitert werden.

```
iface eth0 inet dhcp
        wireless-essid myessid
        wireless-key 123456789e
```

Diese Methode zum Setzen der ESSID sollte nicht verwendet werden, wenn `waproamd` diese Schnittstelle überwacht. Wenn `ifup` gestartet wird, hat `waproamd` die ESSID und den Schlüssel schon gesetzt. Siehe 'Schalten der Netzwerkkonfiguration - `waproamd`' auf Seite 186.

Konfiguration einer PPP-Schnittstelle

Die Programme `ifup` und `ifdown` verwenden `pon` und `poff` um PPP-Schnittstellen zu verwalten; siehe 'Konfiguration der PPP-Schnittstelle' auf Seite 168.

Angenommen eine PPP-Verbindung zu `myisp` soll aufgebaut werden. Die Datei `/etc/network/interfaces` wird die folgende Zeile hinzugefügt:

```
iface ppp0 inet ppp
        provider myisp
```

Diese Zeile sorgt dafür, dass `ifup ppp0` tatsächlich

```
pon myisp
```

aufruft. Leider ist es derzeit nicht möglich weitere Optionen an `pppd` durch diese Zeile in `/etc/network/interfaces` weiterzureichen. [7]

Mit dem Paket `ifupdown` kann keine Hilfskonfiguration der PPP-Schnittstelle vorgenommen werden. Da `pon` beendet wird, bevor `pppd` die Verbindung etabliert hat, startet `ifup` die `up`-Skripte schon bevor die PPP-Schnittstelle bereit ist. Bis zu diesem Fehler[8] ist repariert, muss die Hilfskonfiguration in `/etc/ppp/ip-up` oder `/etc/ppp/ip-up.d` erledigt werden.

Konfiguration der PPPoE-Schnittstelle

Manche Breitband-Internet-Anbieter (ISP) verwenden PPP, um Verbindungen herzustellen, auch wenn die Rechner über Ethernet oder ATM an das Netzwerk angeschlossen sind. Dies wird durch PPP über Ethernet (PPPoE) erreicht, eine Technik, die den PPP-Datenstrom in Ethernet-Frames kapselt. Angenommen, der ISP heiße *`meinisp`* Zunächst werden PPP und PPPoE für den Zugang über *`meinisp`* konfiguriert. Das Einfachste ist es, das Paket `pppoeconf` zu installieren und dann auf der Konsole `pppoeconf` zu starten. Dann muss in `/etc/network/interfaces` folgende Zeile hinzugefügt werden:

```
iface eth0 inet ppp
        provider myisp
```

Manchmal gibt es Probleme mit der Maximum Transfer Unit (MTU), wenn PPPoE über Digital Subscriber Line (DSL) verwendet wird. Siehe DSL-HOWTO (`http://www.tldp.org/HOWTO/DSL-HOWTO/`) für Details.

Wenn ein Breitband-Modem einen Router beinhaltet, dann wird das Modem/der Router die PPPoE-Verbindung verhandeln und im LAN wie ein gewöhnliches Ethernet-Gateway zum Internet erscheinen.

[7] Siehe dazu den Fehlerbericht #196877 (`http://bugs.debian.org/196877`).

[8] Siehe Fehlerbeschreibung #127786 (`http://bugs.debian.org/127786`).

Konfiguration verschiedener Ethernet-Schnittstellen für ein Gateway

Angenommen `eth0` wird mit einer von DHCP vermittelten IP-Adresse an das Internet angeschlossen und `eth1` ist mit der statischen IP-Adresse `192.168.1.1` mit einem LAN verbunden. Die Datei `/etc/network/interfaces` wird um folgende Zeilen erweitert:

```
iface eth0 inet dhcp

iface eth1 inet static
        address 192.168.1.1
        netmask 255.255.255.0
```

Wenn an diesem Rechner NAT (siehe 'Aufbau eines Gateway-Routers' auf Seite 190) aktiviert wird, kann dieser Rechner seine Internet-Verbundung mit allen anderen Rechnern des LAN teilen.

Konfiguration virtueller Schnittstellen

Mittels virtueller Schnittstellen kann eine einzelne Ethernet-Karte so eingestellt werden, dass sie den Zugang zu verschiedenen IP-Subnetzwerken bietet. Angenommen ein Rechner ist an das Netzwerk 192.168.0.x/24 angeschlossen. Dieser Rechner soll sich mit der installierten Ethernet-Karte und einer vorhandenen IP-Adresse, die per DHCP bezogen wurde, mit dem Internet verbinden. In diesem Fall muss `/etc/network/interfaces` folgende Zeilen beinhalten:

```
iface eth0 inet static
        address 192.168.0.1
        netmask 255.255.255.0
        network 192.168.0.0
        broadcast 192.168.0.255

iface eth0:0 inet dhcp
```

Die `eth0:0`-Schnittstelle ist in diesem Fall die virtuelle Schnittstelle. Wenn sie angesprochen wird, so wird auch die übergeordnete Schnittstelle `eth0` aufgeschaltet.

10.6.2 High-Level-Konfiguration mit logischen Schnittstellen von `ifupdown`

Im Folgenden ist es wichtig den Unterschied zwischen **physikalischen Schnittstellen** und **logischen Schnittstellen** zu treffen. [9] Eine **physikalische** Schnittstelle wird "die Schnittstelle"

[9] Die hier verwendete Terminologie ist der `ifupdown`-Dokumentation entnommen.

genannt und vom Kernel `eth0`, `eth1`, `ppp0`, usw. genannt. Eine **logische** Schnittstelle ist ein Satz von Werten der den variablen Parametern einer physikalischen Schnittstelle zugeordnet werden kann. Dies ist kann veranschaulicht werden, indem man statt zu sagen "dies ist die logische Schnittstelle *X*", klarer formuliert: "diese ist die Schnittstelle mit dem Schnittstellen-Profil *X*.

Die `iface`-Definitionen in `/etc/network/interfaces` sind tatsächlich Definitionen logischer und nicht physikalischer Schnittstellen. [10] Wenn die Schnittstellen nicht erneut konfiguriert werden sollen, dass kann diese Tatsache einfach ignoriert werden, da der physikalischen Schnittstelle *foo* per VOreinstellung die logische Schnittstelle *foo* zugeordnet wird.

Angenommen, der Computer sei ein Laptop, der unterwegs verwendet wird. Wenn er mit einem Firmennetzwerk oder am heimischen LAN angeschlossen wird, muss `eth0` entsprechend angepasst konfiguriert werden.

Zunächst werden dazu zwei logische Schnittstellen `heim` und `arbeit` angelegt (statt wie vorher `eth0`), die die Schnittstellen-Konfigurationen für die beiden Arbeitsplätze beinhalten sollen.

```
iface home inet static
        address 192.168.0.123
        netmask 255.255.255.0
        gateway 192.168.0.1

iface work inet static
        address 81.201.3.123
        netmask 255.255.0.0
        gateway 81.201.1.1
```

Dann kann die physikalische Schnittstelle `eth0` von zu Hause mit der folgenden Kommandozeile aufgeschaltet werden:

```
# ifup eth0=home
```

Um `eth0` auf das Firmennetzwerk umzuschalten, reicht:

```
#ifdown eth0
# ifup eth0=work
```

Wenn die Datei `interfaces` wie oben geschildert aussieht, ist es nicht mehr möglich, die Schnittstelle `eth0` durch ein einfaches `ifup eth0` zu starten. Denn `ifup` verwendet den Namen der physikalische Schnittstelle als Namen für die logische Schnittstelle und für `eth0` ist keine logische Schnittstelle definiert.

[10] Die Schnittstellen mit `auto`-Zeilen müssen physikalische Schnittstellen sein, anstatt logische.

10.6.3 Automatische Netzwerkkonfiguration mit `ifupdown`

Schnittstellen-Namen können andere Namen abgebildet werden (”mapping“), sobald `ifup` läuft. Auf welche Art und Weise diese Abbildung geschieht hängt von den Umständen ab. `ifup` kann eine physikalische Schnittstelle als eine bestimmte logische Schnittstelle aus einer Anzahl vorgefertiger Alternativen auswählen.

Die Abbildung logischer Schnittstellen läuft folgendermaßen ab:

- Wenn kein logisches Laufwerk auf der `ifup`-Kommandozeile übergeben wurde, dann wird der Name der physikalischen Schnittstelle auch für die erste logische Schnittstelle verwendet.
- Wenn der Name der logischen Schnittstelle einem glob-Muster in einer `mapping`-Zeile entspricht, dann wird diese map verwendet, um einen neuen logischen Schnittstellenname zu erzeugen. Dies gilt für jede einzelne Umbelegung.
- Wenn der Name der letzten logischen Schnittstelle als Bezeichnung in `/etc/network/interfaces` auftritt, dann wird die physikalische Schnittstelle als diese logische Schnittstelle aufgeschaltet. Ansonsten wird `ifup` folgende Nachricht ausgeben und enden: ”Ignoring unknown interface“ (unbekannte Schnittstelle wird ignoriert)

Die Syntax der `mapping`-Zeile ist:

```
mapping glob-pattern
       script script-name
       [map script input]
```

Das Skript, welches in der `mapping`-Zeile genannt wird, muss immer den Namen der **physikalischen** Schnittstelle als Argument bekommen und über die Standardeingabe die Inhalte aller folgenden ”map“-Zeilen (ohne das Wort ”map“ selbst. Das Skript schreibt das Ergebnis der Umbelegung auf die Standardausgabe und wird dann beendet.

Zum Beispiel wird der folgende ”mapping“-Absatz dafür sorgen, dass `ifup` die Schnittstelle `eth0` als die logische Schnittstelle `aufschalten wird.`

```
mapping eth0
       script /usr/local/sbin/echo-home
```

wobei `/usr/local/sbin/echo-home` so aussieht:

```
#!/bin/sh
echo home
```

Da die Umbelegung mit einem Skript erledigt wird, kann die Wahl der logischen Schnittstelle auch anhand irgendeines Tests erfolgen. In ‘Wahl logischer Schnittstellen mit `guessnet`’ auf der nächsten Seite finden sich Beispiele.

Wahl logischer Schnittstellen mit `guessnet`

Das Paket `guessnet` muss installiert werden und dann folgende Zeilen in `/etc/network/interfaces` eingefügt werden:

```
mapping eth0
       script guessnet-ifupdown
       map home
       map work
```

Wenn nun `ifup eth0` ausgeführt wird, wird `guessnet` prüfen, ob `eth0` als `home` oder `work` aufgeschaltet werden soll. Um diese Entscheidung zu treffen, wird in der Definition der logischen Schnittstelle gespeicherte Information ausgewertet.

10.6.4 Automatische Netzwerkkonfiguration mit `laptop-net`

Das Paket `laptop-net` wählt einen anderen Ansatz zur automagischen Netzwerkkonfiguration. Laptop-net verwendet nicht die logischen Schnittstellen des Pakets `ifupdown` sondern hat sein eigenes System von Konfigurations "Schemata und Systemprofilen". Laptop-net verwendet aber dennoch `ifup` und `ifdown` zur Konfiguration der physikalischen Schnittstellen. Weitere Informationen finden sich in der guten Dokumentation im Paket `laptop-net-doc`.

10.6.5 Automatische Netzwerkkonfiguration mit `network-manager`

Das Programm `network-manager` wird derzeit von Fedora-Entwicklern gepflegt und ist auch für Ubuntu erhältlich. Es könnte auch irgendwann in Debian auftauchen und ifupdown und zugehörige Werkzeuge überflüssig machen.

10.7 Umgang mit inkonsistenten Schnittstellennamen seitens des Kernels

Die Namen `eth0`, `eth1`, etc. werden vom Kernel in der Reihenfolge der Erstellung vergeben. Während die Anschlüsse, die zur Boot-Zeit angeschlossen daher immer die gleichen Namen erhalten, gilt dies nicht für Geräte, die im Betrieb angeschlossen werden ("hot-plugging"). Diese können in jeder beliebigen Reihenfolge erkannt werden und erhalten so unterschiedliche Namen.

Daher funktioniert die Zuordnung mittels logischer Schnittstellen in `/etc/network/interfaces` mit den Namen `eth0`, `eth1`, etc. und der Standard-Umbelegung bei Systemen mit Netzwerk-hot-plugging nicht zuverlässig. Instead you must give distinct names to the logical interfaces and use one of the following two methods to restrict which logical interfaces can be assigned to which adapters.

Eine Möglichkeit ist das Werkzeug `nameif` (im Paket `net-tools`) oder auch das flexiblere `ifrename` (im Paket `ifrename`), das den Kernel veranlasst Namen anhand von Eigenschaften der Anschlüsse zu vergeben. Sobald das Namensschema aktiviert ist, kann aus dem Namen der physikalischen Schnittstelle gefolgert werden, welcher Anschluss tatsächlich verwendet wird.

Eine andere Möglichkeit ist der Umbelegungsmechanismus von `ifup`, der auf ähnliche Weise für eine physikalische Schnittstelle eine logische Schnittstelle wählt anhand der Eigenschaften der Netzwerkkarte.

Angenommen zwei Netzwerkkarten seien für die Netzwerke `net1` und `net2` zuständig. Das Verzeichnis `/usr/share/doc/ifupdown/examples/` enthält ein Mapping-Skript, dass verwendet werden kann, um eine logische Schnittstelle basierend auf der Media Access Controller-Adresse (MAC-Adresse) der jeweiligen Netzwerkkarte auszuwählen. Zunächst wird das Skript in einem angemessenen Verzeichnis installiert.

```
install -m770 /usr/share/doc/ifupdown/examples/get-mac-address.sh # \
       /usr/local/sbin/
```

Daraufhin werden `/etc/network/interfaces` folgende Zeilen hinzugefügt:

```
mapping eth0
       script /usr/local/sbin/get-mac-address.sh
       map 02:23:45:3C:45:3C net1
       map 00:A3:03:63:26:93 net2
```

Siehe 'Multi-stage mapping' auf Seite 187 für komplexere Beispiele.

Meist wird bei diesen Methoden zur Identifikation der Netzwerkkarte die MAC-Adresse verwendet.

10.8 Umschalten der Netzwerkkonfiguration

Es wurde gezeigt, die Schnittstellen konfiguriert und re-konfiguriert werden. Dies muss dann und wann geschehen.

Normalerweise wird das Netzwerk während des Bootens konfiguriert, im Init-Skript `/etc/rcS.d/S40networking`, und dies wird selten geändert. Services, die vom Netzwerk abhängig sind, müssen nach dem genannten Init-Skript gestartet werden. Bei Neustart oder Abschalten des Systems müssen die Init-Skripte in umgekehrter Reihenfolge ausgeführt werden.

Es liegt allerdings im Trend Hardware dynamisch zu konfigurieren. Zuerst wurden in GNU/Linux hot-plug-fähige PCMCIA-Karten unterstützt; später wurde der `hotplug`-Mechanismus hinzugefügt, so dass viel mehr verschiedene Peripherie während des Betriebs ein- und ausgeklinkt werden kann. Dies gilt auch für Netzwerk-Hardware. Note that services that depend on hardware that is hot swapped must only be started after the hardware is inserted and must be stopped when the hardware is removed. Das bedeutet, dass solche Services der Kontrolle des System-V-Init-Systems entzogen werden müssen, damit sie von `ifupdown` gesteuert werden können.

Angenommen, der vom Init-Skript `/etc/init.d/foo` gesteuerte Service `foo` sei von einer dynamisch rekonfigurierten Netzwerkschnittstelle `eth0` ab.

- Zuerst muss `foo` der Kontrolle des Init-Systems entzogen werden. Wenn das Paket `sysv-rc`-Init-System installiert wurde, geschieht dies folgendermaßen: [11]

```
rm /etc/rc[2345].d/S??foo
```

- Dann wird `foo` dem `ifupdown`-Paket unterstellt, indem `up`- und `down`-Optionen zu der `eth0`-Zeile in `/etc/network/interfaces` zugefügt werden, die das `foo`-Init-Skript aufruft.

```
iface eth0 inet dhcp
       up /etc/init.d/foo start
       down /etc/init.d/foo stop
```

10.8.1 Umschalten der Netzwerkkonfiguration zur Boot-Zeit

Während des Bootens startet `/etc/rcS.d/S40networking` den Befehl `ifup -a`. Dieses schaltet alle physikalischen Schnittstellen auf, die in `auto`-Zeilen in `/etc/network/interfaces` aufgelistet sind.

Es wird empfohlen die Netzwerkkonfiguration mit dynamischen Methoden vorzunehmen. Sobald die Mechanismen zur Unterstützung sich dynamisch ändernder Hardware funktioniert, kann auch statische Hardware wie dynamische behandelt werden. Das Booten ist dann nichts anderes als ein weiteres Anschließen mit Hot-plug. (Siehe 'Schalten der Netzwerkkonfiguraion - `hotplug`' auf dieser Seite.)

Die loopback-Schnittstelle `lo` soll jedoch auf jeden Fall beim Booten aktiviert werden. Daher beinhaltet `/etc/network/interfaces` die folgenden Zeilen:

```
auto lo
iface lo inet loopback
```

Es können die Namen weiterer physikalischer Schnittstellen in `auto`-Zeilen angelegt werden, die ebenfalls beim Booten aufgeschaltet werden. Es sollten **niemals** PCMCIA-Schnittstellen mit `auto` eingetragen werden. Der PCMCIA `cardmgr` wird später in der Bootsequenz gestartet, also später als `/etc/rcS.d/S40networking`XXX.

10.8.2 Schalten der Netzwerkkonfiguraion - `hotplug`

Hot-Plug-Unterstützung bietet das Paket `hotplug`.

[11](Die "stop"-Links `/etc/rc?.d/K??foo`bleiben erhalten.) Siehe 'Runlevel' auf Seite 23 für weitere Informationen.

Netzwerk-Hardware kann zu folgenden Gelegenheiten eingebunden werden: zur Bootzeit; nachdem eine Netzwerkkarte (z.B. PCMCIA) eingesteckt wurde oder wenn ein Hilfswerkzeug wie `discover` gestartet wird und die nötigen Treiber geladen werden.

Wenn der Kernel neue Hardware erkennt, wird der Treiber für die Hardware geladen und dann startet `hotplug` um diese Hardware zu konfigurieren. Wenn die Hardware wieder entfernt wird, startet `hotplug` erneut, allerdings mit anderen Einstellungen der Umgebungsvariablen. In Debian werden die Skripte in `/etc/hotplug/` und `/etc/hotplug.d/` von `hotplug` aufgerufen. Siehe `hotplug(8)` für Details.

Neu eingesteckte Netzwerk-Hardware wird vom Skript `/etc/hotplug/net.agent` konfiguriert. [12] Angenommen eine PCMCIA-Netzwerkkarte wurde eingesteckt und die Schnittstelle `eth0` bereits bereitgestellt. `/etc/hotplug/net.agent` tut Folgendes:[13] :

```
ifup eth0=hotplug
```

Wenn keine logische Schnittstellen-Definition und kein Mapping mit dem Namen `hotplug` zu `/etc/network/interfaces` hinzugefügt wurde, dann tut dieser Befehl gar nichts. Damit dieses Kommando `eth0` konfigurieren kann, muss folgende Zeile in `/etc/network /interfaces` hinzugefügt werden:

```
mapping hotplug
        script echo
```

Wie in 'High-Level-Konfiguration mit logischen Schnittstellen von `ifupdown`' auf Seite 179 erklärt, wird dies das obige Kommando zu Folgendem ummünzen:

```
ifup eth0=eth0
```

Eine solche mapping-Zeile sollte **keinesfalls** eingefügt werden, wenn `ifplugd` oder `waproamd` von Hotplug gestartet werden.

Wenn nur `eth0` und sonst keine anderen Schnittstellen von Hotplug aufgeschaltet werden sollen, dann kann `grep` statt `echo` verwendet werden:

```
mapping hotplug
        script grep
        map eth0
```

Siehe 'Automatische Netzwerkkonfiguration mit `ifupdown`' auf Seite 181 und `/usr/share/doc/hotplug/README.Debian` für weitere Tipps.

[12] Sie kann auch mit hook-Skripten konfiguriert werden, die in `/etc/hotplug.d/net/` hinterlegt sind. Bspw. installieren das `ifplugd`- und das `waproamd`-Paket Skripte an diesem Ort

[13] Seit der Version 0.0.20040329-4 (oder so) hat das Paket `hotplug` auch Modi, in denen das Verhalten vom hier beschriebenen abweicht. Einer dieser Modes heißt "all"; in dem `hotplug` all hot-plug-Schnittstellen aufschaltet. Ein anderer Modus heißt "auto", in diesem wird eine Schnittstelle nur aufschalten, wenn diese mit einer `auto`-Zeile in `/etc/network/interfaces` gelistet ist. In diesen alternativen Modes wird `ifup` aufgerufen mit dem `=hotplug`-Suffix.

10.8.3 Schalten der Netzwerkkonfiguration - `ifplugd`

Der `ifplugd`-Dämon schaltet eine Schnittstelle auf oder ab, wenn die zugehörige Hardware an- oder abgestöpselt wird. Das Programm kann ein aktives Netzwerkkabel an der Ethernet-Schnittstelle oder einen Access-Punkt erkennen, der zu einem Wi-Fi-Schnittstelle gehört (jedoch kann `waproamd` hier wahrscheinlich bessere Dienst leisten). Wenn `ifplugs` erkennt, dass sich der Zustand der Schnittstelle geändert hat, dann wird ein Skript gestartet, dass `ifup` oder `ifdown` aufruft.

10.8.4 Schalten der Netzwerkkonfiguration - `waproamd`

Der `waproamd`-Dämon ähnelt `ifplugs`, nur dass er für Wi-Fi-Karten gedacht ist. Er sucht aktiv nach Access-Punkten, mit denen sich die Wi-FI-Hardware verbinden kann. Wenn eine möglicher Verbindungsaufbau erkannt wird, startet `waproamd` dann `ifup`.

Falls `waproamd` verwendet wird, sollte die Wi-Fi-Karte nur mit `waproamd` konfiguriert werden und nicht mit den `wireless-*`-Optionen in `/etc/network/interfaces`.

10.8.5 Netzwerk-Konfiguration und PCMCIA

Es gibt verschiedene Möglichkeiten zur Konfiguration von PCMCIA-Netzwerkkarten (in 2.4 und 2.6 Kernels).

- Für 32-bit-PCI (CardBus) PCMCIA-Netzwerkkarten:
  - `ifupdown` gesteuert von `hotplug`
    - In Woody and Sarge muss `hotplugs` Kontrolle über `ifupdown` lokal aktiviert werden, indem eine mapping-Zeile zu `/etc/network/interfaces` hinzugefügt wird, siehe 'Schalten der Netzwerkkonfiguraion - `hotplug`' auf Seite 184.
- Für 16-Bit ISA-PCMCIA-Netzwerkkarten:
  - `ifupdown` gesteuert von `hotplug` mit `pcmcia-cs`, um die Module zu laden.
    - **Empfohlen**
    - In Woody and Sarge muss `pcmcia-cs`'s Standard-Verhalten abgeschaltet zur Kontrolle von `ifupdown` abgeschaltet werden, indem die Zeile `exit 0` an den Anfang von `/etc/pcmcia/network` gestellt wird. Zudem muss `hotplugs` KOntrolle über `ifupdown` hergestellt werden, indem eine mapping-Zeile zu `/etc/network/interfaces` hinzugefügt wird, so wie in 'Schalten der Netzwerkkonfiguraion - `hotplug`' auf Seite 184

    beschrieben.
  - `ifupdown` gesteuert von `pcmcia-cs` mit den Voreinstellungen in `/etc/pcmcia/network`
    - **Veraltet**, aber immer noch die Voreinstellung in Woody und Sarge
  - Low-Level-Werkzeuge, die vom `pcmcia-cs` mittels spezieller Kodes in `/etc/pcmicia/network` gesteuert werden.
    - **Veraltet**
    - In Woody und Sarge wird der spezielle Code in der Datei `/etc/pcmcia/network.opts` aktiviert.

16-Bit-Karten sollten seit Kernel 2.4 die Hotplug-Unterstützung für PCMCIA-Karten nutzen. [14]

PCMCIA-Netzwerk-Karten können im Betrieb an- und abgestöpselt werden. Daher sollten alle Services, die auf ein Netzwerk via PCMCIA-Karte zugreifen, so eingerichtet sein, dass sie gestartet werden, wenn die Karte eingesteckt wird und beendet werden, wenn die Karte entfernt wird. Dies wird üblicherweise erreicht, indem der Service gemeinsam mit `ifup` gestartet und mit `ifdown` beendet wird. Manche Benutzer beschränken sich darauf, die Geräte nur bei ausgeschaltetem Rechner an- und abzustecken. Sie stecken die Karte vor dem Booten des Systems und starten die Netzwerk-Services in der Boot-Sequenz. Um zu gewährleisten, dass die Karte vollständig konfiguriert ist, bvor der Service startet, sollte Folgendes getan werden:

- In `/etc/default/pcmcia` soll `CARDMGR_OPTS="-f"` gesetzt werden, damit `cardmgr` im Vordergrund läuft.
- Dann sollte `/etc/rc?.d/S20pcmcia` in `/etc/rc?.d/S12pcmcia` (oder ähnlich) umbenannt werden.

Dies funktioniert nur mit 16-Bit PCMCIA-Karten.

Das Paket `pcmcia-cs` wird benötigt, wenn 16 bit PCMCIA-Karten verwendet werden. Der in diesem Paket enthaltene `cardmgr`-Dämon ist für die Verwaltung der Sockets verantwortlich und für das Laden der Treiber. Der Dämon soll aber keine Netzwerk-Konfiguration via `/etc/pcmcia/network` durchführen.

Damit `cardmgr` richtig funktioniert, muss eventuell `/etc/pcmcia/config.opts` editiert werden, damit die 16-Bit PCMCIA-Karten die richtigen Ressourcen zugewiesen bekommen. Siehe 'PCMCIA' auf Seite 88 und Linux PCMCIA HOWTO (`http://www.tldp.org/HOWTO/PCMCIA-HOWTO.html`) für weitere Informationen.

10.9 Multi-stage mapping

Angenommen die Netzwerkkarten können per Hotplug angeschlossen werden und die automatische Konfiguration ist aktiviert, so wie in 'Schalten der Netzwerkkonfiguraion - `hotplug`' auf Seite 184 beschrieben. Weiterhin angenommen, die logischen Schnittstellen müssten auf "physikalische" Schnittstellen abgebildet werden, je nachdem welche Karte angeschlossen ist (wie beschrieben in 'Umgang mit inkonsistenten Schnittstellennamen seitens des Kernels' auf Seite 182) und mit welchem Netzwerk Verbindung aufgenommen werden soll (wie beschrieben in 'Wahl logischer Schnittstellen mit `guessnet`' auf Seite 182). Dies kann mit multi-stage mapping erreicht werden.

[14] Ältere Ausgaben von Debian verwendeten die `cardmgr`-Skripte `/etc/pcmcia/network` und `/etc/pcmcia/network.opts`, um die PCMCIA-Netzwerkkarten zu konfigurieren. Diese Skripte entstammen einer Ära, in der Linux noch keine allgemeines Hotplugging unterstützte. Manchmal werden die Debian Woody-Skripte noch verwendet, welche einfach `ifup` aufrufen, wenn die Schnittstelle hinzugefügt wird und `ifdown`, wenn die Schnittstelle entfernt wird. WIe bereits bemerkt, sollte nun das `hotplug`-Paket verwendet werden. Andere verwenden noch immer die Low-Level-Befehle, die aktiviert werden, wenn bestimmte Variablen in `/etc/pcmcia/network.opts` auf `y` gesetzt sind. Daraus erwachsen verschiedene Probleme. Es können Race-conditions auftreten; es funktioniert nur mit 16-Bit PCMCIA-Karten; es erledigt, was besser mit `ifupdown` getan würde. Daher ist es veraltet und überflüssig.

Die erste Mapping-Stufe nimmt den `hotplug`-Gruppennamen und gibt den vom Kernel zugewiesenen Schnittstellennamen aus, wenn die Schnittstelle "warm" angeschlossen wird. Die zweite mapping-Stufe erwartet einen vom Kernel zugewiesenen Schnittstellennamen und gibt einen Anschlussnamen aus. In der dritten mapping-Stufe werden Adapternamen auf logische Schnittstellen abgebildet unter Berücksichtigung der Netzwerkumgebung.

```
 # Allow hotplug to bring up interfaces
 mapping hotplug
        script echo
 # Determine whether interface is wired or Wi-Fi
 mapping eth?
        script /usr/local/sbin/get-mac-address.sh
        map 02:23:45:3C:45:3C wired
        map 00:A3:03:63:26:93 wifi
# Detect which wired network is available
 mapping wired
        script guessnet-ifupdown
        map work-wired
        map home
# Detect which Wi-Fi network is available
mapping wifi
        script ifscout
        map starbucks
        map work-wireless

iface work-wired inet static
        ...
```

10.10 Konfiguration der Netzwerk-Dienste

Zur Konfiguration der Netzwerk-Dienste dienen unter anderem:

- Der Internet-*Super-Server* und TCP/IP-Dämon-Wrapper, siehe 'Zugriffskontrolle auf Daemonen' auf Seite 130.
  - `/etc/inetd.conf`
- `ssh`: Für die sichere Shell OpenSSH, siehe 'SSH' auf Seite 148.
  - `/etc/ssh/ssh_config`
  - `/etc/ssh/sshd_config`
- `exim`: Ein Mail-Transport-Agent, siehe 'Mailname' auf Seite 172.
  - `/etc/exim/exim.conf`
  - `/etc/mailname`
  - `/etc/aliases`
  - `/etc/email-addresses`
- `fetchmail`: Ein Dämon, um Email von einem POP3-Account abzuholen,.
  - `/etc/fetchmailrc`

- `procmail`: ein Programm zur lokalen Zustellung von Email und Filter.
  - `~/.procmailrc`
- Hostname und DNS (Proxy, Cache, ...), siehe 'Hostname' auf Seite 172 und 'Domain Name Service (DNS)' auf Seite 172.
  - `/etc/host.conf`
  - `/etc/hostname`
  - `/etc/hosts`
  - `/etc/hosts.allow`
  - `/etc/hosts.deny`
  - `/etc/resolv.conf`
  - `/etc/bind/named.conf` (edit)
  - `/etc/bind/db.lan` (muss für LAN-Hosts hinzugefügt werden)
  - `/etc/bind/db.`*`192.168.0`* (muss für LAN reverse hinzugefügt werden)
- DHCP, siehe 'Konfiguration der Netzwerkschnittstelle mit DHCP' auf Seite 175.
  - `/etc/dhcp3/dhclient.conf` (DHCP-Client)
  - `/etc/default/dhcp3-server` (DHCP- Server)
  - `/etc/dhcp3/dhcpd.conf` (DHCP-Server)
- `cvs`: Concurrent Version System, siehe 'Concurrent Versions System (CVS) – System für simultane Versionen' auf Seite 201.
  - `/etc/cvs-cron.conf`
  - `/etc/cvs-pserver.conf`
- `nfs-kernel-server`: Ein Dateisystem für das Netzwerk, siehe 'NFS-Konfiguration' auf Seite 40 (für unix-artige Systeme).
  - `/etc/exports`
- `samba`: Freigabe von Dateien und Druckern über das Netzwerk für Windows, siehe 'Samba-Konfiguration' auf Seite 41 und 'Samba' auf Seite 122.
  - `/etc/samba/smb.conf`
- Drucker-Dämon-System, siehe 'Drucker-Konfiguration' auf Seite 41.
  - `/etc/printcap` (für lpr)
- `apache` und `apache2`: Ein Web-Server.
  - `/etc/apache/*`
  - `/etc/apache2/*`
- `squid`: Ein Web-Proxy-Cache-Server.
  - `/etc/squid/*`

10.11 Netzwerk-Fehlersuche

Wenn im Netzwerk Probleme auftreten, sollten die Ausgaben der folgenden Programme geprüft werden:

```
# ifconfig
# cat /proc/pci
# cat /proc/interrupts
# dmesg | more
```

Weitere Informationen folgen auf das Kapitel 'Grundlagen – Prüfung des Netzwerks' auf Seite 117.

Bei Problemen mit einzelnen Websites, siehe 'Eigenartige Probleme beim Zugriff auf einige Webseiten' auf Seite 46.

10.12 Aufbau eines Gateway-Routers

Ein Debian-Rechner kann als Gateway verwendet werden, der Network Addresss Translation (NAT, auch als Masquerading bekannt), Email-Transfer, DHCP, DNS-Caching, HTTP-Proxy-Caching, CVS-Service, NFS-Service und Samba-Services anbietet. Siehe 'Hosts und IP im LAN' auf Seite 31 für Beispiele derlei Einrichtung.

10.12.1 Konfiguration von Netfilter

Das Netfilter/Iptables-Projekt erstellt ein Firewall-Subsystem für den Linux-Kernel 2.4 und später. Siehe Netfilter (`http://www.netfilter.org/`), dort werden viele Netzwerk-Konfiguraitonen erklärt.

Grundlagen von Netfilter

Netfilter-Prozess-Pakete verwenden fünf eingebaute Ketten. PREROUTING, INPUT, FORWARD, OUTPUT, and POSTROUTING.

```
                  routing
                 decision
IN ------> PRE ---> ------> FORWARD -----> ----> POST -----> OUT
interface  ROUTING  \       filter       /       ROUTING     interface
           DNAT     |       tracking     ^       SNAT
           REDIRECT |                    |       MASQUERADE
                    v                    |
                  INPUT                OUTPUT
                    | filter             ^ filter,DNAT
                    v                    |
                    \-->Lokale Prozesse--/
                     User-Space Programme
```

Netfilter-Tabelle

Pakete der einzelnen eingebauten Ketten werden anhand folgender Tabellen weiterverarbeitet.

- filter (packet filter, default)
  - INPUT (für Pakete, die ankommen)

- FORWARD (für Pakete, die hier weitergeleitet werden)
- OUTPUT (für lokal erzeugte Pakete).

- nat (network address translation)
  - PREROUTING (um Pakete zu verändern, sobald sie ankommen)
  - OUTPUT (um Pakete zu ändern, bevor sie weitergeleitet werden)
  - POSTROUTING (um Pakete zu verändern, bevor sie abgeschickt werden)
- Adressen-Umsetzung (Mangling) (Netzwerk-Adressen-Mangling ist erst ab Kernel 2.4.18 zu gebrauchen)
  - Alle fünf eingebauten Ketten.

Netfilter Ziele

Firewall-Regeln haben verschiedene Ziele:

- Vier elementare Ziele:
  - ACCEPT bedeutet, dass das Paket durchgelassen wird.
  - DROP bedeutet, dass das Paket verworfen wird.
  - QUEUE bedeutet, dass das Paket an den Benutzer-Adressraum (Userspace) weitergeleitet wird (falls der Kernel dies unterstützt).
  - RETURN verläßt diese Kette und setzt die Bearbeitung an der vorherigen (aufrufenden) Kette fort.
- erweiterte Ziele:
  - LOG schaltet das Loggen im Kernel an.
  - REJECT verwirft das Paket und schickt ein Fehlerpaket zurück.
  - SNAT verändert die Quelladresse des Pakets und wird nur in der POSTROUTING-Kette verwendet. (nur für die NAT-Tabelle)

    ```
    --to-source ipaddr[-ipaddr][:port-port]
    ```
  - MASQUERADE ist das gleiche wie SNAT, nur für dynamisch zugewiesene IP (Einwahl)-Verbindungen. (nur für die NAT-Tabelle)
    ```
    --to-ports port[-port]
    ```
  - DNAT verändert die Ziel-Adresse des Pakets und wird in den PREROUTING und OUTPUT-Ketten verwendet, oder in in Benutzer-definierten Ketten, die von diesen Ketten aufgerufen werden. (nur für die NAT-Tabelle)

    ```
    --to-destination ipaddr[-ipaddr][:port-port]
    ```
  - REDIRECT verändert die IP-Zieladresse, um das Paket an die Maschine zu senden.
    ```
    --to-ports port[-port]
    ```

Netfilter-Befehle

Die elementaren Befehle von `iptables` sind:

```
iptables        -N chain             # create a chain
```

```
iptables        -A chain \            # add rule to chain
                -t table \            # use table (filter, nat, mangle)
                -p protocol \         # tcp, udp, icmp, or all,
                -s source-address[/mask] \
                --sport port[:port] \ # source port if -p is tcp or udp
                -d destination-address[/mask] \
                --dport port[:port] \ # dest. port if -p is tcp or udp
                -j target \           # what to do if match
                -i in-interface-name \ # for INPUT, FORWARD, PREROUTING
                -o out-interface-name # for FORWARD, OUTPUT, POSTROUTING
```

Network Address Translation

Rechner un einem lokalen Netzwerk können Internet-Angebote über einen Gateway-Rechner aufrufen, der die IP-Adressen zwischen dem lokalen Netz und den im Internet verwendeten IP-Adressen umsetzt.

```
 # apt-get install ipmasq
```

Um den Schutz durch `ipmasq` zu verbessern, können die Beispielregeln verwendet werden. Siehe `/usr/share/doc/ipmasq/examples/stronger/README`. In Woody mit kernel-image-2.4 muss das richtige Kernelmodul geladen werden. Die Sarge-Version von ipmasq hat dieses Problem nicht. Siehe 'Netzwerk-Funktionalität' auf Seite 89 für Konfigurationsanleitungen.

Wird Debian mit kernel-image-2.2 verwendet, muss `Z92timeouts.rul` in `/etc/masq/rules` wie folgt verändert werden, um sicher zu stellen, dass die Verbindung zu anderen Rechnern länger aufrecht erhalten wird (dies ist hilfreich beim Laden großer Emails, etc.):

```
# tcp, tcp-fin, udp
# 2h, 10 Sek, 160 Sek. - Voreinstellung
# 1 Tag, 10 Min, 10 Min - laengeres Beispiel
$IPCHAINS -M -S 86400 600 600
```

Wenn das Netzwerk mit einer PCMCIA-Netzwerkkarte verwendet wird, muss `ipmasq` aus `/etc/pcmcia/network.opts` (d.h. `/usr/share/doc/ipmasq/ipmasq.txt.gz`) oder aus `/etc/network/interfaces` (siehe 'Netzwerk-Konfiguration und PCMCIA' auf Seite 186 und 'Umschalten der Netzwerkkonfiguration' auf Seite 183) gestartet werden.

Umlenkung von SMTP-Verbindungen (2.4)

Ein Email-Programm kann an verschiedenen Netzwerken betrieben werden, ohne es neu zu konfigurieren.

Folgende Regeln fügen mittels `iptables` Regeln hinzu, die dafür sorgen, dass die SMTP-Verbindung zum Gateway-Rechner aufgebaut wird.

```
# iptables -t nat -A PREROUTING -s 192.168.1.0/24 -j REDIRECT \
-p tcp --dport smtp --to-port 25 # smtp=25, INPUT is open
```

Einen Satz weitaus ausführlicherer Umlenkungsregeln ermöglicht das `ipmasq`-Paket, wenn die Datei `M30redirect.def` (`http://www.debian.org/doc/manuals/debian-reference/examples/`) in das Verzeichnis `/etc/ipmasq/rules` kopiert wird.

10.12.2 Verschiedene Netzwerk-Verbindungen verwalten

[FIXME] Policy routing (by Phil Brutsche `<pbrutsch@tux.creighton.edu>`): Siehe iproute manual (`http://lartc.org/`) für Details. Traffic-Kontrolle (tc) ist ebenfalls interessant.

Umgebung:

```
eth0: 192.168.1.2/24; gateway 192.168.1.1
eth1: 10.0.0.2/24; gateway 10.0.0.1
Kein Masquerading auf dieser Maschine.
```

Spezielle Magie:

1. ip rule add from 192.168.1.2 lookup 1
2. ip rule add from 10.0.0.2 lookup 2
3. ip route add to default via 10.0.0.1 metric 0
4. ip route add to default via 192.168.1.1 metric 1
5. ip route add table 1 to 192.168.1.0/24 via eth0
6. ip route add table 1 to 10.0.0.2/24 via eth1
7. ip route add table 1 to default via 192.168.1.1
8. ip route add table 2 to 192.168.1.0/24 via eth0
9. ip route add table 2 to 10.0.0.2/24 via eth1
10. ip route add table 2 to default via 10.0.0.2

[FIXME] nicht getestet. Wie man eine Einwahlverbindung als Ersatz für eine schnelle Verbindung mit Selbstwahl einrichtet. Hier wird noch ein Patch benötigt. :)

Kapitel 11

Editoren

11.1 Populäre Editoren

Linux bietet viele Alternativen für Texteditoren in der Konsole, unter anderem:

- `vim`: Mächtiger und kleiner Editor aus der BSD-Welt. VI iMproved.
- `emacs`: Ultimativer und leistungsfähiger Editor aus der GNU-Welt von RMS (Richard M. Stallman).
- `xemacs`: Emacs: Die nächste Generation, ursprünglich von Lucid.
- `mcedit`: GNU-Editor für Anfänger. Identisch zu `mc`'s internem Editor. Vergleiche 'Editor' auf Seite 52.
- `ae`: Kleiner Standardeditor (Potato). Nicht verwenden.
- `nano`: Kleiner GNU-Standardeditor (Woody). Emuliert `pico`.
- `joe`: Für altes WordStar- oder TurboPascal-Feeling.
- `jed`: Schneller, reichhaltiger menübasierter Editor mit Emacs-Tastenzuordnungen.
- `jove`: Sehr kleiner Editor mit Emacs-Tastenzuordnungen.
- `nvi`: Neuer vi. Fehler-für-Fehler kompatibel mit dem ursprünglichen vi.

Verwenden Sie `update-alternatives --config editor`, um den Standardeditor zu wechseln. Viele Programme nutzen auch die Umgebungsvariablen `EDITOR` oder `VISUAL` um zu entscheiden, welcher Editor genutzt wird. Vergleichen Sie 'Editor' auf Seite 52.

Einige bemerkenswerte X-basierte Texteditoren:

- `gvim`: Vim mit GUI (graphische Benutzerschnittstelle) (`vim`- und `vim-gtk`-Paket)
- `emacs`: Der eine wahre Emacs (automatische X-Erkennung).
- `xemacs`: Emacs der nächsten Generation (automatische X-Erkennung).

Diese X-basierten Kommandos nutzen Standardoptionen wie `-fn` *`a24`*, was das Leben für ältere Leute wie mich einfacher macht :) Vergleiche 'X-Client Programme' auf Seite 139.

11.2 Rettungseditoren

Es gibt einige Editoren die sich in `/bin/` befinden. Einer von diesen sollte installiert sein, um das Editieren von Dateien zu vereinfachen, wenn auf `/usr/` nicht zugegriffen werden kann.

- elvis-tiny: Minimaler vi-Editor (vi zum Starten)
- nano-tiny: Minimaler nicht-vi-Editor (nano-tiny zum Starten)
- nano: Minimaler nicht-vi-Editor (nano zum Starten) (Sarge)
- ed: Minimaler Editor (immer vorhanden, aber schwierig zu benutzen)

11.3 Emacs und Vim

11.3.1 Vim Hinweise

Lesen Sie das "VIM - main help file"-Dokument durch Drücken von <F1> während das Programm läuft.

```
<F1>           Hilfe
<Esc>          Zurück zum normalen Modus
V              Visueller Modus
i              Eingabemodus
:              Kommandozeilen Kommandos
:set tw=72     Setzen der Textbreite auf 72
<F11>          Eingabe- (Paste) Modus
:r! date -R    Einfügen des Datums entsprechend RFC-822
qa             Aufnahme der Tastenanschläge in Register a
q              Beenden der Aufnahme
@a             Ausführen der Tastenanschläge aus Register a
:edit text.txt Editieren einer anderen Datei text.txt
:wnext         Sichern der Datei und bearbeiten der nächsten
```

q und @ können für einfache Makroaufnahmen und zum Abspielen genutzt werden. Um beispielsweise ein Makro zu erzeugen, das HTML-Italic-Tags um das Wort unter dem Cursor einfügt, kann qii<i>^[ea</i>^[q (wobei ^[die ESC-Taste ist) genutzt werden. Die Eingabe von @i beim Start eines Wortes fügt <i> und </i> ein.

Vergleichen Sie auch 'Benutzung von GnuPG mit Vim' auf Seite 233.

11.3.2 Emacs-Hinweise

```
<F1>              Hilfe
<F10>             Menü
C-u M-! date -R   Einfügen des Datums entsprechend RFC-822
```

11.3.3 Starten des Editors

```
Starten des Editors:                  emacs Datei  vim Datei
Starten mit vi-Kompatibilität:                     vim -C
```

```
Starten ohne vi-Kompatibilität:                         vim -N
Starten mit einkompilierten Standards: emacs -q        vim -N -u NONE
```

11.3.4 Zusammenfassung der Editorkommandos (Emacs, Vim)

```
beenden:                          C-x C-c          :qa /:wq /:xa /:q!
zurücksetzen/Kommandomodus:       C-g              <Esc>
zurück (links):                   C-b              h
vorwärts (rechts):                C-f              l
nächstes (herunter):              C-n              j
vorheriges (hoch):                C-p              k
StArt der Zeile (^):              C-a              0
Ende der Zeile ($):               C-e              $
mehrere Kommandos:                C-u nnn cmd      nnn cmd
mehrere Kommandos:                M-Ziffer cmd
abspeichern:                      C-x C-s          :w Datei
zum Anfang des Puffers:           M-<              1G
zum Ende des Puffers:             M->              G
eine Seite vorwärts scrollen:     C-v              ^F
1/2 Seiten vorwärts scrollen:                      ^D
eine Zeile vorwärts scrollen:                      ^E
eine Seite rückwärts scrollen:    M-v              ^B
1/2 Seiten rückwärts scrollen:                     ^U
eine Zeile rückwärts scrollen:                     ^Y
das andere Fenster scrollen:      M-C-v
unter Cursor löschen:             C-d              x
löschen bis zum Zeilenende:       C-k              D
inkrementelle Suche vorwärtS:     C-s
inkrementelle Suche Rückwärts:    C-r
vorwärtS suchen:                  C-s Enter        /
Rückwärts suchen:                 C-r Enter        ?
inkr. Suche nach reg. Ausd.:      M-C-s
inkr. Suche rückw. n. reg. Ausd.: M-C-r
suche nach regulären Ausdruck:    M-C-s Enter      /
suche rückw. nach reg. Ausd.:     M-C-r Enter      ?
Hilfe:                            C-h C-h          :help
Hilfe Apropos:                    C-h a
Hilfe zu Tastenzuordnungen:       C-h b            :help [key]
Info Seiten anzeigen:             C-h i
Hilfe Hauptmodus:                 C-h m
Hilfe Tutorial:                   C-h t            :help howto
rückgängig:                       C-_              u
nochmals tun (redo):              C-f              ^R
markiere die Cursorposition:      C-@              m{a-zA-Z}
wechseln der Markierung und Pos.: C-x C-x
```

```
springe zur Mark. in akt. Datei:                '{a-z}
springe zur Mark. in jeder Datei:               '{A-Z}
Bereich kopieren:              M-w              {visual}y
Bereich entfernen:             C-w              {visual}d
Bereich einfügen:              C-y
einfügen des gelöschten Textes: M-y             p
in Großbuchstaben umwandeln:    C-x C-u         {visual}U
in Kleinbuchstaben umwandeln:   C-x C-l         {visual}u
einfügen spezieller Zeichen:    C-q octalnum/keystroke
                                                ^V decimal/keystroke
ersetzen:                      M-x replace-string  :%s/aaa/bbb/g
ersetzen regulärer Ausdrücke:   M-x replace-regexp  :%s/aaa/bbb/g
vor Ersetzen nachfragen:       M-%                  :%s/aaa/bbb/gc
vor Ersetzen nachfragen:       M-x query-replace
vor Ersetzen reg. Aus. nachfragen: M-x query-replace-regexp
Datei öffnen:                  C-x C-f          :r file
Datei sichern:                 C-x C-s          :w
alle Puffer speichern:         C-x s            :wa
speichern unter:               C-x C-w file     :w file
nach Puffer fragen:            C-x b
Puffer anzeigen:               C-x C-b          :buffers
Read-only Flag wechseln:       C-x C-q          :set ro
nach Puffer fragen und löschen: C-x k
vertikal aufteilen:            C-x 2            :split
horizontal aufteilen:          C-x 3            :vsplit (ver. 6)
in anderes Fenster wechseln:    C-x o           ^Wp
entfernen dieses Fensters:      C-x 0           :q
entfernen anderer Fenster:      C-x 1           ^Wo
Shell im Hintergrund starten:   M-x compile
Shell im Hintergrund killen:    M-x kill-compilation
make starten:                                   :make Makefile
Fehlermeldungen überprüfen:     C-x`            :echo errmsg
Shell starten und aufzeichnen:  M-x shell       :!script -a tmp
...clean BS, ...                                :!col -b <tmp >record
...sichern/abrufen der Aufz.:   C-x C-w record  :r record
Shell starten:                 M-! sh           :sh
Kommando starten:              M-! Kmd          :!Kmd
Kommando starten und einfügen:  C-u M-! Kmd     :r!Kmd
Filter starten:                M-| Datei        {visual}:w Datei
Filter starten und einfügen:    C-u M-| Filter  {visual}:!Filter
Option anzeigen:                                :se[t] {Option}?
Option auf Standard zurücksetzen:               :se[t] {Option}&
boolsche Option zurücksetzen:                   :se[t] no{Option}
boolsche Option ändern:                         :se[t] inv{Option}
Textumbruch bei Spalte 72:                      :se tw=72
keinen Zeilenumbruch:                           :se tw=0
```

```
automatische Einrückung:                              :se ai
Tabulatoren expandieren:                              :se et
Kommentar angeben (E-Mail):                           :se comments=n:>,n:\|

GDB starten                     M-x gdb
beschreibe den GDB-Mode         C-h m
eine Zeile ausführen            M-s
nächste Zeile                   M-n
eine Anweisung ausführen (stepi) M-i
aktuelle Funktion beenden       C-c C-f
weiter                          M-c
zur übergeordneten Funktion     M-u
zur untergeordneten Funktion    M-d
copy number from point, insert at the end
                                C-x &
setze einen Haltepunkt          C-x SPC
```

11.3.5 Vim-Konfiguration

Um alle Vim-Eigenschaften und Syntax-Highlighting zu nutzen, müssen die folgenden Zeilen in `~/.vimrc` oder `/etc/vimrc` eingefügt werden:

```
set nocompatible
set nopaste
set pastetoggle=<f11>
syn on
```

Der Paste-Modus unterbindet Probleme mit der automatischen Einrückung bei Markieren-und-Einfügen-Operationen in konsolenbasierten Terminals. Dies bewirkt mehr als ein einfaches ":set noai".

Vergleichen Sie 'Benutzung von GnuPG mit Vim' auf Seite 233 für die GnuPG-Integration.

11.3.6 Ctags

Starten Sie `apt-get install exuberant-ctags` und `ctags` auf den Quelldateien. Geben Sie `:tag` *`Funktionsname`* in Vim ein, um zur Zeile zu springen, wo *Funktionsname* beginnt. Die Tags funktionieren mit C, C++, Java, Python und vielen anderen Sprachen.

Emacs besitzt die gleichen ctags-Fähigkeiten.

11.3.7 Konvertierung eines mit Syntax-Highlighting versehenen Schirms in HTML

`so \$VIMRUNTIME/syntax/2html.vim` vom Vim-Kommandomodus konvertiert hervorgehobenen Text nach HTML. Sichern Sie dies mit `:w Datei.html` und `:q`. Dies ist nützlich für C-Code, etc.

11.3.8 Aufteilen des Bildschirms mit `vim`

`vim` kann mehrere Dateien in mehrfach aufgeteilten Bildschirmen editieren. Geben Sie `:help usr_08.txt` für Details ein.

Um die Bildschirmdarstellung für verschiedene Dateien aufzuteilen, ist am vi-Kommandoprompt Folgendes einzugeben:

```
:split andere-Datei
:vsplit andere-Datei
```

Beim Shell-Prompt führt

```
$ vi -o Datei1.txt Datei2.txt   # horizontale Aufteilung
$ vi -O Datei1.txt Datei2.txt   # vertikale Aufteilung
```

zu einem Mehr-Fenster-vi.

```
$ vimdiff Datei.txt~ Datei.txt   # überprüfen der Änderungen von
$ vimdiff Datei.en Datei.fr      #  Datei.txt bzw. der Übersetzung
$ gvimdiff Datei.txt~ Datei.txt  # in X
```

ermöglicht einen schönen Blick auf die Unterschiede zwischen der ursprünglichen und der Backup-Datei. In SGML erkennt es Tags, so dass der Vergleich von Übersetzungen in diesem Modus sehr gut funktioniert.

Spezielle Cursor-Bewegungen mit Strg-W:

```
Strg-W +      vergrößern des Fensters
Strg-W -      verkleinern des Fensters
Strg-W h      wechseln ins linke Fenster
Strg-W j      wechseln in untere Fenster
Strg-W k      wechseln ins obere Fenster
Strg-W l      wechseln ins rechte Fenster
...
```

Nutzen Sie Folgendes zur Kontrolle (Synchronisation) des Bildschirm Scrollens:

```
:set scrollbind
:set noscrollbind
```

Kapitel 12

Versions-Kontroll-Systeme

12.1 Concurrent Versions System (CVS) – System für simultane Versionen

Überprüfen Sie `/usr/share/doc/cvs/html-cvsclient`, `/usr/share/doc/cvs/html-info` und `/usr/share/doc/cvsbook` mit `lynx` oder starten Sie `info cvs` und `man cvs` für detaillierte Informationen.

12.1.1 Installation eines CVS-Servers

Das folgende Setup erlaubt commit's ins CVS-Repository nur durch ein Mitglied der ″src“ Gruppe und die Administration des CVS nur durch ein Mitglied der ″staff“ Gruppe. Dies reduziert die Möglichkeit, den Server versehentlich zu misskonfigurieren.

```
# cd /var/lib; umask 002; mkdir cvs # [Woody] FSH
# apt-get install cvs cvs-doc cvsbook
# export CVSROOT=/var/lib/cvs
# cd $CVSROOT
# chown root:src .          # "staff" um bei neuen Projekten mehr
                            # zu restriktieren
# chmod 3775 .              # bei "staff" statt "src", 2775 verwenden
# cvs -d /var/lib/cvs init # es ist sicherer -d anzugeben
# cd CVSROOT
# chown -R root:staff .
# chmod 2775 .
# touch val-tags
# chmod 664 history val-tags
# chown root:src history val-tags
```

12.1.2 Beispiele für CVS-Sitzungen

Das Folgende stellt die Shell-Umgebung für den CVS-Repository-Zugriff ein.

Anonymes CVS (nur Download)

Entfernter nur lesbarer Zugang:

```
$ export CVSROOT=:pserver:anonymous@cvs.sf.net:/cvsroot/qref
$ cvs login
$ cvs -z3 co qref
```

Nutzen eines lokalen CVS-Servers

Lokaler Zugang von der Shell auf der selben Maschine:

```
$ export CVSROOT=/var/lib/cvs
```

Nutzen eines entfernten CVS-pserver

Entfernter Zugriff ohne SSH (verwenden Sie das RSH Protokoll in `cvs`):

```
$ export CVSROOT=:pserver:account@cvs.foobar.com:/var/lib/cvs
$ cvs login
```

Dies ist für Lauschangriffe anfällig.

Nutzen von entfernten CVS mittels `ssh`

Entfernter Zugriff mit SSH:

```
$ export CVSROOT=:ext:account@cvs.foobar.com:/var/lib/cvs
```

oder für SourceForge:

```
$ export CVSROOT=:ext:account@cvs.sf.net:/cvsroot/qref
```

Man kann auch RSA Authentifizierung nutzen ('Mit weniger Passwörter agieren' auf Seite 150), was die Passwortabfrage überflüssig macht.

Erzeugen eines neuen CVS-Archivs

Für

```
EINTRAG          WERT              BEDEUTUNG
Quellbaum:       ~/Projekt-x       gesamter Quellcode
Projektname:     Projekt-x         der Name dieses Projekts
Vendor Tag:      Hauptzweig        Wert für ganzen Zweig
Release Tag:     Release-initial   Wert für eine spezielle Version
```

Danach

```
$ cd ~/Projekt-x                # wechseln ins Quellverzeichnis
 ... Erzeugen des Quellcodes ...
$ cvs import -m "Starte Projekt-x" Projekt-x Hauptzweig Release-initial
$ cd ..; rm -R ~/Projekt-x
```

Arbeiten mit CVS

Um mit *Projekt-x* unter Verwendung des lokalen CVS-Repositorys zu arbeiten:

```
$ cd                            # ins Arbeitsverzeichnis wechseln
$ cvs co Projekt-x              # Quellen aus dem CVS kopieren
$ cd Projekt-x
 ... ändern des Codes ...
$ cvs diff -u                   # wie diff -u Repository/ lokal/
$ cvs up -C modifizierte_Datei  # Änderungen rückgängig machen
$ cvs ci -m "Änderungen"        # sichern der lokalen Quellen
$ vi neue_hinzugefügte_Datei    #   ... ins CVS
$ cvs add neue_hinzugefügte_Datei
$ cvs ci -m "Neue Datei neue_hinzugefügte_Datei"
$ cvs up                        # verschmelzen mit der letzten
                                # Version aus dem CVS
 ... zum Erzeugen aller neu erstellten Unterverzeichnisse im CVS
 ... ist "cvs up -d -P" stattdessen zu verwenden
 ... achten Sie auf Ausgaben, die mit "C Dateiname" starten
 ... nicht modifizierter Code wird nach `.#Dateiname.Version'
     verschoben
 ... suchen Sie nach "<<<<<<<" und ">>>>>>>" in der Datei
$ cvs tag Release-1             # hinzufügen des Release Tag
 ... weitere Änderungen ...
$ cvs tag -d Release-1          # entfernen des Release Tag
$ cvs ci -m "mehr Kommentare"
$ cvs tag Release-1             # erneutes hinzufügen des Tag
```

```
$ cd                            # ins Arbeitsverzeichnis wechseln
$ cvs co -r Release-initial -d alt Projekt-x
 ... Originalversion ins alt Verzeichnis kopieren
$ cd alt
$ cvs tag -b Release-initial-bugfixes # erzeuge Zweig (-b)
 ... nun kann mit der alten Version gearbeitet werden (Tag=sticky)
$ cvs update -d -P              # keine leeren Verz. erstellen
 ... Quellbaum hat nun den sticky Tag "Release-initial-bugfixes"
 ... arbeiten in diesem Zweig
$ cvs up -d -P                  # synchronisieren mit von anderen
                           # modifizierten Dateien in diesem Zweig
$ cvs ci -m "einchecken in diesen Zweig"
$ cvs update -kk -A -d -P
 ... entfernen des sticky Tag und Inhalt vergessen
 ... Aktualisierung des Haupt"trunk" ohne Schlüsselwortersetzung
$ cvs update -kk -d -P -j Release-initial-bugfixes
 ... vermengen des Release-initial-bugfixes Zweigs in den
 ... Haupt"trunk" ohne Schlüsselwortersetzung. Beseitigen Sie
 ... Konflikte mit einem Editor.
$ cvs ci -m "vermenge Release-initial-bugfixes"
$ cd
$ tar -cvzf Projekt-x-alt.tar.gz alt # Archiv erstellen, -j für bz2
$ cvs release -d alt            # lokale Quellen entfernen (optional)
```

Nette Optionen zur Erinnerung (als erstes Argument von `cvs` nutzen):

```
-n      Probelauf, hat keinen Effekt
-t      Anzeigen von Mitteilungen zur CVS-Aktivität
```

Exportieren von Dateien aus dem CVS

Um die letzte Version aus dem CVS zu nutzen, verwenden Sie "tomorrow“ (morgen):

```
$ cvs ex -D tomorrow Modulname
```

Verwalten des CVS

Fügen Sie ein Alias zum Projekt hinzu (lokaler Server):

```
$ su - admin           # ein Mitglied von staff
$ export CVSROOT=/var/lib/cvs
$ cvs co CVSROOT/modules
$ cd CVSROOT
```

```
$ echo "px -a Projekt-x" >>modules
$ cvs ci -m "Nun ist px ein Alias für Projekt-x"
$ cvs release -d .
$ exit                 # oder Strg-D um von su zurückzukehren
$ cvs co -d Projekt px
 ... auschecken von Projekt-x (Alias: px) aus dem CVS ins
 ... Verzeichnis Projekt
$ cd Projekt
 ... Änderungen vornehmen ...
```

12.1.3 Fehlersuche im CVS

Dateirechte im Repository

CVS überschreibt nicht die Datei im Repository, sondern ersetzt sie mit einer anderen. Deshalb sind *Schreibrechte im Repository Verzeichnis* wichtig. Für jedes neue Repository ist Folgendes zu starten, um diese Bedingung zu sichern.

```
# cd /var/lib/cvs
# chown -R root:src Repository
# chmod -R ug+rwX   Repository
# chmod 2775 Repository # wenn nötig, auch bei Unterverzeichnissen
```

Das ausführbar-Bit

Das ausführbar-Bit einer Datei wird beibehalten beim Checkout. Wann immer Sie Probleme mit dem ausführbar-Recht in ausgecheckten Dateien haben, ändern Sie die Rechte der Datei im CVS-Repository mit dem folgenden Kommando:

```
# chmod ugo-x Dateiname
```

12.1.4 CVS-Kommandos

Es folgen CVS-Kommandos mit deren Kürzeln.

```
{add|ad|new} [-k kflag] [-m 'Bemerkung'] Dateien...
{admin|adm|rcs} [rcs-Optionen] Dateien...
{annotate|ann} [Optionen] [Dateien...]
{checkout|co|get} [Optionen] Module...
{commit|ci|com} [-lnR] [-m 'Bemerkung fürs Protokoll' | -f Datei] \
        [-r Revision] [Dateien...]
{diff|di|dif} [-kl] [rcsdiff_Optionen] [[-r rev1 | -D Datum1] \
```

```
         [-r rev2 | -D Datum2]] [Dateien...]
{export|ex|exp} [-flNn] -r rev|-D Datum [-d dir] [-k kflag] Modul...
{history|hi|his} [-report] [-flags] [-options args] [Dateien...]
{import|im|imp} [-options] Repository vendortag releasetag...
{login|logon|lgn}
{log|lo|rlog} [-l] rlog-Optionen [Dateien...]
{rdiff|patch|pa} [-flags] [-V vn] [-r t|-D d [-r t2|-D d2]] Module...
{release|re|rel} [-d] Verzeichnisse...
{remove|rm|delete} [-lR] [Dateien...]
{rtag|rt|rfreeze} [-falnR] [-b] [-d] [-r Tag | -D Datum] \
         symbolic_tag Module...
{status|st|stat} [-lR] [-v] [Dateien...]
{tag|ta|freeze} [-lR] [-F] [-b] [-d] [-r Tag | -D Datum]  [-f] \
         symbolic_tag [Dateien...]
{update|up|upd} [-AdflPpR] [-d] [-r Tag|-D Datum] Dateien...
```

12.2 Subversion

Subversion ist ein Versions-Kontroll-System der nächsten Generation, das einmal CVS ersetzen soll. Die Entwickler betrachten es zurzeit als im "alpha" Stadium befindlich, aber es ist wahrscheinlich für die meisten Anwendungen stabil genug. Während dies geschrieben wird, ist Subversion nur in Debian `unstable` verfügbar.

12.2.1 Installation eines Subversion Servers

Das Meta-Paket `subversion` hängt von den benötigten Paketen `libapache2-svn` und `subversion-tools` ab, um einen Server aufzusetzen.

Aufsetzen eines Repository

Zurzeit setzt das `subversion`-Paket kein Repository auf, man muss es manuell erzeugen. Eine mögliche Stelle für ein Repository ist in `/var/local/repos`.

Erzeugen des Verzeichnisses

```
# mkdir -p /var/local/repos
```

und der Repository Datenbank:

```
# svnadmin create /var/local/repos
```

Nun ist das Repository für den WWW Server schreibbar zu machen:

```
# chown -R www-data:www-data /var/local/repos
```

Konfiguration von Apache2

Um den Zugriff auf das Repository mittels Nutzerauthentifizierung zu ermöglichen, fügen Sie Folgendes zu `/etc/apache2/mods-available/dav_svn.conf` hinzu (oder kommentieren Sie dies aus):

```
<Location /repos>
  DAV svn
  SVNPath /var/local/repos
  AuthType Basic
  AuthName "Subversion repository"
  AuthUserFile /etc/subversion/passwd
  <LimitExcept GET PROPFIND OPTIONS REPORT>
    Require valid-user
  </LimitExcept>
</Location>
```

Danach erzeugen Sie eine Nutzerauthentifizierungsdatei mit dem Kommando:

```
htpasswd2 -c /etc/subversion/passwd ein-Nutzer
```

Starten Sie Apache2 neu und Ihr neues Subversion Repository wird unter der URL http://*hostname*/repos verfügbar sein.

12.2.2 Verschieben eines CVS-Repositorys nach Subversion

12.2.3 Anwendungsbeispiele von Subversion

Die folgenden Abschnitte erklären die Verwendung verschiedener Kommandos in Subversion.

Erzeugen eines neuen Subversion-Archivs

Um ein neues Subversion-Archiv zu erstellen, verwenden Sie Folgendes:

```
$ cd ~/Projekt         # ins Quellcodeverzeichnis wechseln
$ svn import http://localhost/repos Projekt Projektname \
      -m "erster Projektimport"
```

Dies erzeugt ein Verzeichnis namens *Projektname* im Subversion Repository, das die Projektdateien enthält. Schauen Sie unter http://localhost/repos/, um zu sehen, ob die Datei vorhanden ist.

Arbeiten mit Subversion

Arbeiten mit *Projekt-y* und Subversion:

```
$ cd                            # ins Arbeitsverzeichnis wechseln
$ svn co http://localhost/repos/Projekt-y  # Quellcode auschecken
$ cd Projekt-y
 ... Änderungen durchführen ...
$ svn diff                      # wie diff -u Repository/ lokal/
$ svn revert modifizierte_Datei # Änderungen rückgängig machen
$ svn ci -m "Änderungen"        # Änderungen einchecken
$ vi neue_hinzugefügte_Datei
$ svn add neue_hinzugefügte_Datei
$ svn add Verzeichnis1          # alle Dateien unter Verzeichnis1
                                # rekursiv hinzufügen
$ svn add -N Verzeichnis2       # Verz. nicht rekursiv hinzufügen
$ svn ci -m "neue Dateien hinzugefügt"
$ svn up                        # vermenge mit neuester Version
                                # aus Repository
$ svn log                       # zeige alle eingebrachten Änd.
$ svn copy http://localhost/repos/Projekt-y \
      http://localhost/repos/Projekt-y-branch \
      -m "erzeuge Zweig von Projekt-y"
$ svn copy http://localhost/repos/Projekt-y \
      http://localhost/repos/Proj-y_release1.0 \
      -m "Projekt-y 1.0 Release"# fügte release Tag hinzu
 ... Es ist zu beachten, dass das Erzeugen eines Zweigs und eines
 ... Tags das selbe ist. Der einzige Unterschied ist, dass Zweige
 ... "committed" werden, Tags nicht.
 ... Änderungen im Zweig durchführen ...

$ # vermengen der Kopie des Zweiges mit Hauptkopie
$ svn merge http://localhost/repos/Projekt-y \
   http://localhost/repos/Projekt-y-branch
$ svn co -r 4 http://localhost/repos/Projekt-y # Rev. 4 besorgen
```

Kapitel 13

Programmierung

Verwenden Sie nicht "test" als Namen für eine ausführbare Testdatei. `test` ist ein Shell-builtin.

13.1 Wo man startet

Referenzen:

- Dokumente und Beispiele unter `/usr/share/doc/Paket`
- Unix / Programming Information (`http://arioch.unomaha.edu/~jclark/#info`)
- *Linux Programming Bible* (John Goerzen/IDG books)

Viele lange info-Dokumente können als Taschenbuch von GNU (`http://www.gnu.org/`) erhalten werden.

Die nächsten vier Abschnitte enthalten Beispielskripte in verschiedenen Sprachen, die eine Textdatei mit Zugangsinformationen erzeugen, welche an `/etc/passwd` mittels einer Stapelverarbeitung wie das `newusers`-Programm angehangen werden. Jedes Skript erfordert als Eingabe eine Datei mit Zeilen der Gestalt `Vorname Nachname Passwort`. (Homeverzeichnisse werden mit diesen Skripten nicht erzeugt.)

13.2 Shell

Das Lesen von Shellskripten ist der **beste** Weg, um zu verstehen, wie ein Unix-artiges System arbeitet. Ich gebe hier einige Hinweise zur Shellprogrammierung an. Lesen Sie Shell-Fehler (`http://www.greenend.org.uk/rjk/2001/04/shell.html`) um aus Fehlern zu lernen.

13.2.1 Bash – interaktive GNU-Standard-Shell

Referenzen für Bash:

- `bash(1)`
- `info bash`
- das LDP BASH Programming - Introduction HOWTO (`http://www.tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html`) als Information für Starter.
- `mc /usr/share/doc/bash/examples/ /usr/share/doc/bash/` (Installieren Sie das `bash-doc`-Paket für Beispieldateien.)
- *Learning the bash Shell*, 2. Ausgabe (O'Reilly)

Kurzes Programmbeispiel (erzeugt Zugangswerte für `newusers` von der Standardeingabe):

```
#!/bin/bash
# (C) Osamu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
pid=1000;
while read n1 n2 n3 ; do
if [ ${n1:0:1} != "#" ]; then
let pid=$pid+1
echo ${n1}_${n2}:password:${pid}:${pid}:,,,/home/${n1}_${n2}:/bin/bash
fi
done
```

13.2.2 POSIX-Shells

Verschiedene Pakete bieten eine POSIX-Shell in Debian an:

- `dash` (Sarge)
  - Priority: optional
  - Installed-Size: 176
  - Kleinste und schnellste – das Beste zum Booten
- `ash` (Woody)
  - Priority: optional
  - Installed-Size: 180
  - Kleiner und viel schneller – gut für Bootprozess
- `bash`
  - Essential: yes
  - Priority: required
  - Installed-Size: 580
  - Größer und vielfältig – viele implementierte Erweiterungen
- `pdksh`
  - Priority: optional
  - Installed-Size: 408
  - Komplett AT&T ksh ähnlich

Wollen Sie portable Shellskripte schreiben, ist es am besten es als POSIX-Shellskript zu schreiben. Verwenden Sie `/bin/sh` verlinkt auf `ash` (oder `dash`), um dessen POSIX-Konformität zu testen. Vermeiden Sie Skripte mit "bashisms" oder "zshisms" zu schreiben. Vermeiden Sie zum Beispiel:

- `if [ `*`foo`*` == `*`bar`*` ] ; then ...`

- `diff -u` *`Datei`*`.c{.orig,}`
- `mkdir` *`/foo`*`{`*`bar`*`,`*`baz`*`}`

Die Erläuterungen zu Shells in diesem Dokument beziehen sich nur auf POSIX-Shells und damit nicht auf `csh`-artige Shells wie `tcsh`.

13.2.3 Shellparameter

Verschiedene **spezielle Parameter** zum Einprägen:

```
$0      = Name der Shell oder des Shellskripts
$1      = erstes Shellargument
 ...
$9      = neuntes Shellargument
$#      = Anzahl der Parameter
"$*"    = "$1 $2 $3 $4 ... $n"
"$@"    = "$1" "$2" "$3" "$4" ... "$n"
$?      = Exit-Status des zuletzt ausgeführten Kommandos
$$      = Prozessnummer (PID) dieses Shellskripts
$!      = PID des zuletzt ausgeführten Hintergrundkommandos
```

Grundlegende **Parameterauswertungen** zum Einprägen:

```
Ausdruck        Wenn var gesetzt ist Wenn var nicht gesetzt ist
${var:-string}  $var                 string
${var:+string}  string               null
${var:=string}  $var                 string (und setzt var=string)
${var:?string}  $var                 (string Ausgabe und Abbruch)
```

Der Doppelpunkt ':' in all diesen Operatoren ist optional.
- Mit ':' = Operatortest für "existiert" und "nicht Null".
- Ohne ':' = Operatortest nur für "existiert".

Grundlegende **Parametersubstitutionen** zum Einprägen:

```
Ausdruck        Ergebnis
${var%suffix}   Entferne kleinstes suffix-Muster
${var%%suffix}  Entferne größtes suffix-Muster
${var#präfix}   Entferne kleinstes präfix-Muster
${var##präfix}  Entferne größtes präfix-Muster
```

13.2.4 Shellumleitung

Grundlegende **Umleitungen** zum Einprägen ([*n*] ist eine optionale Nummer die den Datei-Descriptor spezifiziert):

```
[n]> Datei          Umleiten von stdout (oder n) zu Datei.
[n]>> Datei         Anhängen von stdout (oder n) an Datei.
[n]< Datei          Umleiten von stdin (oder n) von Datei.
[n1]>&n2            Umleiten von stdout (oder n1) zu n2.
2> Datei  >&2       Umleiten von stdout und stderr zu Datei.
 > Datei 2>&1       Umleiten von stdout und stderr zu Datei.
| Kommando          stdout an Kommando weiterreichen.
2>&1 | Kommando     stderr und stdout an Kommando weiterreichen.
```

Es wurden folgende Schreibweisen verwendet:
- stdin: Standardeingabe (Datei-Descriptor = 0)
- stdout: Standardausgabe (Datei-Descriptor = 1)
- stderr: Fehlerausgabe (Datei-Descriptor = 2)

Die Shell erlaubt mittels des eingebauten `exec`, Dateien mit einem beliebigen Datei-Descriptor zu öffnen.

```
$ echo Hallo >foo
$ exec 3<foo 4>bar  # Dateien öffnen
$ cat <&3 >&4       # umleiten von stdin nach 3, stdout nach 4
$ exec 3<&- 4>&-    # Dateien schließen
$ cat bar
Hallo
```

Dabei bedeutet *n*`<&-` und *n*`>&-` das Schließen des Datei-Descriptors *n*.

13.2.5 Bedingte Ausdrucke in der Shell

Jedes Kommando gibt einen **Exit-Status** zurück, was für einen bedingten Ausdruck verwendet werden kann:
- Erfolg: 0 (wahr)
- Fehler: 1–255 (falsch)

Es ist zu beachten, dass die Verwendung des 0 Werts im Sinne von "wahr" sich von den üblichen Konventionen in anderen Bereichen der Programmierung unterscheidet. Auch ist '[' das Äquivalent des `test`-Kommandos, das sein Argument bis zu ']' als bedingten Ausdruck auswertet.

Grundlegende **bedingte Ausdrucke** zum Einprägen sind:

```
Kommando && bei_Erfolg_dies_starten || true
Kommando || bei_Misserfolg_dies_starten

if [ bedingter_Ausdruck ]; then
 bei_Erfolg_dies_starten
else
 bei_Misserfolg_dies_starten
fi
```

Es wurde `||  true` verwendet, um sicherzustellen, dass das Shellskript in dieser Zeile nicht versehentlich abgebrochen wird, wenn die Shell (oder `set`) mit der Option `-e` gestartet wird.

Dateivergleichsoperatoren im bedingten Ausdruck sind:

```
Ausdruck          Wahr wenn ...
-e Datei          Datei existiert.
-d Datei          Datei existiert und ein Verzeichnis ist.
-f Datei          Datei existiert und eine reguläre Datei ist.
-w Datei          Datei existiert und schreibbar ist.
-x Datei          Datei existiert und ausführbar ist.
Datei1 -nt Datei2 Datei1 neuer als Datei2 ist. (Modifizierungszeit)
Datei1 -ot Datei2 Datei1 älter als Datei2 ist. (Modifizierungszeit)
Datei1 -ef Datei2 beide die selbe Device- und Inode-Nummer haben.
```

Stringvergleichsoperatoren im bedingten Ausdruck sind:

```
Ausdruck       Wahr wenn ...
     -z str    die Länge von str Null ist.
     -n str    die Länge von str nicht Null ist.
str1 == str2   str1 und str2 gleich sind.
str1 =  str2   str1 und str2 gleich sind.
   ("=" sollte statt "==" für POSIX-Konformität verwendet werden)
str1 != str2   str1 und str2 ungleich sind.
str1 <  str2   str1 vor str2 einsortiert wird (locale abhängig).
str1 >  str2   str1 nach str2 einsortiert wird (locale abhängig).
```

Arithmetische Ganzzahlvergleiche im bedingten Ausdruck sind `-eq`, `-ne`, `-lt`, `-le`, `-gt` und `-ge`.

13.2.6 Kommandozeilenbearbeitung

Die Shell führt ein Skript wie folgt aus:

- aufsplitten in **Token** durch Metazeichen: LEERZEICHEN, TABULATOR, NEUEZEILE, ;, (,), <, >, |, &
- überprüfen auf **Schlüsselworte** außerhalb von ″. . . “ und ′. . . ′ (Schleife)
- expandieren der **Aliase** außerhalb von ″. . . “ und ′. . . ′ (Schleife)
- expandieren der **geschweiften Klammern**, `a{1,2}` -> `a1 a2` außerhalb von ″. . . “ und ′. . . ′
- expandieren der **Tilde**, *~Nutzer* -> *Nutzer*s Homeverzeichnis außerhalb von ″. . . “ und ′. . . ′
- expandieren der **Parameter**, $*PARAMETER* außerhalb von ′. . . ′
- expandieren der **Kommandosubstitutionen**, $(*Kommando*) außerhalb von ′. . . ′
- aufsplitten in **Wörter** mit $IFS außerhalb von ″. . . “ und ′. . . ′
- expandieren von *?[] in **Pfadnamen** außerhalb von ″. . . “ und ′. . . ′
- schauen nach **Kommando**

  - Funktion
  - builtin
  - Datei in $PATH
- Schleife

Einfache Anführungszeichen innerhalb von doppelten Anführungszeichen haben keinen Effekt.

Das Ausführen von `set -x` in der Shell oder das Starten der Shell mit der Option `-x` veranlasst die Shell alle gestarteten Kommandos auszugeben. Dies ist sehr nützlich zum Debuggen.

13.3 Awk

Referenzen für Awk:
- *Effective awk Programming*, 3. Ausgabe (O'Reilly)
- *Sed & awk*, 2. Ausgabe (O'Reilly)
- `mawk(1)` und `gawk(1)`
- `info gawk`

Kurzes Programmbeispiel (erzeugt `newusers`-Kommandoeintrag):

```
#!/usr/bin/awk -f
# Skript zum Erzeugen einer für das 'newusers'-Kommando geeigneten
# Datei aus einer Datei bestehend aus Nutzer-IDs und Passwörtern
# in der Form:  Vorname Nachname Passwort
# Copyright (c) KMSelf Sat Aug 25 20:47:38 PDT 2001
# Verteilbar unter GNU GPL v 2 oder (je nach Ihrer Wahl) einer
# späteren Version.
# Dieses Programm wird OHNE JEGLICHE HAFTUNG vertrieben.

BEGIN {
    # zuweisen der anfänglichen UID, GID
    if ( ARGC > 2 ) {
        startuid = ARGV[1]
        delete ARGV[1]
    }
    else {
        printf( "Verwendung: newusers startUID Datei\n" \
        " wobei:\n" \
        "  startUID die erste Nutzer-ID zum Hinzufügen ist und\n"\
        "  Datei eine Eingabedatei der folgenden Gestalt ist:\n" \
        "    Vorname Nachname Passwort\n" \
        )
        exit
    }
```

```
    infile = ARGV[1]
    printf( "Erste UID: %s\n\n", startuid )
}

/^#/ { next }

{
    ++record
    first = $1
    last = $2
    passwd = $3
    user= substr( tolower( first ), 1, 1 ) tolower( last )
    uid = startuid + record - 1
    gid = uid
    printf( "%s:%s:%d:%d:%s %s,,/home/%s:/bin/bash\n",  \
        user, passwd, uid, gid, first, last, user \
        )
}
```

Zwei Pakete bieten POSIX-`awk` in Debian an:

- `mawk`
  - Priority: required
  - Installed-Size: 228
  - Kleiner und viel schneller – gut zur Standardinstallation
  - Compilezeit-Limitierungen bestehen
    - NF = 32767
    - sprintf buffer = 1020
- `gawk`
  - Priority: optional
  - Installed-Size: 1708
  - Größer und reichhaltig – viele implementierte Erweiterungen
    - System V Release 4 Version von UNIX
    - Bell Labs awk
    - GNU-spezifisch

13.4 Perl

Dies ist **der** Interpreter auf einem Unix-artigen System.

Referenzen für Perl:

- `perl(1)`
- *Programming Perl*, 3. Ausgabe (O'Reilly)
- Das Perl-Archiv (`http://www.perl.org/`)

Kurzes Programmbeispiel (erzeugt `newusers`-Kommandoeintrag):

```
#!/usr/bin/perl
# (C) Osamu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
$pid=1000;
while (<STDIN>) {
        if (/^#/) { next;}
        chop;
        $pid++;
        ($n1, $n2, $n3) = split / /;
        print $n1,"_",$n2,":", $n3, ":",$pid,
                  ":",$pid,",,,/home/",$n1,"_",$n2,":/bin/bash\n"
}
```

Installieren eines Perl-Moduls *Modul*:

```
# perl -MCPAN -e 'install Modul'
```

13.5 Python

Dies ist ein netter objektorientierter Interpreter.

Referenzen für Python:

- python(1)
- *Learning Python* (O'Reilly).
- Die Python Programmiersprache (http://www.python.org/)

Kurzes Programmbeispiel (erzeugt newusers-Kommandoeintrag):

```
#! /usr/bin/env python
import sys, string

# (C) Osamu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
# Portiert von awk Skript durch KMSelf Sat Aug 25 20:47:38 PDT 2001
# Dieses Programm wird OHNE JEGLICHE HAFTUNG vertrieben.

def usages():
    print \
"Verwendung: ", sys.argv[0], " start_UID [Dateiname]\n" \
"\tstartUID ist die erste Nutzer-ID zum Hinzufügen.\n" \
"\tDateiname ist eine Eingabedatei. " \
"Ohne Angabe wird die Standardeingabe verwendet.\n\n" \
"Format der Eingabedatei:\n" \
"\tVorname Nachname Passwort\n"
                return 1
```

```
def parsefile(startuid):
    #
    # filtern
    #
    uid = startuid
    while 1:
        line = infile.readline()
        if not line:
            break
        if line[0] == '#':
            continue
        (first, last, passwd) = string.split(string.lower(line))
        # obiges stürzt bei falscher Parameteranzahl ab :-)
        user = first[0] + last
        gid = uid
        lineout = "%s:%s:%d:%d:%s %s,,/home/%s:/bin/bash\n" %  \
            (user, passwd, uid, gid, first, last, user)
        sys.stdout.write(lineout)
        +uid

if __name__ == '__main__':
    if len(sys.argv) == 1:
        usages()
    else:
        uid = int(sys.argv[1])
        #print "# UID Start von: %d\n" % uid
        if len(sys.argv) > 1:
            infilename   = string.join(sys.argv[2:])
            infile = open(infilename, 'r')
            #print "# Lese Datei von: %s\n\n" % infilename
        else:
            infile = sys.stdin
        parsefile(uid)
```

13.6 Make

Referenzen für Make:

- `info make`
- `make(1)`
- *Managing Projects with make,* 2. Ausgabe (O'Reilly)

Einfache automatische Variablen:

Regelsyntax:

```
Ziel: [ Voraussetzungen ... ]
```

```
 [TAB]  Kommando1
 [TAB]  -Kommando2 # ignoriere Fehler
 [TAB]  @Kommando3 # unterdrücke Kommandozeilenausgabe
```

Hier ist `[TAB]` ein Tabulator. Jede Zeile wird von der Shell nach einer Variablensubstitution interpretiert. Verwenden Sie \ am Ende einer Zeile zur Fortsetzung des Skripts. Statt `$` ist `$$` für Umgebungsvariablen zu schreiben.

Implizite Regeln für *Ziel* und *Voraussetzungen* können beispielsweise wie folgt geschrieben werden:

```
%: %.c header.h
```

oder

```
%.o: %.c header.h
```

Hier enthält *Ziel* das Zeichen `%` (exakt einmal). `%` passt auf jeden nicht leeren Teilstring in den aktuellen Zieldateinamen. *Voraussetzungen* verwendet `%` ähnlich um auszudrücken, wie sich diese Namen zum aktuellen Ziel verhalten.

Suffixregeln sind der **veraltete** Weg zur Definition impliziter Regeln für `make`. Sie werden noch in GNU-`make` zur Kompatibilität unterstützt, aber man sollte äquivalente Musterregeln wann immer möglich verwenden:

```
alte Suffixregel --> neue Musterregel
.c:              --> %  : %.c
.c.o:            --> %.o: %.c
```

Automatische Variablen für diese Regel:

```
foo.o: neu1.c neu2.c alt1.c neu3.c
$@ == foo.o                         (Ziel)
$< == neu1.c                        (erstes der neueren Objekte)
$? == neu1.c neu2.c neu3.c          (alle neueren Objekte)
$^ == neu1.c neu2.c alt1.c neu3.c   (alle)
$* == '%' aktuelles Ziel ohne Suffix.
```

Variablenreferenzen:

```
foo1 := bar    # einmalige Expandierung
foo2  = bar    # rekursive Expandierung
foo3 += bar    # anhängen
SRCS := $(wildcard *.c)
```

```
OBJS := $(foo:c=o)
OBJS := $(foo:%.c=%.o)
OBJS := $(patsubst %.c,%.o,$(foo))
DIRS  = $(dir directory/filename.ext) # Extracts "directory"
 $(notdir NAMES...), $(basename NAMES...), $(suffix NAMES...) ...
```

Starten Sie `make -p -f/dev/null`, um alle internen automatischen Regeln zu sehen.

13.7 C

Vorbereitung:

```
# apt-get install glibc-doc manpages-dev libc6-dev gcc
```

Referenzen für C:
- `info libc` (Referenz der C-Bibliotheksfunktionen)
- `gcc(1)`
- `jede_C_Bibliotheksfunktion(3)`
- Kernighan & Ritchie, *Die Programmiersprache C*, 2. Ausgabe (Prentice Hall). `http://cm.bell-labs.com/cm/cs/cbook/`
- ISO C standard. `http://www.open-std.org/jtc1/sc22/wg14/`

13.7.1 Ein einfaches C-Programm (gcc)

Ein einfaches Beispiel zum Kompilieren von `example.c` mit einer Bibliothek `libm` in eine ausführbare Datei namens `Testbeispiel`:

```
$ cat > example.c <<EOF
#include <stdio.h>
#include <math.h>
#include <string.h>

int main(int argc, char **argv, char **envp){
        double x;
        char y[11];
        x=sqrt(argc+7.5);
        strncpy(y, argv[0], 10); /* Pufferüberlauf verhindern */
        y[10] = '\0'; /* füllen, so dass String mit '\0' endet */
        printf("%5i, %5.3f, %10s, %10s\n", argc, x, y, argv[1]);
        return 0;
}
EOF
```

```
$ gcc -Wall -g -o Testbeispiel example.c -lm
$ ./Testbeispiel
        1, 2.915, ./Testbeis,     (null)
$ ./Testbeispiel 1234567890qwertz
        2, 3.082, ./Testbeis, 1234567890qwertz
```

Hier wird `-lm` benötigt, um mit der Bibliothek `libm` für `sqrt()` zu linken. Die eigentliche Bibliothek ist in `/lib/` mit dem Dateinamen `libm.so.6`, was ein symbolischer Link auf `libm-2.1.3.so` ist.

Schauen Sie auf den letzten Parameter im Ausgabetext. Es gibt mehr als 10 Buchstaben, obwohl `%10s` spezifiziert ist.

Die Verwendung von Funktionen die Zeiger auf Speicherbereiche ohne Bereichscheck nutzen, wie `sprintf` und `strcpy` wird missbilligt, um das Ausnutzen von Pufferüberläufen zu verhindern, die obige Überlaufeffekte verwenden. Stattdessen sollte man `snprintf` und `strncpy` verwenden.

13.7.2 Fehlersuche

Fehlersuche (Debugging) mit `gdb`

Vorbereitung:

```
# apt-get install gdb
```

Referenzen für `gdb`:
- `info gdb` (Tutorial)
- `gdb(1)`
- `http://www.unknownroad.com/rtfm/gdbtut/gdbtoc.html`

Verwenden Sie `gdb`, um ein mit der `-g`-Option kompiliertes Programm zu debuggen. Viele Kommandos können abgekürzt werden. Vervollständigungen arbeiten wie in der Shell mit der Tabulator-Taste.

```
$ gdb Programm
(gdb) b 1                # Haltepunkt in Zeile 1 setzen
(gdb) run arg1 arg2 arg3 # Programm starten
(gdb) next               # nächste Zeile
...
(gdb) step               # einen Schritt vorwärts
...
(gdb) p parm             # parm ausgeben
...
(gdb) p parm=12          # Wert auf 12 setzen
```

Um aus Emacs heraus zu debuggen, wird auf 'Zusammenfassung der Editorkommandos (Emacs, Vim)' auf Seite 197 verwiesen.

Da alle installierten Binarys standardmäßig in einem Debian-System gestript sein sollten, sind die meisten Debug-Informationen entfernt. Um `gdb` sinnvoll zum Debuggen von Debian-Paketen einsetzen zu können, müssen entsprechende Pakete unter Beachtung von Folgendem neu gebaut werden:

- Erhöhen Sie die Paketversion (`http://www.debian.org/doc/debian-policy/ch-controlfields#s-f-Version`) durch Editieren von `debian/control`.
- Überprüfen Sie die Build-Skripte und stellen Sie sicher, dass `CFLAGS=-g -Wall` zum Kompilieren der Binarys verwendet wird.
- Exportieren Sie `DEB_BUILD_OPTIONS=nostrip,noopt` zum Bauen des Debian-Pakets.

Vergleichen Sie Policy 10.1 (`http://www.debian.org/doc/debian-policy/ch-files#s10.1`) für weitere Informationen.

Überprüfen der Abhängigkeiten von Bibliotheken

Verwenden Sie `ldd`, um die Abhängigkeiten eines Programms von Bibliotheken zu bestimmen:

```
$ ldd /bin/ls
        librt.so.1 => /lib/librt.so.1 (0x4001e000)
        libc.so.6 => /lib/libc.so.6 (0x40030000)
        libpthread.so.0 => /lib/libpthread.so.0 (0x40153000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Damit `ls` in einer `chroot`-Umgebung funktioniert, müssen die obigen Bibliotheken in der `chroot`-Umgebung vorhanden sein.

Die folgenden Kommandos sind auch nützlich:

- `strace`: verfolgt Systemaufrufe und Signale
- `ltrace`: verfolgt Bibliotheksaufrufe

Debuggen mit Tools zur Erkennung von Speicherlecks

Es gibt verschiedene Tools zur Erkennung von Speicherlecks in Debian.

- `njamd`
- `valgrind`
- `dmalloc`
- `electric-fence`
- `memprof`
- `memwatch` (nicht paketiert, verfügbar unter memwatch (`http://directory.fsf.org/devel/debug/memwatch.html`).)
- `mpatrol`
- `leaktracer`

- `libgc6`
- `Insure++` von Parasoft (`http://www.parasoft.com`). (nicht frei, kommerziell)

Überprüfen Sie auch Debugging Tools für dynamische Speicherzuordnung und -Management (`http://www.cs.colorado.edu/homes/zorn/public_html/MallocDebug.html`).

13.7.3 Flex – ein besseres Lex

`flex` ist ein schneller lexikalischer Analysegenerator.

Referenzen für `flex`:
- `info flex` (Tutorial)
- `flex(1)`

Sie müssen Ihre eigenen Funktionen `main()` und `yywrap()` implementieren oder Ihr `Programm.l` sollte wie folgt aussehen, um ohne eine Bibliothek zu kompilieren (`yywrap` ist ein Makro; `%option main` aktiviert `%option noyywrap` implizit):

```
%option main
%%
.|\n    ECHO ;
%%
```

Alternativ kann mit der `-lfl`-Linker-Option am Ende Ihrer `cc`-Kommandozeile kompiliert werden (wie AT&T-Lex mit `-ll`). Keine `%option` wird in diesem Fall benötigt.

13.7.4 Bison – ein besseres Yacc

Einige Pakete bieten einen Yacc kompatiblen LALR-Parser-Generator in Debian an:
- `bison`: GNU-LALR-Parser-Generator
- `byacc`: Der Berkeley-LALR-Parser-Generator
- `btyacc`: Rückfolgender Parser-Generator basierend auf `byacc`

Referenzen für `bison`:
- `info bison` (Tutorial)
- `bison(1)`

Sie müssen Ihre eigenen `main()` und `yyerror()` Funktionen implementieren. `main()` ruft `yyparse()`, was wiederum `yylex()` aufruft, das gewöhnlich von FleX erzeugt wird.

```
%%

%%
```

13.7.5 Autoconf

`autoconf` ist ein Tool zum Erzeugen von Shellskripten, die Softwarequellpakete automatisch so konfigurieren, dass sie sich an viele UNIX-artige Systeme unter Verwendung des vollständigen GNU-Build-Systems anpassen.

`autoconf` erzeugt das Konfigurationsskript `configure`. `configure` erzeugt automatisch ein angepasstes `Makefile` aus `Makefile.in`.

Kompilieren und Installieren eines Programms

Debian ändert keine Dateien unter `/usr/local/` (vergleichen Sie 'Unterstützung von Vielfalten' auf Seite 24). Kompilieren Sie ein Programm aus den Quellen, so sollten Sie es in `/usr/local/` installieren, damit es nicht mit Debian interferiert.

```
$ cd src
$ ./configure --prefix=/usr/local
$ make
$ make install # dies kopiert die Dateien ins System
```

Deinstallation eines Programms

Wenn Sie noch den Quellcode haben, dieser `autoconf/automake` nutzt und Sie noch wissen, wie Sie es konfiguriert haben, verfahren Sie wie folgt:

```
$ ./configure alle-verwendeten-Optionen
# make uninstall
```

Sind Sie sich absolut sicher, dass der Installationsprozess Dateien nur unter `/usr/local/` ablegt und es nichts wichtiges darunter gibt, können Sie alles wie folgt löschen:

```
# find /usr/local -type f -print0 | xargs -0 rm -f
```

Sind Sie nicht sicher, wo Dateien installiert sind, sollten Sie einen Blick auf `checkinstall` werfen, was einen leeren Pfad bei der Deinstallation liefert.

13.8 Web

Einfache interaktive dynamische Webseiten können wie folgt erstellt werden:

- Abfragen werden mittels HTML-Formularen dem Browser-Nutzer präsentiert.

- Ausfüllen von und Klicken auf Formulareinträge sendet eine URL mit kodierten Parametern [1] vom Browser zum Webserver. Zum Beispiel:
  - http://www.foo.dom/cgi-bin/program.pl?VAR1=VAL1&VAR2=VAL2&VAR3=VAL3
  - http://www.foo.dom/cgi-bin/program.py?VAR1=VAL1&VAR2=VAL2&VAR3=VAL3
  - http://www.foo.dom/program.php?VAR1=VAL1&VAR2=VAL2&VAR3=VAL3
- Ein CGI-Programm (ein beliebiges `program.*`) auf dem Webserver wird kodierte Parameter "`VAR1=VAL1 VAR2=VAL2 VAR3=VAL3`" als Inhalt der Umgebungsvariablen "`QUERY_STRING`" erhalten und sich selbst ausführen.
- Die Standardausgabe `stdout` des CGI-Programms wird zum Web-Browser gesendet und als eine interaktive dynamische Webseite präsentiert.

Aus Sicherheitsgründen wird es empfohlen keinen eigenen zusammengebastelten Code zum Parsen von CGI-Parametern zu verwenden. Es gibt etablierte Module dafür in Perl (man vergleiche 'Perl' auf Seite 215) und Python (man vergleiche 'Python' auf Seite 216). PHP (http://www.php.net/) unterstützt diese Funktionalität. Wenn eine Speicherung der Daten auf dem Client erfolgen soll, werden Cookys verwendet. Wenn eine Verarbeitung der Daten auf dem Client erwünscht ist, wird häufig Javascript benutzt.

Für weitere Informationen wird auf Das Common-Gateway-Interface (http://hoohoo.ncsa.uiuc.edu/cgi/), Die Apache-Software-Foundation (http://www.apache.org/) und JavaScript (http://www.mozilla.org/js/) verwiesen.

Die Suche nach "CGI tutorial" auf Google durch Eingabe einer kodierten URL http://www.google.com/search?hl=en&ie=UTF-8&q=CGI+tutorial direkt im Browser ist eine gute Möglichkeit, die Wirkung des CGI-Skripts auf dem Google-Server zu betrachten.

13.9 Dokument-Aufbereitung

13.9.1 Das `roff`-Satzsystem

Traditionell ist `roff` das wichtigste Unix-Textbearbeitungssystem.

Vergleichen Sie `roff(7)`, `groff(7)`, `groff(1)`, `grotty(1)`, `troff(1)`, `groff_mdoc(7)`, `groff_man(7)`, `groff_ms(7)`, `groff_me(7)`, `groff_mm(7)` und `info groff`.

Es gibt eine gute Einführung zu `-me` Makros. Haben Sie `groff` (1.18 oder neuer), suchen Sie `/usr/share/doc/groff/meintro.me.gz` und führen Sie das Folgende aus:

```
$ zcat /usr/share/doc/groff/meintro.me.gz | \
     groff -Tascii -me - | less -R
```

Das Folgende erzeugt eine einfache Textdatei:

```
$ zcat /usr/share/doc/groff/meintro.me.gz | \
    GROFF_NO_SGR=1 groff -Tascii -me - | col -b -x > meintro.txt
```

[1] `%nn` wird hier für das kodierte Zeichen mit dem Hexadezimalwert `nn` verwendet.

Verwenden Sie die PostScript-Ausgabe zum Drucken.

```
$ groff -Tps meintro.txt | lpr
$ groff -Tps meintro.txt | mpage -2 | lpr
```

13.9.2 SGML

Vorbereitung:

```
# apt-get install debiandoc-sgml debiandoc-sgml-doc
```

Referenzen für `debiandoc-sgml`:

- `/usr/share/doc/debiandoc-sgml-doc`
- `debiandoc-sgml(1)`
- *DocBook: The Definitive Guide* (*`/usr/share/doc/docbook-defguide/html/docbook.html`*) von Walsh und Muellner (O'Reilly) (Paket `docbook-defguide`)

SGML ermöglicht das Verwalten mehrerer Dokumentformate. Ein einfaches SGML-System ist Debiandoc, was hier verwendet wird. Dies erfordert kleinere Umwandlungen der Textdateien für die folgenden Buchstaben:

- ″<“ –> `<`
- ″>“ –> `>`
- ″ “ –> ` ` (nicht trennbares Leerzeichen)
- ″&“ –> `&`
- ″%“ –> `%`
- ″©“ –> `©`
- ″–“ –> `–`
- ″—“ –> `—`

Um einen Abschnitt als nicht ausgebbaren Kommentar zu markieren, wird Folgendes verwendet:

```
<!-- Darlegen des Sachverhalts ... -->
```

Für einen Abschnitt mit abschaltbarem Kommentar:

```
<![ %FIXME; [ Darlegen des Sachverhalts ... ]]>
```

In SGML gewinnt die *erste Definition* einer Entität. Zum Beispiel:

```
<!entity % qref "INCLUDE">
<![ %qref; [ <!entity param "Daten 1"> ]]>
<!entity param "Daten 2">
&param;
```

Dies endet mit "Daten 1". Enthält die erste Zeile "IGNORE" anstatt "INCLUDE", endet es mit "Daten 2" (die zweite Zeile ist eine bedingte Anweisung). Es ist auch möglich, einzelne wiederholende Textabschnitte separat vom Inhalt zu definieren.

```
<!entity wessen "mein">
Hallo &wessen; Freund.
Dies ist &wessen; Buch.
```

Dies ergibt das Folgende:

```
Hallo mein Freund.
Dies ist mein Buch.
```

Vergleichen Sie mit dem kurzen SGML-Beispiel `sample.sgml` im Beispielverzeichnis (`http://www.debian.org/doc/manuals/debian-reference/examples/`).

Wenn SGML-Dokumente größer werden, hat TeX, das intern zur Textbearbeitung verwendet wird, manchmal Probleme. Vergleichen Sie 'TeX/LaTeX' auf dieser Seite.

13.9.3 TeX/LaTeX

Vorbereitung:

```
# tasksel # wählen Sie Miscellaneous  --> TeX/LaTeX-Umgebung
```

Referenzen für LaTeX:
- The teTeX HOWTO: The Linux-teTeX Local Guide (`http://www.tldp.org/HOWTO/TeTeX-HOWTO.html`)
- `tex(1)`
- `latex(1)`
- *The TeXbook* von Donald E. Knuth (Addison-Wesley) [2]
- *Das LaTeX-Handbuch* von Leslie Lamport (Addison-Wesley)
- *Der LaTeX-Begleiter* von Goossens, Mittelbach, Samarin (Addison-Wesley)

Dies ist das mächtigste Textsatzprogramm. Viele SGML-Bearbeitungsprogramme verwenden dies zur Textbearbeitung im Hintergrund. Lyx und GNU-TeXmacs, die mit den Paketen `lyx`, `lyx-xforms` und `lyx-qt` bzw. `texmacs` bereitgestellt werden, bieten eine nette WYSIWYG-Umgebung zum Editieren für LaTeX, zu der viele Emacs und Vim als Quellcode-Editor wählen.

Es gibt viele Online-Verweise:

[2] Der TeX-Quellcode dieses Buches ist unter `ftp://ftp.dante.de/pub/tex/systems/knuth/tex/texbook.tex` erhältlich. `ftp://ftp.dante.de/pub/tex/systems/knuth/lib/manmac.tex` enthält die meisten der benötigten Makros. Das Dokument kann mit `tex` bearbeitet werden, nachdem die Zeilen 7 bis 10 auskommentiert und `\input manmac \proofmodefalse` hinzugefügt wurde. Es ist sehr empfehlenswert dieses Buch (und alle anderen Bücher von Donald E. Knuth) zu kaufen, statt die Online-Version zu verwenden, aber der Quellcode ist ein sehr gutes Beispiel für TeX-Code!

- teTeX - A Documentation Guide (/usr/share/doc/texmf/newhelpindex.html) (tetex-doc Paket)
- LaTeX2e-Kurzbeschreibung (/usr/share/doc/texmf/latex/general/l2kurz.dvi.gz)
- Viele weitere deutschsprachige Dokumente zu LaTeX (ftp://ftp.dante.de/tex-archive/info/german/) (u.a. *Das kleine TeX-Buch*, eine deutscher Auszug aus *The TeXbook*)
- A Quick Introduction to LaTeX (http://www.msu.edu/user/pfaffben/writings/)
- A Simple Guide to Latex/Lyx (http://www.stat.rice.edu/~helpdesk/howto/lyxguide.html)
- Word Processing Using LaTeX (http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/latex_basic/latex_basic.html)
- Local User Guide to teTeX/LaTeX (http://supportweb.cs.bham.ac.uk/documentation/LaTeX/lguide/local-guide/local-guide.html)

Wenn Dokumente größer werden, hat TeX manchmal Probleme. Um dies zu korrigieren, muss die Pool-Größe in /etc/texmf/texmf.cnf erhöht werden (oder editieren Sie besser /etc/texmf/texmf.d/95NonPath und starten Sie update-texmf).

13.9.4 Literate Programming

Anstatt Code zu schreiben, der Dokumentationen enthält, schreibt der gebildete Programmierer (literate programmer) Dokumentation die Code enthält. Dieser Ansatz sichert eine gute Dokumentation für ein Programm.

Für weitere Informationen zu literate-programming wird auf Literate Programming (http://www.literateprogramming.com/) verwiesen.

Noweb

Vorbereitung:

```
# apt-get install nowebm
```

Referenzen für Noweb:

- Noweb — A Simple, Extensible Tool for Literate Programming (http://www.eecs.harvard.edu/~nr/noweb/)
- noweb(1)

Dies ist ein WEB-artiges literate-programming Werkzeug, das einfacher ist, sowie erweiterbar und sprachunabhängig. [3] Wenn noweb gestartet wird, schreibt es den Programmquellcode in Ausgabedateien, die in der noweb-Datei angegeben sind, und es erstellt eine TeX-Datei für die Dokumentation.

[3] Dieses WEB hat **nichts** zu tun mit dem World Wide Web. WEB (für PASCAL) und CWEB (für C/C++) sind traditionelle literate-programming Werkzeuge.

Das Debian-Paket `ifupdown` ist ein gutes Beispiel.

```
$ apt-get source ifupdown
$ cd ifupdown*
$ make ifupdown.pdf ifupdown.ps
```

Doxygen

Vorbereitung:

```
# apt-get install doxygen doxygen-doc doxygen-gui
```

Referenzen für Doxygen (von `doxygen` erstellt!):

- Homepage (`http://www.doxygen.org/`)
- `/usr/share/doc/doxygen-doc/html/index.html`

Es kann HTML-, RTF-, Unix-Handbuchseiten-, PostScript- und PDF- (durch LaTeX) Dokumentation für C++-, C-, Java-, IDL- und zum Teil PHP- sowie C#-Programme erstellen. Doxygen ist kompatibel zu JavaDoc (1.1), Qt-Doc, KDOC und wurde speziell entworfen, um Projekte die Troll Tech's Qt (`http://www.trolltech.no/qt/`) Bibliothek nutzen, zu unterstützen. Es erstellt Include-Abhängigkeitsdiagramme, Diagramme zu Zusammengehörigkeiten und grafische Darstellungen der Klassenhierarchien sogar für nicht dokumentierte Programme. Die Ausgabe ist ähnlich zu Qt's Dokumentation.

13.10 Paketerzeugung

Vorbereitung:

```
# apt-get install debian-policy developers-reference \
                  maint-guide dh-make debhelper
# apt-get install packaging-manual # für Potato
```

Referenzen für die Paketerzeugung:

- 'Das Debian-Paketverwaltungssystem' auf Seite 11 (Grundlagen)
- Debian New Maintainers' Guide (Tutorial)
- `dh-make(1)`
- Debian Developer's Reference (sehr praxisnah)
- Debian Policy Manual (die ultimative Dokumentation)
- Packaging Manual (Potato)

13.10.1 Paketerzeugung für ein einzelnes Programm

Eine schnelle und unsaubere Methode der Paketerzeugung für ein einzelnes Programm ist nach Joey Hess:

```
# mkdir -p mypkg/usr/bin mypkg/DEBIAN
# cp binary mypkg/usr/bin
# cat > mypkg/DEBIAN/control
Package: mypackage
Version: 1
Architecture: i386
Maintainer: Joey Hess <joeyh@debian.org>
Description: my little package
 Don't expect much.
^D
# dpkg-deb -b mypkg
```

13.10.2 Paketerzeugung mit Tools

Verwenden Sie `dh_make` aus dem `dh-make`-Paket, um eine solide Grundlage für ein Paket zu gewinnen. Folgen Sie danach den Anweisungen in `dh-make(1)`. Dies nutzt `debhelper` in `debian/rules`.

Ein älterer Zugang ist das Verwenden von `deb-make` aus dem `debmake`-Paket. Dies nutzt keine `debhelper`-Skripte und hängt nur von der Shell ab. Verwenden Sie dies nicht länger.

Für Beispiele von Paketen mit mehreren Quelldateien vergleichen Sie mit "mc" (`dpkg-source -x mc_4.5.54.dsc`), das "sys-build.mk" von Adam Heath (`<doogie@debian.org>`) nutzt und "glibc" (`dpkg-source -x glibc_2.2.4-1.dsc`), das ein anderes System von Joel Klecker (`<espy@debian.org>`) nutzt.

Kapitel 14

GnuPG

Referenzen:

- gpg(1).
- /usr/share/doc/gnupg/README.gz
- *GNU privacy handbook* in /usr/share/doc/gnupg-doc/GNU_Privacy_Handbook/ (installieren Sie das gnupg-doc-Paket)

14.1 Installation von GnuPG

```
# gpg --gen-key                  # erzeugen eines neuen Schlüssels
# gpg --gen-revoke Nutzer_ID     # erzeugen eines Widerrufungs-
                                 # schlüssels für Nutzer_ID
# host -l pgp.net |grep www |less# pgp Schlüsselserver bestimmen
```

Ein guter Standard-Schlüsselserver sollte in $HOME/.gnupg/gpg.conf (oder der alten Datei $HOME/.gnupg/options) eingetragen sein:

```
keyserver hkp://subkeys.pgp.net
```

Hier muss man vorsichtig sein **nicht** mehr als 2 Unterschlüssel zu erzeugen. Sollten Sie dies tun, werden die Schlüsselserver auf pgp.net Ihren Schlüssel **korrumpieren**. Nutzen Sie das neuere gnupg (>1.2.1-2), um mit diesen korrupten Unterschlüssel umzugehen. Vergleichen Sie http://fortytwo.ch/gpg/subkeys.

14.2 Verwendung von GnuPG

Handhabung von Dateien:

```
$ gpg [Optionen] Kommando [args]
$ gpg {--armor|-a} {--sign|-s} Datei # signiere Datei in Datei.asc
$ gpg --clearsign Datei              # clear-sign Nachricht
$ gpg --clearsign --not-dash-escaped Patchdatei # clear-sign
                                     # Patchdatei
$ gpg --verify Datei                 # überprüfe clear-signed Datei
$ gpg -o Datei.sig {-b|--detach-sig} Datei # erzeuge losgelöste
                                     # Signatur
$ gpg --verify Sign Datei            # überprüfe Datei mit Sign
$ gpg -o crypt_Datei {--recipient|-r} name {--encrypt|-e} Datei
                             # Public-Key-Verschlüsselung für name
$ gpg -o crypt_Datei {--symmetric|-c} Datei # sym. Verschlüsselung
$ gpg -o Datei --decrypt crypt_Datei # Entschlüsselung
$ gpg --refresh-keys                   # Update local keyring
```

14.3 Umgang mit GnuPG

Schlüsselmanagement:

```
$ gpg --edit-key Nutzer_ID   # "help" für Hilfe, interaktiv
$ gpg -o Datei --exports     # exportiere alle Schlüssel nach Datei
$ gpg --imports Datei        # importiere alle Schlüssel von Datei
$ gpg --send-keys Nutzer_ID  # sende Schlüssel von Nutzer_ID zum
                             # Schlüsselserver
$ gpg --recv-keys Nutzer_ID  # empfange Schlüssel von Nutzer_ID
                             # vom Schlüsselserver
$ gpg --list-keys Nutzer_ID  # liste Schlüssel von Nutzer_ID auf
$ gpg --list-sigs Nutzer_ID  # liste Signatur von Nutzer_ID auf
$ gpg --check-sigs Nutzer_ID # überprüfe Signatur von Nutzer_ID
$ gpg --fingerprint Nutzer_ID# überprüfe Fingerabdruck von Nutzer_ID
$ gpg --list-sigs | grep '^sig' | grep '[User id not found]' \
  | awk '{print $2}' | sort -u | xargs gpg --recv-keys
                             # unbekannte Schlüssel besorgen
  # aktualisiere Schlüssel für alle unbekannten Signaturen.
$ gpg --refresh-keys         # aktualisiere den lokalen Schlüsselring
```

Vertrauenscode:

```
-         Kein Besitzervertrauen zugewiesen / noch nicht berechnet.
e         Vertrauensberechnung schlug fehl.
q         Nicht genug Informationen für Berechnung.
n         Niemals diesem Schlüssel vertrauen.
m         Gerade noch vertrauenswert.
```

```
f          Voll vertrauenswert.
u          Äußerst vertrauenswert.
```

Das Folgende lädt meinen Schlüssel *"A8061F32"* auf den bekannten Schlüsselserver `hkp://subkeys.pgp.net`:

```
$ gpg --keyserver hkp://subkeys.pgp.net --send-keys A8061F32
```

14.4 Benutzung von GnuPG mit Anwendungen

14.4.1 Benutzung von GnuPG mit Mutt

Fügen Sie Folgendes zu `~/.muttrc` hinzu, um das langsame GnuPG vom automatischen Starten abzuhalten, während es beim Drücken von '`S`' im Indexmenü gestartet werden kann.

```
macro index S ":toggle pgp_verify_sig\n"
set pgp_verify_sig=no
```

14.4.2 Benutzung von GnuPG mit Vim

Fügen Sie den Inhalt von `_vimrc` aus dem Beispielverzeichnis (`http://www.debian.org/doc/manuals/debian-reference/examples/`) in `~/.vimrc` ein, um GnuPG transparent zu starten.

Kapitel 15

Unterstützung für Debian

Die folgenden Ressourcen bieten Hilfe, Rat und Unterstützung für Debian an. Versuchen Sie Ihr Bestes bei der Suche nach Hilfe bevor Sie sich an die Mailinglisten wenden. :)

Auf eine Vielzahl von Dokumentationen in Ihrem System kann mittels eines WWW Browsers und den `dwww` oder `dhelp` Kommandos zugegriffen werden, die in den entsprechenden Paketen gefunden werden können.

15.1 Referenzen

Die folgenden Referenzen sind generell für Debian und Linux verfügbar. Wenn der Inhalt sich widerspricht, vertrauen Sie mehr den primären Informationsquellen als sekundären wie diesem Dokument.

- Installationshandbuch (primär)
  - Lesen Sie es bevor Sie installieren und upgraden.
  - Web: `http://www.debian.org/releases/stable/installmanual`
  - Web: `http://www.debian.org/releases/testing/installmanual` (befindet sich in Arbeit und ist manchmal eventuell nicht vorhanden)
  - Paket: `Not available in install-doc: Bug#155374`
  - Datei: `DebianCDunder/doc/`
- Release Notes (primär)
  - Dies müssen Sie vor der Installation und einem Upgrade unbedingt lesen, selbst wenn Sie erfahren sind.
  - Web: `http://www.debian.org/releases/stable/releasenotes`
  - Web: `http://www.debian.org/releases/testing/releasenotes` (befindet sich in Arbeit und ist manchmal eventuell nicht vorhanden)
  - Paket: `Not available in install-doc: Bug#155374`
  - Datei: `DebianCDunder/doc/`
- FAQ (sekundär)

  - Oft gestellte Fragen (FAQ, Frequently asked questions)
  - Web: `http://www.debian.org/doc/manuals/debian-faq/`
  - Paket: `doc-debian`
  - Datei: `/usr/share/doc/debian/FAQ/index.html`

- Debian-Referenz (sekundär)
  - Das umfassendste Benutzerhandbuch für die Zeit nach der Installation
  - Web: `http://www.debian.org/doc/manuals/debian-reference/`
  - Paket: `debian-reference-de`
  - Datei: `/usr/share/doc/Debian/reference/`

- APT HOWTO (sekundär)
  - Detailliertes Benutzerhandbuch für das Debian-Paketmanagement. (Woody)
  - Web: `http://www.debian.org/doc/manuals/apt-howto/`
  - Paket: `apt-howto`
  - Datei: `/usr/share/doc/Debian/apt-howto/`

- Debian-Sicherheitshandbuch (sekundär)
  - Detailliertes Benutzerhandbuch zur Sicherung und Härtung der Standard Debian Installation. (Woody)
  - Web: `http://www.debian.org/doc/manuals/securing-debian-howto/`
  - Paket: `harden-doc`
  - Datei: `/usr/share/doc/harden-doc/html/securing-debian-howto/`

- `dselect` Dokumentation für Anfänger (sekundär)
  - Tutorial für `dselect`
  - Web: `http://www.debian.org/releases/woody/i386/dselect-beginner`
  - Paket: `Not available in install-doc: Bug#155374`
  - Datei: `DebianCDunder/doc/`

- Debian Policy Manual (primär)
  - Das technische Rückgrat von Debian.
  - Web: `http://www.debian.org/doc/debian-policy/`
  - Paket: `debian-policy`
  - Datei: `/usr/share/doc/debian-policy/`

- Debian-Referenz für Entwickler (primär)
  - Grundwissen für Entwickler.
  - Die anderen sollten dies auch durchblättern.
  - Web: `http://www.debian.org/doc/manuals/developers-reference/`
  - Paket: `developers-reference`
  - Datei: `/usr/share/doc/developers-reference/`

- Debian-Handbuch für neue Betreuer (primär)
  - Praktische Anleitung für Entwickler.
  - Tutorials zur Paketierung für den Rest von uns.

  - Web: `http://www.debian.org/doc/manuals/maint-guide/`
  - Paket: `maint-guide`
  - Datei: `/usr/share/doc/maint-guide/`

- Paketierungshandbuch (Potato)
  - `packaging-manual`-Paket in Potato. (Wurde in den Anhang der *Referenz für Entwickler* verschoben.)

- Unix-Handbuchseiten (primär)
  - `dlocate -man` *`Paket`* (zeigt verfügbare Seiten)
  - `man` *`Abschnitt Kommando`*

- GNU-info-Seiten (primär)
  - `info` (Zugriff auf oberste Hierarchie)
  - `info` *`Kommando`*

- Paketspezifische Dokumente (primär)
  - Diese sind unter `/usr/share/doc/`*`Paket`* zu finden

- LDP: Linux Dokumentations Projekt (sekundär)
  - Allgemeine Linux HOWTOs und mini-HOWTOs
  - Web: `http://www.tldp.org/`
  - Paket: `doc-linux-text` und `doc-linux-html`
  - Datei: `/usr/share/doc/HOWTO/`

- Linux Gazette (sekundär) – monatlich neue Ausgaben
  - Linux Gazette
  - Web: `http://www.linuxgazette.com/`
  - Paket: `lg-all` oder `lg-latest-two`
  - Datei: `/usr/share/doc/lg/`

- DDP: Debian-Dokumentations-Projekt (sekundär)
  - Debian-spezifische Handbücher
  - Web: `http://www.debian.org/doc/`

- Debian-Entwicklerecke (sekundär)
  - Schlüsselinformationen für Debian-Entwickler
  - Einsichtig für Endbenutzer
  - Web: `http://www.debian.org/devel/`

- Quellcode (absolut primär)
  - Niemand kann dies bestreiten :-)
  - Laden Sie den Quellcode entsprechend 'Der Quellcode' auf Seite 11 herunter

- Internet Assigned Numbers Authority (primär)
  - Web: `http://www.iana.org/`
  - Paket: `doc-iana`
  - Datei: `/usr/share/doc/doc-iana/`

- Internet requests for comments (IETF Standards) (primär)
  - Web: `http://www.ietf.org/rfc.html`
  - Paket: `doc-rfc`
  - Datei: `/usr/share/doc/RFC/`

Die folgenden Referenzen sind für Unix im Allgemeinen verfügbar. Beachten Sie bitte, dass es einige kleinere Unterschiede zwischen verschiedenen Unix Systemen gibt. Gerätenamen und Initialisierungsmethoden bedürfen spezieller Beachtung.

- *The UNIX Programming Environment*
  - Das Buch das erklärt wie UNIX arbeitet.
  - Von B. W. Kernighan und R. Pike
  - Veröffentlicht von Princeton Hall Software Series

- *The C Programming Language* (zweite Ausgabe)
  - Das Buch über ANSI C.
  - Von B. W. Kernighan und D. M. Ritchie
  - Veröffentlicht von Princeton Hall Software Series

- *UNIX Power Tools*
  - Das Buch zu Unix Tipps.
  - Von Jerry Peek, Tim O'Reilly und Mike Loukides
  - Veröffentlicht von O'Reilly and Associates

- *Essential System Administration* (zweite Ausgabe)
  - Das Buch über Unix Systemadministration für viele Unixarten.
  - Von Aeleen Frisch
  - Veröffentlicht von O'Reilly and Associates

- *Linux: Rute User's Tutorial and Exposition*
  - Ein nettes Online-Buch das die GNU/Linux-Systemadministration behandelt und als gedrucktes Buch verfügbar ist.
  - Von Paul Sheer
  - Veröffentlicht von Prentice Hall
  - Paket: `rutebook` (in `non-free`)
  - Datei: `/usr/share/doc/rutebook/`

- Bell Labs: Computing Sciences Research
  - Reichhaltiges Archiv der Unix Geschichte
  - Main: `http://cm.bell-labs.com/cm/cs/`
  - Ausgewählte technische Berichte: `http://cm.bell-labs.com/cm/cs/cstr.html`
  - Einige Schriftstücke: `http://cm.bell-labs.com/cm/cs/papers.html`

- Allgemeine online-Ressourcen zur Linux-Unterstützung
  - Debian-Planet (`http://www.debianplanet.org/`)

  - debianHELP (`http://www.debianhelp.org/`)
  - Linux.com (`http://linux.com/`)
  - Die Linux Homepage auf Linux Online (`http://www.linux.org/`)
  - Red Hat (kommerzieller Linux Verkäufer) (`http://www.redhat.com/`) (RPM, Sys-V-Initialisierung)
  - SuSE, Inc. (kommerzieller Linux Verkäufer) (`http://www.suse.de/`) (RPM, Sys-V-Initialisierung)
  - Slackware (`http://www.slackware.com/`) (TGZ, BSD-artige Initialisierung)

- Allgemeine online Ressourcen und Unix-Anleitungen
  - Das UNIX-System von The Open Group (`http://www.unix.org/`)
  - Ein UNIX-Einführungskurs der Ohio State University (`http://www-wks.acs.ohio-state.edu/unix_course/unix.html`)
  - UNIXhelp der University of Edinburgh (`http://unixhelp.ed.ac.uk/`)
  - Unix / Programmierungsinformationen (`http://arioch.unomaha.edu/~jclark/#info`)
  - comp.unix.questions FAQ (`http://www.faqs.org/faqs/unix-faq/faq/`)
  - comp.unix.user-friendly FAQ (`http://www.camelcity.com/~noel/usenet/cuuf-FAQ.htm`)
  - FreeBSD Dokumentation (`http://www.freebsd.org/docs.html`)
  - Das FreeBSD Handbuch (`http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/index.html`)
  - UNIX ANLEITUNG (`http://ieee.uow.edu.au/documents/`)
  - Verein des Unix-Erbes (`http://www.tuhs.org/`)

- Homepages freier Software-Projekte
  - GNU Projekt (`http://www.gnu.org/`)
  - Das Linux Dokumentations Projekt (`http://www.tldp.org/`)
  - Die Linux Kernel Archive (`http://www.linux.org/`)
  - Das XFree86 Projekt, Inc (`http://www.xfree86.org/`)
  - GNOME (`http://www.gnome.org/`)
  - K Desktop Environment (`http://www.kde.org/`)
  - GNU Software bei Red Hat (`http://sources.redhat.com/`)
  - Mozilla (`http://www.mozilla.org`)
  - FreeBSD (`http://www.freebsd.org/`)
  - OpenBSD (`http://www.openbsd.org/`)
  - NetBSD (`http://www.netbsd.org/`)

15.2 Finden der Bedeutung eines Wortes

Viele in Debian verwendete Worte sind kryptisch oder Abkürzungen. Das Folgende beantwortet die meisten Fragen (benötigt `dict` und verwandte Pakete):

```
$ dict sonderbares-Wort
```

15.3 Bestimmen der Popularität eines Debian-Pakets

Viele Pakete existieren in Debian und es ist manchmal schwierig zu wissen, was man zuerst ausprobieren sollte. Vergleichen Sie die Ergebnisse des Debian Popularitäts Wettbewerbs (`http://popcon.debian.org/`), für einen Überblick was andere nutzen. Installieren Sie das Paket `popularity-contest`, um dazu beizutragen.

15.4 Das Debian-Fehlerverfolgungssystem

Die Debian-Distribution hat ein Fehlerverfolgungssystem (`http://bugs.debian.org/`) (bug tracking system, BTS), das Details von Fehlerberichten enthält, die von Nutzern und Entwicklern gemeldet wurden. Jedem Fehler ist eine Nummer zugeordnet, und wird solange aufbewahrt, bis er als erledigt markiert wurde.

Sie sollten überprüfen, ob Ihr Fehlerbericht bereits von jemand anderem eingereicht wurde, bevor Sie ihn übertragen. Listen von aktuell ausstehenden Fehlern sind im World Wide Web (`http://bugs.debian.org/`) und sonstwo (`http://www.debian.org/Bugs/Access`) verfügbar. Vergleichen Sie auch 'Überprüfen von Bugs in Debian und Suche nach Hilfe' auf Seite 70.

Es sind viele Berichte zu veröffentlichungskritischen Fehlern mit **FTBFS** markiert. Dies bedeutet "kann nicht aus dem Quellcode erzeugt werden" ("Fails To Build From Source").

Die Methode zum Berichten von Fehlern wird unter `http://www.debian.org/Bugs/Reporting` beschrieben.

15.5 Mailinglisten

Lesen Sie zumindest "debian-devel-announce" (englisch, nur lesbar und geringes Aufkommen), um mit Debian auf dem Laufenden zu bleiben.

Die für Debian-Nutzer am interessantesten Mailinglisten sind "debian-user" (englisch, offen und hohes Aufkommen) und andere "debian-user-*Sprache*" Listen (für andere Sprachen).

Für Informationen zu diesen Listen und Details zur Abonnierung vergleichen Sie `http://lists.debian.org/`. Bitte überprüfen Sie die Archive für Antworten zu Ihren Fragen bevor Sie Anfragen stellen und achten Sie auf für die Liste geeignete Umgangsformen.

Wenn Sie keine Antwort auf Ihre E-Mails an die Mailingliste auf Grund eines `CC:` (Kopie an:) Headers erhalten wollen, sollten Sie den `Mail-Followup-To:` Header verwenden, der sehr effektiv ist. Dies ist die nicht formelle Konvention von Mailinglisten wie in `http://cr.yp.to/proto/replyto.html` erklärt wird.

15.6 Internet Relay Chat (IRC)

IRC (Internet Relay Chat) ist ein Weg um sich mit Personen in der gesamten Welt in Echtzeit zu unterhalten. IRC Kanäle die sich Debian widmen können im freenode (`http://www.freenode.info/`) IRC Netzwerk gefunden werden. Zur Verbindung benötigen Sie einen IRC Client. Einige der populärsten Clients sind XChat, BitchX, ircII, irssi, epic4 und KSirc, all diese sind für Debian paketiert. Haben Sie den Client installiert, müssen Sie ihn auffordern, sich mit dem Server zu verbinden. In vielen Clients können Sie das durch Eingabe von

```
/server irc.debian.org
```

erreichen. Sind Sie verbunden, öffnen Sie den #debian Kanal mit

```
/join #debian
```

Zum Verlassen des Kanals #debian geben Sie

```
/part #debian
```

und zum Beenden des IRC-Clients

```
/quit
```

ein. Mit

```
/msg foo Hallo Fred
```

senden Sie die Mitteilung "Hallo Fred" an *foo*. Beachten Sie das alles, was Sie ohne vorangestelltes `/` eingeben, an den Kanal als Mitteilung geschickt wird.

Bemerkung: Clients wie XChat haben oft eine unterschiedliche graphische Benutzerschnittstelle zum Verbinden mit Servern/Kanälen.

15.7 Suchmaschinen

Es gibt viele Suchmaschinen, die Dokumentationen zu Debian anbieten:

- Debian WWW Suchseite (`http://search.debian.org/`).
- Google (`http://www.google.com/`): fügen Sie "site:debian.org" als einen Suchbegriff ein.

- Google Groups (http://groups.google.com/): eine Suchmaschine für Newsgroups. Fügen Sie "group:linux.debian.*" als Suchbegriff ein.
- AltaVista (http://www.altavista.com/)

Eine Suche nach "cgi-perl" ergibt umfangreichere Erklärungen zu diesem Paket als die kurze Beschreibung in der Kontrolldatei. Vergleichen Sie 'Überprüfen von Bugs in Debian und Suche nach Hilfe' auf Seite 70 für ähnliche Ratschläge.

15.8 Webseiten

Es folgen einige URLs, die ich für spezielle Probleme sammelte.

- IBM developerWorks: Linux (http://www.ibm.com/developerworks/linux/)
- Adrian Bunks aktuellste Pakete (Backports nach stable) (http://www.fs.tum.de/~bunk/)
- Linux auf Laptops (http://www.linux-laptop.net/)
- Xterm-FAQ (http://dickey.his.com/xterm/xterm.faq.html)
- EXT3-Dateisystem mini-HOWTO (http://www.zip.com.au/~akpm/linux/ext3/ext3-usage.html)
- Unterstützung für große Dateien in Linux (http://www.suse.de/~aj/linux_lfs.html)
- Window Manager für X (http://www.xwinman.org)
- Linux-USB-Projekt (http://www.linux-usb.org/)
- SuSE-Seiten für CJK (http://www.suse.de/~mfabian/suse-cjk/suse-cjk.html)
- LNX-BBC (Das visitenkartengroße Boot-CD-Projekt) (http://www.lnx-bbc.org/)
- Linux-Informationen von Karsten Self (Partitionierung, Backup, Browser, ...) (http://kmself.home.netcom.com/Linux/index.html)
- Backup-Informationen-HOWTO von Alvin Oga (http://www.Linux-Backup.net/)
- Sicherheitsinformationen-HOWTO von Alvin Oga (http://www.Linux-Sec.net/)
- Verchiedene INOFFIZIELLE Quellen für APT (http://www.apt-get.org/)
- Laptop-Ethernet-Konfiguration (http://www.orthogony.com/gjw/lap/lap-ether-intro.html)

Anhang A

Anhang

A.1 Autoren

Die Debian-Referenz wurde ursprünglich von Osamu Aoki <osamu\#at\#debian.org> als eine persönliche Installationsanleitung geschrieben, die ”Schnellreferenz . . . “ hieß. Ein großer Teil des Inhalts kam vom Archiv der ”debian-user“-Mailingliste. Auch ”Debian Installation Manual“ und ”Debian Release Notes“ waren Referenzen.

Einem Vorschlag von Josip Rodin folgend, der sehr aktiv im Debian-Dokumentations-Projekt (`http://www.debian.org/doc/ddp`) (DDP) ist und der der aktuelle Betreuer der ”Debian-FAQ“ ist, wurde dieses Dokument in ”Debian-Referenz“ umbenannt und einige Kapitel der ”Debian-FAQ“ mit verweisartigem Inhalt hinzugefügt. Danach wurde ”Debian-Schnellreferenz“ als ein Auszug geschaffen.

Dieses Dokument wurde geschrieben, übersetzt und erweitert durch die folgenden QREF-Teammitglieder:

- Englisches Original für die ursprüngliche ”Schnellreferenz . . . “
  - Osamu Aoki <osamu\#at\#debian.org> (Koordinator des gesamten Inhalts)
- Englisches Korrekturlesen und zusätzliche Beiträge
  - David Sewell <dsewell\#at\#virginia.edu> (umfangreiche Arbeit am Stil)
  - Thomas Hood <jdthood\#at\#yahoo.co.uk> (Netzwerk-bezogen)
  - Brian Nelson <nelson\#at\#bignachos.com> (besonders X-bezogen)
  - Jan Michael C Alonzo <jmalonzo\#at\#spaceants.net>
  - Daniel Webb <webb\#at\#robust.colorado.edu>
  - Rückmeldungen von allen Übersetzern
- Französische Übersetzung
  - Guillaume Erbs <gerbs\#at\#free.fr> (Koordinator fr)
  - Rénald Casagraude <rcasagraude\#at\#interfaces.fr>
  - Jean-Pierre Delange <adeimantos\#at\#free.fr>
  - Daniel Desages <daniel\#at\#desages.com>
- Italienische Übersetzung
  - Davide Di Lazzaro <mc0315\#at\#mclink.it> (Koordinator it)
- Portugiesische Übersetzung (brasilianisches Portugiesisch)

  - Paulo Rogério Ormenese <pormenese\#at\#uol.com.br> (Koordinator pt-br)
  - Andre Luis Lopes <andrelop\#at\#ig.com.br>
  - Marcio Roberto Teixeira <marciotex\#at\#pop.com.br>
  - Rildo Taveira de Oliveira <to_rei\#at\#yahoo.com>
  - Raphael Bittencourt Simoes Costa <raphael-bsc\#at\#bol.com.br>
  - Gustavo Noronha Silva <kov\#at\#debian.org> (Koordinator)
- Spanische Übersetzung
  - Walter Echarri <wecharri\#at\#infovia.com.ar> (Koordinator es)
  - José Carreiro <ffx\#at\#urbanet.ch>
- Deutsche Übersetzung
  - Jens Seidel <tux-master\#at\#web.de> (Koordinator de)
  - Willi Dyck <wdyck\#at\#gmx.net>
  - Stefan Schröder <stefan\#at\#fkp.uni-hannover.de>
  - Agon S. Buchholz <asb\#at\#kefk.net>
- Polnische Übersetzung – Mitglieder von PDDP (http://debian.linux.org.pl):
  - Marcin Andruszkiewicz
  - Mariusz Centka <mariusz.centka\#at\#debian.linux.org.pl>
  - Bartosz Feński <fenio\#at\#debian.linux.org.pl> (Koordinator pl)
  - Radosław Grzanka <radekg\#at\#debian.linux.org.pl>
  - Bartosz ’Xebord’ Janowski
  - Jacek Lachowicz
  - Rafał Michaluk
  - Leonard Milcin, Jr.
  - Tomasz Z. Napierała <zen\#at\#debian.linux.org.pl>
  - Oskar Ostafin <cx\#at\#debian.linux.org.pl>
  - Tomasz Piękoś
  - Jacek Politowski
  - Mateusz Prichacz <mateusz\#at\#debian.linux.org.pl>
  - Marcin Rogowski
  - Paweł Różański
  - Mariusz Strzelecki
  - Krzysztof Ścierski
  - Przemysław Adam Śmiejek <tristan\#at\#debian.linux.org.pl>
  - Krzysztof Szynter
  - Mateusz Tryka <uszek\#at\#debian.linux.org.pl>
  - Cezary Uchto
  - Krzysztof Witkowski <tjup\#at\#debian.linux.org.pl>
  - Bartosz Zapałowski <zapal\#at\#debian.linux.org.pl>
- Chinesische Übersetzung (vereinfachtes Chinesisch)
  - Hao ”Lyoo“ LIU <iamlyoo\#at\#163.net>
  - Ming Hua <minghua\#at\#rice.edu>
  - Xiao Sheng Wen <atzlinux\#at\#163.com> (Koordinator zh-cn)
  - Haifeng Chen <optical.dlz\#at\#gmail.com>
  - Xie Yanbo <xieyanbo\#at\#gmail.com>
  - easthero <easthero\#at\#gmail.com>
- Chinesische Übersetzung (traditionelles Chinesisch)

- Asho Yeh <asho\#at\#debian.org.tw> (Koordinator zh-tw)
- Tang Wei Ching <wctang\#at\#csie.nctu.edu.tw> (ehemaliger Koordinator zh-tw)

- Japanische Übersetzung
  - Shinichi Tsunoda <tsuno\#at\#ngy.1st.ne.jp> (Koordinator ja)
  - Osamu Aoki <osamu\#at\#debian.org>

QREF ist die Kurzform für den ursprünglichen Dokumenttitel "Quick Reference. . . " ("Schnellreferenz . . . ") und ist auch der Projektname auf qref.sourceforge.net.

Viele Handbuch- und Info-Seiten des Debian-Systems wurden als primäre Referenzen zum Schreiben dieses Dokuments verwendet. Neben der Verwendung von gekennzeichneten Zitaten hat Osamu Aoki große Teile davon, insbesondere Befehlsdefinitionen, in den Text mit eingebunden, nachdem diese sorgfältig dem Stil als auch der Zielsetzung dieses Dokuments entsprechend, angepasst wurden.

Der Großteil des Inhalts von 'Debian-Grundlagen' auf Seite 5 kam ursprünglich von der "Debian-FAQ" (März 2002):

- 5. Die Debian-FTP-Archive: `ftparchives.sgml` (gesamtes Kapitel)
- 6. Grundlagen des Debian-Paketmanagement-Systems: `pkg_basics.sgml` (gesamtes Kapitel)
- 7. Die Debian-Paketmanagement-Tools: `pkgtools.sgml` (gesamtes Kapitel)
- 8. Aktuell halten des Debian-Systems: `uptodate.sgml` (gesamtes Kapitel)
- 9. Debian und der Kernel: `kernel.sgml` (gesamtes Kapitel)
- 10. Anpassen Ihrer Installation von Debian GNU/Linux: `customizing.sgml` (Teil des Kapitels)

Diese Abschnitte der "Debian-FAQ" wurden in diesem Dokument nach großer Umorganisation eingebunden, um Änderungen im Debian-System zu folgen. Beide Dokumente werden nun gleichzeitig aktualisiert.

Die ursprüngliche "Debian-FAQ" wurde erstellt und betreut von J. H. M. Dassen (Ray) und Chuck Stickelman. Autoren der umgeschriebenen "Debian-FAQ" sind Susan G. Kleinmann und Sven Rudolph. Nach diesen wurde die "Debian-FAQ" von Santiago Vila betreut. Der aktuelle Betreuer ist Josip Rodin.

Teile der Informationen für die "Debian-FAQ" kommen von:

- Der Debian-1.1-Ausgabe-Ankündigung von Bruce Perens (`http://www.perens.com/`).
- Der Linux-FAQ von Ian Jackson (`http://www.chiark.greenend.org.uk/~ijackson/`).
- Dem Debian-Mailinglist-Archiv (`http://lists.debian.org/`).
- Dem dpkg-Programmierer-Handbuch und dem Debian-Policy-Handbuch (vergleichen Sie 'Referenzen' auf Seite 235)
- vielen Entwicklern, Freiwilligen und Beta-Testern sowie
- dem unzureichenden Gedächtnis der Autoren. :-)

Einige Teile des "Tutorial"-Abschnitts stammen aus

- "Debian-Tutorial" von Havoc Pennington, Oliver Elphick, Ole Tetlie, James Treacy, Craig Sawyer und Ivan E. Moore II. (Dieses Dokument ging aus "Linux User's Guide" von

Larry Greenfield hervor.)

- ”Debian GNU/Linux: Guide to Installation and Usage“ von John Goerzen und Ossama Othman.

Die Autoren möchten all jenen danken, die halfen dieses Dokument zu ermöglichen.

A.2 Haftung

Da ich kein Experte bin, gebe ich nicht vor, ein volles Wissen über Debian oder Linux im Allgemeinen zu besitzen. Die Sicherheitsüberlegungen die ich aufführe sind eventuell nur für private Anwendungen anwendbar.

Dieses Dokument ersetzt nicht die ultimativen Anleitungen.

Jegliche Haftung wird abgelehnt. Alle Handelsmarken sind das Eigentum der entsprechenden Markeninhaber.

A.3 Rückmeldungen

Kommentare und Ergänzungen zu diesem Dokument sind immer willkommen. Senden Sie bitte eine E-Mail an das Debian-BTS-System (`http://bugs.debian.org/`) unter dem `debian-reference` Paket oder unter dem entsprechend übersetzten Paket. Das Verwenden von `reportbug` vereinfacht das Erstellen eines vollständigen Fehlerberichts. Sie können auch in Englisch eine E-Mail an Osamu Aoki (`http://people.debian.org/~osamu/`) nach `<osamu\#at\#debian.org>` oder zu jedem Übersetzer in dessen Sprache senden.

Obwohl ich in den USA lebte, bin ich kein einheimischer englischer Nutzer. Jegliche grammatikalische Korrekturen sind willkommen.

Die beste Rückmeldung ist ein Patch gegen die SGML-Version, aber ein Patch gegen die Textversion ist auch willkommen. Vergleichen Sie ‘Das offizielle Dokument’ auf Seite 1 für die offizielle Dokumentseite.

Die ursprünglichen SGML-Dateien, die zur Erzeugung dieses Dokuments verwendet wurden, sind auch im CVS erhältlich unter: `:pserver:anonymous@cvs.sf.net/cvsroot/qref` oder `http://qref.sourceforge.net/Debian/qref.tar.gz`.

A.4 Dokumentformat

Dieses Dokument wurde mit der DebianDoc-SGML-DTD geschrieben (umgeschrieben von LinuxDoc-SGML). Das DebianDoc-SGML-System ermöglicht es, Dateien in verschiedenen Formaten aus einer Quelle zu erzeugen, z.B. kann dieses Dokument in HTML, einfachem Text, TeX-DVI, PostScript, PDF und GNU-info betrachtet werden.

Konvertierungshilfsmittel für DebianDoc-SGML sind im Debian-Paket `debiandoc-sgml` erhältlich.

A.5 Das Debian-Labyrinth

Das Linux-System ist eine sehr mächtige Computerplattform für einen vernetzten Computer. Dennoch ist das Lernen wie man all diese Eigenschaften nutzt nicht einfach. Das Einrichten eines Druckers ist ein gutes Beispiel.

Es gibt ein komplettes, detailliertes Verzeichnis namens "QUELLCODE". Dies ist sehr akkurat aber sehr schlecht zu verstehen. Es gibt auch Referenzen namens HOWTO und mini-HOWTO. Diese sind einfacher zu verstehen, tendieren aber dazu, zu viele Details anzugeben, so dass man das große Bild aus den Augen verliert. Ich habe manchmal das Problem, einen bestimmten Abschnitt in einem langen HOWTO zu finden, wenn ich nach einigen Kommandos suche.

Um durch dieses Labyrinth der Linux-Systemkonfiguration zu navigieren, begann ich einige Notizen zur Erinnerung im Textformat zu meiner schnellen Referenz zu schreiben. Diese Liste von Notizen wuchs schnell und ich lernte debiandoc in der Zwischenzeit. Das Ergebnis ist diese *Debian-Referenz*.

A.6 Debian-Zitate

Es folgen einige interessante Zitate von der Debian-Mailingliste.

- "Dies ist Unix. Es gibt dir genug Seil, um dich selbst aufzuhängen." – Miquel van Smoorenburg `<miquels@cistron.nl>`
- "Unix IST benutzerfreundlich… Es ist einfach nur wählerisch, wer seine Freunde sind." – Tollef Fog Heen `<tollef@add.no>`