

open source library



Spende an Debian GNU/Linux
inkl.
1,00 €

Frank Ronneburg

Debian GNU/Linux 4

Anwenderhandbuch

für Einsteiger, Umsteiger und Fortgeschrittene

 ADDISON-WESLEY

Mit Debian GNU/Linux 4 »Etch«





6 Systemadministration

6.1 Bootloader

Vor dem Laden des Linux-Kernels wird mit dem so genannten Bootloader ein kleines Programm geladen, das Ihnen eine Auswahl zwischen verschiedenen installierten Betriebssystemen erlaubt. Die unter GNU/Linux gebräuchlichsten Bootloader sind LILO und GRUB auf der i386-Hardware-Architektur. Andere Architekturen verwenden Bootloader wie SILO, MILO oder auch yaboot. Eine kleine Übersicht finden Sie unter <http://www.metroworks.com/MW/Develop/Embedded/Linux/bootloaders.htm>.

Mit Debian Sarge (3.1) wurde vom bisher verwendeten Bootloader `lilo` zu `grub` gewechselt.

6.1.1 LILO

Bei der Installation von Debian GNU/Linux auf Ihrem System wurde bis zur Version 3.1 von Debian der Bootloader LILO auf Ihrer Festplatte installiert und so eingerichtet, dass das neue Debian GNU/Linux-System gestartet wird.

LILO einrichten

LILO (<http://lilo.go.dyndns.org/>) ist ein vollständiger Boot-Manager, mit dem nicht nur Linux gebootet werden kann, sondern auch jedes andere System, das sich an die im PC vorherrschenden Konventionen hält. Konfiguriert wird dieser Boot-Manager über die Datei `/etc/lilo.conf`.

Wann immer Sie diese Datei ändern, müssen Sie das Programm `lilo` auf der Kommandozeile als Superuser aufrufen, um die Änderungen tatsächlich zu übernehmen. Es reicht nicht, nur die Konfigurationsdatei zu verändern!

Wichtig in diesem Zusammenhang sind die Zeilen, die mit `image` und `other` anfangen, sowie die jeweils nachfolgende Zeile. Diese Schlüsselwörter dürfen mehrfach verwendet werden. Jede so beginnende Zeile bezeichnet ein System, das von LILO gebootet werden kann. Eine solche Partition kann einen Kernel, eine Root-Partition oder ein

anderes, Nicht-Linux-System (`other`) beinhalten. Die Reihenfolge dieser Systeme ist entscheidend, denn das erste wird automatisch gebootet, wenn die Wartezeit (Konfigurationsoption `delay` in der LILO-Konfiguration) abgelaufen ist und LILO nicht durch Drücken der `↑`-Taste angehalten wurde. Nach einer Erstinstallation existiert lediglich ein einziger Eintrag, der das aktuelle Debian GNU/Linux-System bootet. Hat der Debian Installer weitere Betriebssysteme auf anderen Partitionen gefunden, so werden diese in die Bootloader-Konfiguration aufgenommen.

Um ein zweites Linux-System zu booten (z.B. unter Verwendung eines anderen Kernels), müssen Sie die Datei `/etc/lilo.conf` um folgende Zeilen ergänzen:

```
image=/boot/vmlinuz.neu
label=neu
append='mcd=0x320,11'
read-only
```

Lediglich die ersten beiden Zeilen sind erforderlich. Um die Bedeutung der nachfolgenden Zeilen zu erfahren, lesen Sie bitte die Dokumentation zu LILO.

LILO und fremde Betriebssysteme

Um ein anderes System als Linux zu booten, verwenden Sie das Schlüsselwort `other` wie folgt:

```
other=/dev/hda1
label=win
```

Rufen Sie danach `lilo` als Superuser auf, um LILO neu zu installieren. Bei einem Neustart des Systems können Sie nun am Bootprompt durch Eingeben von `win` das Betriebssystem auf der Partition `/dev/hda1` starten.

6.1.2 GRUB

GRUB (GRand Unified Bootloader, bedeutet in deutscher Sprache »Larve«) <http://www.gnu.org/software/grub/grub.html> ist ein Bootloader, dessen Aufgabe es ist, nach dem Einschalten des Rechners die weitere Kontrolle an das Betriebssystem (also zunächst den Kernel) zu übergeben. Das Betriebssystem initialisiert dann über die geeigneten Treiber alle weitere Hardware im System. GRUB ist in der Lage, viele verschiedene Betriebssysteme zu laden. Wird ein Betriebssystem nicht direkt von GRUB unterstützt, so kann dieses dennoch von GRUB über einen so genannten »Chain-Bootloader« geladen werden. Für die wichtigsten Betriebssysteme sind diese bereits im GRUB-Paket enthalten.

Debian verwendet ab der Version 3.1 *Sarge* GRUB als Standard-Bootloader.

Einer der größten Vorteile von GRUB ist die Möglichkeit des direkten Zugriffs auf das Dateisystem. Das bedeutet, dass GRUB nicht wissen muss, wo genau der zu ladende Kernel auf der Festplatte liegt. Sie können also jederzeit einen neuen Kernel erzeugen und müssen hinterher nicht noch einmal GRUB im MBR der Festplatte installieren. Es ist ebenfalls nicht notwendig, alle Kernel-Versionen, die sich auf der Festplatte befinden,

den, in die Konfigurationsdatei einzutragen. Sie können mit GRUB jederzeit auf jeden Kernel auf Ihrer Festplatte zugreifen, wenn auch mit ein wenig Tipparbeit. Unterstützte Dateisysteme sind dabei Fat32, Fat16, ext2 und ReiserFS (unter Linux), UFS (BSD) und Minix.

GRUB kann durch die Unterstützung von BIOS-Erweiterungen Partitionen oberhalb der 8 GByte booten, und es ist möglich, Betriebssysteme von der zweiten Festplatte zu starten. Auch kann ein Betriebssystem über das Netzwerk geladen werden.

Installation

Die Installation von GRUB unter Debian ist wie üblich mit einem einfachen `apt-get install grub` so gut wie abgeschlossen. Sie sollten danach zunächst das Paket `lilo`, falls installiert, entfernen; dieses wird nicht mehr benötigt.

Um die notwendigen Daten (die Stage-1- und -2-Bootloader) in den Masterboot-Record der Festplatte zu schreiben, benutzen Sie das Programm `grub-install`. Wenn Sie `grub-install` zunächst ohne Optionen aufrufen, bekommen Sie eine kurze Übersicht der möglichen Optionen angezeigt:

```
debian:~# grub-install
install_device not specified.
Usage: grub-install [OPTION] install_device
Install GRUB on your drive.
-h, --help                print this message and exit
-v, --version             print the version information and exit
--root-directory=DIR     install GRUB images under the directory DIR
                        instead of the root directory.
--grub-shell=FILE       use FILE as the grub shell.
--force-lba              force GRUB to use LBA mode even for a buggy BIOS.
--recheck                probe a device map even if it already exists.
```

INSTALL_DEVICE can be a GRUB device name or a system device filename.

Reports bugs to <bug-grub@gnu.org>.

In den meisten Fällen ist es ausreichend, lediglich das gewünschte Device anzugeben, auf dem die Daten installiert werden sollen. Dies wird bei einer IDE-Festplatte meist `/dev/hda` sein.

```
surimi:/home/fr# grub-install /dev/hda
Installation finished. No error reported.
This is the contents of the device map /boot/grub/device.map.
Check if this is correct or not. If any of the lines is incorrect,
fix it and re-run the script `grub-install'.
(fd0)    /dev/fd0
(hd0)    /dev/hda
(hd1)    /dev/hdb
```

GRUB ist nun funktionsfähig installiert. Sie sollten aber noch ein Menü einrichten, um nicht bei jedem Systemstart die Parameter für das Root-Device und den Kernel von Hand eingeben zu müssen. Kopieren Sie hierzu am besten das mitgelieferte Beispiel an die entsprechende Stelle.

Konfiguration

Die Konfiguration von GRUB kann herkömmlich, mittels eines Editors, erfolgen oder aber ganz elegant mit `update-grub`.

`update-grub` ist Bestandteil des GRUB Debian-Pakets. Existiert noch keine Konfigurationsdatei, so wird diese automatisch erstellt. Als Basis dienen dazu die Kernel- und RAM-Disk-Images im Verzeichnis `/boot/`. Zu jedem vorhandenen Kernel wird jeweils ein Eintrag für den normalen Start sowie für den Start im Single-User-Modus erstellt.

```
wasabi:~# update-grub
Searching for GRUB installation directory ...
found: /boot/grub .
Testing for an existing GRUB menu.list file...
Could not find /boot/grub/menu.lst file. Would you like one
generated for you? (y/N) y
Updating /boot/grub/menu.lst ... done
```

Please note that configuration parameters for GRUB are stored in `/boot/grub/menu.lst`. You must edit this file in order to set the options which GRUB passes to the kernel, as well as the drive which GRUB looks in to for the kernel.

Everything on the line after "kopt=" is passed to the kernel as parameters, and "groot=" must be set to the partition(in GRUB terms, such as "(hd0,0)") which GRUB will load the kernel from.

After you have edited `/boot/grub/menu.lst`, please re-run 'update-grub'.

Natürlich sind Veränderungen an der automatisch erstellten Konfiguration möglich. In der Konfigurationsdatei gibt es einen Abschnitt, der mit

```
### BEGIN AUTOMAGIC KERNELS LIST
```

beginnt. Diese Einträge können verändert werden; es dürfen aber auf keinen Fall die Kommentarzeichen (`#`) am Anfang jeder Zeile verändert werden! Ein Beispiel: Aus

```
# kopt=root=/dev/hda1 ro
```

wird:

```
# kopt=root=/dev/hda1 ro apm=on ide2=0x180,0x386
```

Die hinzugefügten Parameter werden beim Starten des Kernels übergeben. Nach der Änderung wird nochmals `update-grub` aufgerufen, und die Änderungen werden in die eigentliche Konfiguration am Ende der Datei übernommen.

Das Kommando `grub-reboot` erlaubt es, einen bestimmten Eintrag aus der GRUB-Konfiguration auszuwählen und einmalig zu starten. Dies entspricht der von LILO bekannten Option `-R`.

Wird dieses Programm auf einem entfernten Rechner ausgeführt, so ist zu beachten, dass Einträge, die in der GRUB-Konfiguration mit einem Passwort versehen sind, einen Administrator vor Ort verlangen, der dieses Passwort eingibt. Das Passwort kann nicht als Option auf der Kommandozeile von `grub-reboot` angegeben werden.

Manuelle Konfiguration

Wenn Sie im Verzeichnis `/boot/grub/` eine Datei `menu.lst` erzeugen, so wird GRUB beim nächsten Start aus dieser Datei ein Menü erstellen, und Sie können dann aus den verschiedenen Einträgen dieses Menüs den gewünschten auswählen. Ein Beispiel für ein solches Menü finden Sie unter `/usr/share/doc/grub/examples/menu.lst`.

```
debian:~# cp /usr/share/doc/grub/examples/menu.lst /boot/grub/
```

In dem Beispiel finden Sie einige Einträge für die verschiedensten Betriebssysteme sowie einige andere Goodies, die Sie mit GRUB anstellen können. Eine funktionsfähige Minimalkonfiguration könnte wie folgt aussehen:

```
timeout 10
default 0

# For booting Linux
title Debian GNU/Linux
root (hd0,0)
kernel /vmlinuz root=/dev/hda1
```

Wenn ein Kernel-Paket verwendet wird, so ist darauf zu achten, dass diese normalerweise eine Init-Ramdisk (`initrd`) verwenden. Ein entsprechender Eintrag, um eben diese Ramdisk ergänzt, sieht wie folgt aus:

```
title GNU/Linux 2.6 SMP
root (hd0,6)
kernel /vmlinuz-2.6-686-smp root=/dev/hda1
read-only
initrd /initrd-2.6-686-smp
```

Dieses Beispiel verwendet einen SMP-Kernel und eine etwas andere Festplattenaufteilung, aber das Prinzip sollte erkennbar sein.

Splashscreen

Die Benutzung von Grafiken innerhalb des Bootloaders GRUB ist eine nicht-offizielle Erweiterung, die noch nicht in den Quellcode von GRUB eingeflossen ist. Trotzdem ist ein optisch ansprechender Hintergrund eine sehr attraktive Funktion, die bereits in viele Linux-Distributionen eingeflossen ist. Zuerst tauchte diese Erweiterung im Jahr 2001 auf einer GRUB-Mailingliste auf. Ab der Version 1.0 von GRUB tauchte sie auch im offiziellen Quellcode von GRUB auf.

Unabhängig davon ist im Debian-Paket von GRUB dieser Patch bereits integriert. Es sind aber einige Änderungen an der Konfigurationsdatei notwendig.

Ältere Versionen des Debian-Pakets von GRUB waren noch nicht mit diesem Patch ausgestattet, so dass bei einem Upgrade von GRUB auch unbedingt das Kommando `grub-install` nochmals auszuführen ist.

Weiterhin benötigt man eine Grafikdatei im komprimierten XPM-Format mit maximal 14 Farben und einer Auflösung von 640x480 Punkten. Diese kann mit folgenden Kommandozeilen aus einer JPEG-Datei erzeugt werden:

```
$ convert -colors 14 -geometry 640x480 grubsplash.jpg grubsplash.xpm
$ gzip grubsplash.xpm
```

Abschließend ist noch eine Anpassung an der Konfigurationsdatei von GRUB vorzunehmen. Um die Grafikdatei zu laden, muss in der GRUB-Konfigurationsdatei (`/boot/grub/menu.lst`) angegeben werden, wo diese Datei zu finden ist. Da GRUB das Dateisystem direkt lesen kann, muss diese Datei nicht zwingend auf der Partition liegen, auf der auch die GRUB-Konfigurationsdatei zu finden ist. Ein entsprechender Eintrag könnte wie folgt aussehen:

```
splashimage=(hd0,5)/boot/grub/grubsplash.xpm.gz
```

Nach einem Neustart müssen eventuell noch die Farben angepasst werden, so dass der Text lesbar ist.

Hardware-Bezeichnungen

Vielleicht haben Sie sich schon über die etwas ungewohnten Bezeichnungen der Laufwerke in der Konfigurationsdatei gewundert. GRUB benutzt die vom HURD-Kernel verwendeten Bezeichnungen für Gerätedateien; die Umstellung ist aber für den geübten Linux-Admin nicht sehr groß.

Zunächst ist zu bemerken, dass alle Bezeichnungen für Geräte in Klammern () geschrieben werden. Statt der unter Linux üblichen Buchstaben (z.B. `hda`) werden die einzelnen Festplatten mit fortlaufenden Zahlen ab 0 bezeichnet. `hda` entspricht also `hd0`. Die einzelnen Partitionen einer Festplatte werden wie unter Linux mit Zahlen bezeichnet, allerdings beginnend mit 0 und getrennt mit einem Komma. Somit entspricht `/dev/hda1` also `(hd0,0)`.

Weitere Geräte werden als `(fd0)` (Diskette) und `(sd0)` (SCSI-Festplatte) bezeichnet. Sie können GRUB auch zum Booten über das Netz, über PXE, einsetzen oder das Root-Filesystem per NFS mounten. Die Bezeichnung des Netzwerk-Device lautet `(nd)`.

Kommandozeile

GRUB verfügt auch über eine sehr leistungsfähige Kommandozeile. Bedenken Sie dabei, dass zu dieser Zeit noch kein kompletter Kernel oder gar ein komplettes Betriebssystem geladen ist! Sollte es Ihnen nicht gelungen sein, ein funktionsfähiges Menü zu erstellen, so landen Sie automatisch auf der GRUB-Kommandozeile. Sie können hier alle Kommandos, Laufwerksbezeichnungen und Dateinamen mit der -Taste automatisch vervollständigen lassen; dieses Feature kennen Sie sicher schon aus der Shell.

Wenn jedoch ein Menü angezeigt wird, können Sie dieses mit der Taste  verlassen und auf der Kommandozeile zum Beispiel von Hand einen anderen Kernel starten. Fehlerhafte Einträge im Menü lassen sich aus dem Menü mit der Taste  temporär verändern. Vergessen Sie nicht, diese Änderungen später in die Datei `/boot/grub/menu.lst` einzutragen.

Da zu diesem Zeitpunkt (GRUB wurde von der Festplatte gestartet, es ist kein Betriebssystem geladen) noch keine Tastaturbelegung geladen wurde, ist zu beachten, dass eine US-Tastaturbelegung gilt. Dies kann durch Anpassungen in der GRUB-Konfigurationsdatei (`/boot/grub/menu.lst`) verändert werden. Das GRUB-Kommando `setkey alt neu` erlaubt es, Tasten umzubelegen. Um die wichtigsten Tasten einer in Deutschland gebräuchlichen Tastatur richtig zu belegen, sind die folgenden Einträge in der GRUB-Konfigurationsdatei notwendig:

```
setkey y z
setkey z y
setkey Y Z
setkey Z Y
setkey underscore question
setkey question underscore
setkey ampersand percent
setkey percent caret
setkey equal parenright
setkey semicolon less
setkey parenright parenleft
setkey less numbersign
setkey parenleft asterisk
setkey numbersign backslash
setkey doublequote at
setkey colon greater
setkey plus bracketright
setkey greater bar
setkey minus slash
setkey asterisk braceright
setkey slash ampersand
```

Wird GRUB von einem laufenden Betriebssystem aus aufgerufen, so gilt die aktuell geladene Tastaturbelegung.

Tastaturbelegung eines neuen Kernels mit GRUB

GRUB unterstützt eine Option, die es erlaubt, einen Kernel nur ein einziges Mal zu starten, um dann beim folgenden Start wieder einen Default-Kernel zu starten. Dies ist insbesondere nützlich, wenn das System in einem Rechenzentrum untergebracht ist oder am Standort des Systems keine Administratoren Zugriff auf das System haben, die in der Lage sind, einen Fehler beim Starten des Kernels zu beheben. Auch der Bootloader LILO kennt hierzu die Option `-R`.

Wird zur Konfiguration von GRUB das Skript `update-grub` eingesetzt, so sind keine Anpassungen an der Konfigurationsdatei notwendig. Sie müssen lediglich auf der Kommandozeile die GRUB-Shell starten (durch den Aufruf von `grub`).

Dort ist die Kommandozeile

```
savedefault --default=N --once
```

einzugeben. *N* steht dabei für die Nummer des Eintrags in der Konfigurationsdatei; die Zählung beginnt bei 0. Beim nächsten Neustart wird nun der gewünschte (neue) Kernel gestartet, beim übernächsten Start der in der GRUB-Konfiguration eingestellte Kernel. Das folgende Skript erspart Ihnen das Eintippen des Kommandos in der GRUB-Shell, Sie müssen lediglich den gewünschten Eintrag angeben.

```
#!/bin/sh

if [ `id -u` != 0 ] ; then
    echo "you must be root!"
else
    if [ ! $1 ] ; then
        echo "Usage: "$0" kernelnumber"
    else
        echo "setting kernel "$1" once"
        grub --batch <<EOT
        savedefault --default=$1 --once
        quit
    EOT

    fi
fi
```

Weitere Informationen

Weitere Informationen zu GRUB finden Sie auf Ihrem System unter `usr/share/doc/grub-doc/` (hierzu muss auch das Paket `grub-doc` installiert sein) und auf den Projekt-Webseiten zu GRUB.

6.2 Init-Skripte

Nach dem Laden des Linux-Kernels wird als erstes Programm überhaupt das Programm `init` gestartet. Dieses Programm startet alle weiteren Programme; welche das sind, wird über die Skripte im Verzeichnis `/etc/init.d/` aus dem Paket `sysv-rc` bestimmt. Je nach gewünschtem Runlevel des Systems zeigen Links aus den Verzeichnissen `/etc/rcN.d/` (*N* ist die Nummer des aktuellen Runlevels, normalerweise 2) auf die Skripte im Verzeichnis `/etc/init.d/`, so ist beispielsweise `/etc/rc2.d/S20exim4` ein Link auf das Startskript für Exim (`/etc/init.d/exim4`). Den aktuellen Runlevel des Systems zeigt der Befehl `runlevel` an. Das Anlegen und Löschen dieser Links wird bei der Installation der jeweiligen Dienste automatisch vorgenommen. Ein installierter Dienst wird auf einem Debian-System immer auch gestartet. Natürlich können Links von Hand hinzugefügt oder entfernt werden, aber auch für diese Arbeiten am System stellt Debian einige Werkzeuge zur Verfügung.

6.2.1 rcconf

rcconf ist ein einfaches Werkzeug, mit dem einzelne Dienste aus den Init-Skripten aktiviert oder deaktiviert werden können. Es ist mit rcconf nicht möglich, den Runlevel eines Dienstes zu bestimmen.



Abbildung 6.1: rcconf

Leider werden bei einer Aktualisierung eines Paketes alle Links auf die Skripte wieder hergestellt. Dies bedeutet auch, dass mittels rcconf deaktivierte Dienste nach einem Update der Software wieder aktiviert werden.

6.2.2 update-rc.d

Detailliertere Einstellmöglichkeiten bietet update-rc.d. Zunächst sollte update-rc.d in jedem Fall mit der Option `-n` aufgerufen werden; diese zeigt lediglich an, welche Aktionen durchgeführt werden sollen, lässt die entsprechenden Links aber unangetastet.

```
usage: update-rc.d [-n] [-f] <basename> remove
       update-rc.d [-n] <basename> defaults [NN | sNN kNN]
       update-rc.d [-n] <basename> start|stop NN runlvl [runlvl] [...] .
       -n: not really
       -f: force
```

Grundsätzlich ist natürlich immer der Name des Pakets anzugeben, dessen Init-Skripte angepasst werden sollen. Bei Verwendung der Option `remove` werden alle Links auf das entsprechende Skript entfernt. Hierbei prüft update-rc.d, ob das Skript bereits entfernt wurde. Ist dies nicht der Fall, so bricht update-rc.d ab. Mit der Option `-f` kann erzwungen werden, dass die Links auch bei Vorhandensein eines Skriptes gelöscht werden.

Mit den Optionen `start`, `stop` bzw. `defaults` werden die notwendigen Links angelegt, gelöscht bzw. wiederhergestellt. Werden zusätzlich einer oder mehrere Runlevel angegeben, so können die Links entsprechend den Wünschen des Systemadministrators angepasst werden.

Leider werden bei einer Aktualisierung eines Paketes alle Links auf die Skripte wieder hergestellt. Dies bedeutet auch, dass mittels `update-rc.d` deaktivierte Dienste nach einem Update der Software wieder aktiviert werden.

6.2.3 file-rc

Als Alternative zu den System-V-Init-Skripten steht das Paket `file-rc` zur Verfügung. Bei diesem Paket wird die Konfiguration des Systemstarts zentral in einer einzelnen Datei verwaltet. Natürlich kann nur ein Paket den Systemstart steuern, so dass bei der Installation dieser für das System sehr essenziellen Funktionen eine deutliche Warnung ausgegeben wird.

```
wasabi:~# apt-get install file-rc
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut... Fertig
Die folgenden Pakete werden ENTFERNT:
  sysv-rc
Die folgenden NEUEN Pakete werden installiert:
  file-rc
WARNUNG: Die folgenden essenziellen Pakete werden entfernt.
Die sollte NICHT geschehen, wenn Sie nicht genau wissen, was Sie tun!
  sysv-rc (wegen sysvinit)
0 aktualisiert, 1 neu installiert, 1 zu entfernen und 0 nicht aktualisiert.
Es müssen 36,2kB Archive geholt werden.
Nach dem Auspacken werden 4096B Plattenplatz freigegeben sein.
Sie sind im Begriff, etwas potenziell Schädliches zu tun.
Zum Fortfahren geben Sie bitte "Ja, tu was ich sage!" ein.
?] Ja, tu was ich sage!
Hole:1 http://ftp.de.debian.org sarge/main file-rc 0.8.5 [36,2kB]
Es wurden 36,2kB in 0s geholt (49,9kB/s)
dpkg: sysv-rc: Abhängigkeitsproblem, aber lösche es auf Anfrage dennoch:
  sysvinit hängt ab von sysv-rc (>= 2.85-2) | file-rc (>> 0.7.0); aber:
  Paket sysv-rc soll gelöscht werden.
  Paket file-rc ist nicht installiert.
(Lese Datenbank ... 43720 Dateien und Verzeichnisse sind derzeit installiert.)
Entferne sysv-rc ...
Wähle vormals abgewähltes Paket file-rc.
(Lese Datenbank ... 43703 Dateien und Verzeichnisse sind derzeit installiert.)
Entpacke file-rc (aus .../archives/file-rc_0.8.5_all.deb) ...
Richte file-rc ein (0.8.5) ...
wasabi:~#
```

`file-rc` vereinheitlicht den Zugriff auf die Init-Skripte der einzelnen Dienste, indem alle (bisher von `sysv-rc` verwendeten) symbolischen Links entfernt werden und die komplette Verwaltung über die Datei `/etc/runlevel.conf` erfolgt.

```
# This file was automatically generated by /usr/share/file-rc/rclink2file.sh.
# You can use your favourite editor or update-rc.d(8) to modify it.
# Read runlevel.conf(5) man page for more information about this file.
#
# Format:
# <sort> <off-> <on-levels>      <command>
```

```

01  0,1,6  -      /etc/init.d/gdm
02  -      S      /etc/init.d/mountvirtfs
05  -      1      /etc/init.d/single
05  -      S      /etc/init.d/bootlogd
05  -      S      /etc/init.d/initrd-tools.sh
05  -      S      /etc/init.d/keymap.sh
10  -      2,3,4,5  /etc/init.d/syslogd
10  -      S      /etc/init.d/checkroot.sh
11  0,1,6  -      /etc/init.d/cron
11  -      2,3,4,5  /etc/init.d/klogd
14  -      2,3,4,5  /etc/init.d/ppp
18  -      2,3,4,5  /etc/init.d/portmap
18  -      S      /etc/init.d/hwclockfirst.sh
19  0,6    -      /etc/init.d/setserial
20  0,1,6  2,3,4,5  /etc/init.d/exim4
20  0,1,6  2,3,4,5  /etc/init.d/fam
20  0,1,6  2,3,4,5  /etc/init.d/inetd
20  0,1,6  2,3,4,5  /etc/init.d/lpd
20  0,1,6  2,3,4,5  /etc/init.d/makedev
20  0,1,6  2,3,4,5  /etc/init.d/ssh
20  -      0,6    /etc/init.d/sendsigs
20  -      S      /etc/init.d/modutils
21  -      2,3,4,5  /etc/init.d/nfs-common
25  0,6    -      /etc/init.d/hwclock.sh
30  -      0,6    /etc/init.d/urandom
30  0,6    S      /etc/init.d/etc-setserial
30  -      S      /etc/init.d/checkfs.sh
30  -      S      /etc/init.d/procps.sh
31  -      0,6    /etc/init.d/umountnfs.sh
32  -      0,6    /etc/init.d/portmap
35  -      0,6    /etc/init.d/networking
35  -      S      /etc/init.d/mountall.sh
36  -      S      /etc/init.d/discover
36  -      S      /etc/init.d/mountvirtfs
38  -      S      /etc/init.d/pppd-dns
39  -      S      /etc/init.d/dns-clean
39  -      S      /etc/init.d/ifupeupdown
40  -      0,6    /etc/init.d/umountfs
40  -      S      /etc/init.d/hostname.sh
40  -      S      /etc/init.d/hotplug
40  -      S      /etc/init.d/networking
43  -      S      /etc/init.d/portmap
45  -      S      /etc/init.d/mountnfs.sh
46  -      S      /etc/init.d/setserial
48  -      S      /etc/init.d/console-screen.sh
50  -      S      /etc/init.d/hwclock.sh
55  -      S      /etc/init.d/bootmisc.sh
55  -      S      /etc/init.d/urandom
70  -      S      /etc/init.d/nviboot
70  -      S      /etc/init.d/screen-cleanup
70  -      S      /etc/init.d/xfree86-common
79  0,1,6  -      /etc/init.d/nfs-common
81  1      -      /etc/init.d/portmap
86  0,1,6  -      /etc/init.d/ppp

```

```

89      0,1,6  2,3,4,5    /etc/init.d/atd
89      0,1,6  -          /etc/init.d/klogd
89      0,6    -          /etc/init.d/hotplug
89      -     2,3,4,5    /etc/init.d/cron
90      0,1,6  -          /etc/init.d/syslogd
90      -     0          /etc/init.d/halt
90      -     6          /etc/init.d/reboot
99      -     2,3,4,5    /etc/init.d/gdm
99      -     2,3,4,5    /etc/init.d/rmnologin
99      -     2,3,4,5    /etc/init.d/stop-bootlogd
# THE LAST LINE IS NEVER READ!

```

6.3 Alternativen (update-alternatives)

»Alternativen« sind eine weitere dieser vielen netten Kleinigkeiten, die das Leben in einem Debian-System leichter und schöner machen. Beliebte Programme (beispielsweise vi) können durch »Alternativen« (beispielsweise vim) ersetzt werden. Hierbei sind keine Einträge in irgendwelchen Konfigurationsdateien notwendig.

Die Debian-Distribution liefert einige tausend Programme mit, für jeden Einsatzzweck kann der Anwender meist zwischen verschiedenen Programmen wählen. Dies geschieht im Normalfall auf der Kommandozeile durch Eintippen des entsprechenden Kommandos oder via Maus über ein Menü. Mit den »Alternativen« ist der Administrator in der Lage, eine sinnvolle Voreinstellung für das Gesamtsystem zu geben.

Sehen wir uns das einmal am Beispiel vi an. Zunächst versuchen wir zu ermitteln, wo das Programm vi eigentlich zu finden ist:

```

wasabi:/home/fr# ls -l `which vi`
lrwxrwxrwx 1 root root 20 24. Feb 13:47 /usr/bin/vi -> /etc/alternatives/vi

```

Und der Link

```

wasabi:/home/fr# ls -l /etc/alternatives/vi
lrwxrwxrwx 1 root root 12 20. Mär 08:15 /etc/alternatives/vi -> /usr/bin/vim

```

zeigt wiederum auf das Programm vim. Nun lassen sich auch diese Links natürlich mit Unix-Bordmitteln verändern, doch halt! Auch hier gibt es wieder ein Werkzeug aus der Debian Werkzeugkiste.

update-alternatives ist das Werkzeug des Debian-Systemadministrators, um die Konfiguration dieser Links vorzunehmen. Hierbei ist es möglich, aus sinnvollen Alternativen auszuwählen; es werden sogar Einschätzungen zum am besten geeigneten Programm für den jeweiligen Zweck abgegeben.

```

update-alternatives: need --display, --config, --install,
--remove or --auto

```

```

Debian update-alternatives 1.9.20.
Copyright (C) 1995 Ian Jackson.
Copyright (C) 2000,2001,2002 Wichert Akkerman
This is free software; see the GNU General Public Licence

```

version 2 or later for copying conditions. There is NO warranty.

```
Usage: update-alternatives --install <link> <name>
      <path> <priority>
      [--slave <link> <name> <path>] ...
update-alternatives --remove <name> <path>
update-alternatives --auto <name>
update-alternatives --display <name>
update-alternatives --config <name>
<name> is the name in /etc/alternatives.
<path> is the name referred to.
<link> is the link pointing to /etc/alternatives/<name>.
<priority> is an integer; options with higher numbers are chosen.
```

```
Options: --verbose|--quiet --test --help --version
         --altdir <directory> --admindir <directory>
```

Zunächst sollte man sich einen Überblick über den aktuellen Zustand des Programms, hier `vi`, verschaffen:

```
wasabi:/home/fr# update-alternatives --display vi
vi - status is manual.
  link currently points to /usr/bin/vim
/usr/bin/nvi - priority 30
  slave vi.1.gz: /usr/share/man/man1/nvi.1.gz
/usr/bin/vim - priority 120
  slave vi.1.gz: /usr/share/man/man1/vim.1.gz
Current `best' version is /usr/bin/vim.
```

Hier ist sehr schön zu sehen, dass zwei Programme zur Verfügung stehen, die dem System als Editor `vi` dienen können: `nvi` und `vim`. Beide Versionen sind mit unterschiedlich hohen Prioritäten versehen; die höhere Version wird bevorzugt und in der letzten Zeile als Empfehlung ausgegeben.

Mit der Option `--config` können Veränderungen an der aktuellen Einstellung vorgenommen werden. Die Auswahl des neuen Wertes erfolgt mit der jeweils vorangestellten Zahl.

6.4 Systemzeit

(Fast) jeder Rechner verfügt heute über eine eingebaute Uhr, die über eine eingebaute Batterie die Uhrzeit auch im ausgeschalteten Zustand weiterlaufen lässt, so dass immer eine korrekte Systemzeit eingestellt ist. Das Betriebssystem kümmert sich während der Laufzeit um die richtige Zeit und liest sie einmalig beim Systemstart aus der Hardware-Uhr aus. Die Zeit des Betriebssystems lässt sich mit dem Kommando `date` anzeigen.

```
bash-2.03$ date
Sat Jan 22 17:54:00 CET 2000
```

Die Uhrzeit, die vom Uhrenchip auf dem Motherboard geliefert wird, ist normalerweise ausreichend genau. Wenn diese Zeit nach dem Einschalten des Rechners um mehrere Monate oder Jahre abweicht, sollten Sie die Batterie auf dem Motherboard auswechseln (lassen).

Das Kommando `date`, so wie oben gezeigt, zeigt den Wochentag, den Monat, den Tag, die Uhrzeit, die Zeitzone sowie das Jahr an.

Ein weiteres Kommando, welches auf die Systemzeit zugreift, ist das Kommando `time`. Dieses zeigt allerdings nicht die Zeit an, wie man vielleicht vermuten könnte, sondern dient dazu, die Zeit festzulegen, die bestimmte Kommandos zur Ausführung benötigen.

```
time sleep 60

real    1m0.057s
user    0m0.000s
sys     0m0.000s
```

Wundern Sie sich nicht: Dieses Kommando benötigt eine Minute, bis es beendet wird. Das Kommando `sleep` »schläft« in diesem Beispiel 60 Sekunden, und `time` ermittelt diese Zeit ;-). Wenn Sie sich einen neuen Rechner zugelegt haben und Ihre Freunde wissen, dass Sie diesen unter GNU/Linux betreiben, werden sie vielleicht fragen, wie lange es dauert, einen neuen Kernel zu übersetzen. Sie können diese Zeit mit dem Kommando `time` ermitteln. Natürlich kommt es auch darauf an, welche Treiber Sie in den Kernel einbinden wollen und welche Kernel-Version Sie übersetzen. Doch das ist ein anderes Thema.

6.4.1 date

Um die Systemzeit auf Ihrem Rechner zu korrigieren, können Sie dieses Kommando als Superuser benutzen. `date` hat eine Vielzahl von Optionen, um die Zeit zu stellen, Sie können die gewünschte Zeit in allen nur denkbaren Formaten angeben. Es gibt folgende Optionen:

```
bash-2.03$ date --help
Usage: date [OPTION]... [+FORMAT]
or: date [OPTION] [MMDDhhmm[[CC]YY][.ss]]
Display the current time in the given FORMAT, or set the system date.
-d, --date=STRING      display time described by STRING, not 'now'
-f, --file=DATEFILE   like --date once for each line of DATEFILE
-I, --iso-8601[=TIMESPEC] output an ISO-8601 compliant date/time string.
TIMESPEC='date' (or missing) for date only, 'hours', 'minutes', or 'seconds'
                        for date and time to the indicated precision.
-r, --reference=FILE   display the last modification time of FILE
-R, --rfc-822          output RFC-822 compliant date string
-s, --set=STRING       set time described by STRING
-u, --utc, --universal print or set Coordinated Universal Time
--help                display this help and exit
--version              output version information and exit
FORMAT controls the output. The only valid option for the second form
```

specifies Coordinated Universal Time. Interpreted sequences are:

```

%% a literal %
%a locale's abbreviated weekday name (Sun..Sat)
%A locale's full weekday name, variable length (Sunday..Saturday)
%b locale's abbreviated month name (Jan..Dec)
%B locale's full month name, variable length (January..December)
%c locale's date and time (Sat Nov 04 12:02:33 EST 1989)
%d day of month (01..31)
%D date (mm/dd/yy)
%e day of month, blank padded ( 1..31)
%h same as %b
%H hour (00..23)
%I hour (01..12)
%j day of year (001..366)
%k hour ( 0..23)
%l hour ( 1..12)
%m month (01..12)
%M minute (00..59)
%n a newline
%p locale's AM or PM
%r time, 12-hour (hh:mm:ss [AP]M)
%s seconds since 00:00:00, Jan 1, 1970 (a GNU extension)
%S second (00..60)
%t a horizontal tab
%T time, 24-hour (hh:mm:ss)
%U week number of year with Sunday as first day of week (00..53)
%V week number of year with Monday as first day of week (01..52)
%w day of week (0..6); 0 represents Sunday
%W week number of year with Monday as first day of week (00..53)
%x locale's date representation (mm/dd/yy)
%X locale's time representation (%H:%M:%S)
%y last two digits of year (00..99)
%Y year (1970...)
%z RFC-822 style numeric timezone (-0500) (a nonstandard extension)
%Z time zone (e.g., EDT), or nothing if no time zone is determinable
By default, date pads numeric fields with zeroes. GNU date recognizes
the following modifiers between '%' and a numeric directive.
`-' (hyphen) do not pad the field
`_' (underscore) pad the field with spaces
Report bugs to <bug-sh-utils@gnu.org>.

```

In der Praxis werden Sie die Zeit aber nur selten korrigieren müssen, so dass an dieser Stelle ein Beispiel reichen sollte. Sehen Sie sich noch einmal die Hilfe zu `date` an. In der zweiten Zeile finden Sie Folgendes: `or: date [OPTION] [MMDDhhmm[[CC]YY][.ss]]`. Die gewünschte Zeit kann im Format MM (Month – Monat), DD (Day – Tag), hh (Hour – Stunde) und mm (Minute) angegeben werden. Die anderen Werte können Sie ignorieren, diese werden Sie nur selten gebrauchen. Um die Zeit drei Tage und eine halbe Stunde zurückzustellen, benutzen Sie folgendes Kommando:

```

linux:/home# date
Sun Jan 23 10:47:59 CET 2000
linux:/home# date 01201017
Thu Jan 20 10:17:00 CET 2000

```

6.4.2 rdate

Wenn Sie über einen Internetanschluss verfügen, können Sie das Setzen der korrekten Uhrzeit auch Programmen wie zum Beispiel `rdate` überlassen. `rdate` übernimmt die Zeit von einem Server aus dem Netz, der über eine Funkuhr (DCF77-Signal) verfügt.

Sie können dieses Kommando in die Skripte einbinden, die Sie zum Aufbau der Internetverbindung benutzen. Wenn Ihr Rechner ständig oder immer zu bestimmten Zeiten mit dem Internet verbunden ist, können Sie `rdate` auch über das Programm `cron` aufrufen, indem Sie es in die `crontab` des Superusers eintragen. Rufen Sie dazu das Kommando `crontab` mit der Option `-e` (edit - verändern) auf:

```
bash-2.03$ su
Password:
linux:/home# crontab -e
```

Ergänzen Sie nun die Einträge in der `Crontab` um folgende Zeile:

```
10 01 * * * /usr/sbin/rdate time.fu-berlin.de > /dev/null
```

Ein solcher Eintrag stellt jede Nacht um zehn Minuten nach eins die Systemzeit nach den Angaben des Servers `time.fu-berlin.de` neu. Benutzen Sie bitte einen Server in Ihrer Nähe. Viele Provider bieten einen solchen Service an - fragen Sie einfach nach.

Wenn Sie keinen geeigneten Server finden können, sollten Sie die Server der Physikalisch-Technischen Bundesanstalt (PTB) in Braunschweig benutzen. Wie Sie vielleicht wissen, wird von der PTB das amtliche Zeitsignal ausgesendet, das von DCF-77-Empfängern (Funkuhren) empfangen werden kann. Seit Anfang 2000 stellt die PTB diesen Dienst auch über das Internet zur Verfügung. Sie können die beiden öffentlichen Zeitserver der PTB als `ptbtime1.ptb.de` und `ptbtime2.ptb.de` erreichen.

Um die vorhin in die Vergangenheit beförderte Systemzeit zu korrigieren, können Sie `rdate` natürlich auch auf der Kommandozeile ausführen:

```
linux:/home# /usr/sbin/rdate time.fu-berlin.de
Sun Jan 23 10:49:22 2000
```

6.5 Verwalten von Konfigurationsdateien

Mit fast jedem auf einem Linux-System installierten Paket werden auch die notwendigen Konfigurationsdateien für die jeweilige Software-Komponente mitgeliefert. Das Debian-Team leistet an dieser Stelle sehr gute Arbeit und erstellt zu allen Paketen angepasste Konfigurationsdateien, die nicht nur auf das Debian-System angepasst sind, sondern auch in fast 100 Prozent aller Fälle sofort lauffähig sind.

Dies bedeutet für den Administrator eine große Arbeitserleichterung, da die gewünschte Software sofort ohne weitere Konfigurationsarbeiten lauffähig ist. Natürlich können diese allgemeinen Vorgaben nicht alle Bedürfnisse abdecken, und so sind viele Pakete bereits mit Konfigurationsprogrammen ausgestattet, die den Administrator durch die Konfiguration führen. Mit `debconf` wurde sogar eine zentrale Daten-

bank bereitgestellt, die auch im Netz via LDAP verfügbar gemacht werden kann. In ihr werden alle Einstellungen gespeichert.

In einem größeren Netzwerk kann es aber wünschenswert sein, ein einheitliches System für die Verwaltung von Konfigurationsdateien, vielleicht sogar mit einer Versionskontrolle, zu etablieren. Hier bietet sich prinzipiell das Erstellen von eigenen Debian-Paketen mit den gewünschten Konfigurationsdateien an. Das Debian-Paketmanagement verhindert jedoch das Überschreiben von Dateien, so dass das Erstellen von Paketen mit angepassten Konfigurationsdateien ein Problem darstellt. Abhilfe schafft an dieser Stelle `dpsyco`.

6.5.1 `dpsyco`

`dpsyco` (Debian Packages of System Configurations) kann Debian-»Konfigurationspakete« erzeugen und verwalten. Ein »Konfigurationspaket« kann »über« ein normales Debian-Paket installiert werden und so die in dem Debian-Paket enthaltenen Konfigurationsdateien ersetzen oder auch ergänzen.

`dpsyco`-Pakete können Konfigurationsdateien aus einem Debian-Paket überschreiben, patchen oder auch Benutzer und Gruppen hinzufügen. Es sind verschiedene Pakete verfügbar, die Sie mit `dpsyco` verwenden können:

```
sushi:/home/fr# apt-cache search dpsy
dpsyco - Debian packages of system configurations
dpsyco-base - Base package for the debian packages of system configurations
dpsyco-cfengine - Automate applying of cfengine configs
dpsyco-devel - Tools to create configuration packages
dpsyco-doc - Documentation for the debian packages of system configurations
dpsyco-lib - Libraries for the debian packages of system configurations
dpsyco-mysql - Automate administration of access to mysql
dpsyco-patch - Automatically patch the debian file-system
dpsyco-samba - Automate administration of access to samba
dpsyco-skel - Automatically install a add-on skeleton
dpsyco-ssh - Automate administration of access via ssh
dpsyco-sudo - Automate administration of sudo privileges
```

Zunächst müssen Sie das Paket `dpsyco` selbst installieren. Dies hat zur Folge, dass auch die Pakete `dpsyco-base` und `dpsyco-lib` installiert werden:

```
sushi:/home/fr# apt-get install dpsyco
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
dpsyco-base dpsyco-lib
Suggested packages:
dpsyco-doc dpsyco-mysql
Recommended packages:
dpsyco-skel dpsyco-sudo
The following NEW packages will be installed:
dpsyco dpsyco-base dpsyco-lib
0 upgraded, 3 newly installed, 0 to remove and 522 not upgraded.
```

```
Need to get 54,2kB of archives.
After unpacking 218kB of additional disk space will be used.
continue? [Y/n]
Get:1 ftp://ftp.de.debian.org sid/main dpsyco-lib 1.0.23 [19,5kB]
Get:2 ftp://ftp.de.debian.org sid/main dpsyco-base 1.0.23 [24,3kB]
Get:3 ftp://ftp.de.debian.org sid/main dpsyco 1.0.23 [10,4kB]
Fetched 54,2kB in 1s (53,8kB/s)
```

Für die ersten Schritte können die Kommandos `dpsyco-delhome` und `dpsyco-restorehome` dienen. Hiermit kann ein Heimat-Verzeichnis eines Benutzers gelöscht und wieder hergestellt werden. Wichtig sind dabei die Einstellungen in der Datei `/etc/dpsyco/adduser.conf`.

6.6 Installations- und Rettungsdiskette

Sie können die Installationsdiskette oder die CD - falls Sie von dieser das System bei der Installation gestartet haben - auch zur Reparatur Ihres Systems benutzen. Dies kann in seltenen Fällen erforderlich sein, z. B. falls Sie den Rechner nicht ordnungsgemäß heruntergefahren haben, was ebenso bei einem Stromausfall der Fall ist. Manchmal kommt es auch vor, dass Sie etwas an den Dateien in `/etc` geändert haben oder dass ein neu erstellter Kernel nicht startet. Aber auch, wenn Sie das Passwort für den Superuser vergessen haben, kann die Bootdiskette Sie vor der Neuinstallation bewahren.

In einem Notfall legen Sie die Rettungsdiskette ein und starten das System neu. Statt wie bei der Installation einfach die Eingabetaste zu drücken, können Sie nun am Boot-Prompt zusätzliche Parameter eingeben. Eine umfangreiche Übersicht der Befehle befindet sich auf der Diskette. Diese können Sie sich mit den Funktionstasten ansehen. Wir werden hier nur zwei gebräuchliche Fälle vorstellen.

Sollte zum Beispiel die LILO-Konfiguration fehlgeschlagen sein oder lässt sich der neu übersetzte Kernel nicht booten, so sind immer noch alle Daten auf der Root-Partition in einwandfreiem Zustand. Sie haben jetzt die Möglichkeit, den auf der Rettungsdiskette befindlichen Kernel zu booten und das Dateisystem auf der Root-Partition zu mounten. Geben Sie hierzu am Prompt Folgendes ein: `rescue root=/dev/hda1`

Hierbei ist zu beachten, dass Sie – je nachdem, auf welcher Partition sich Ihre Root-Partition befindet – das passende Device angeben. Nachdem der Kernel von der Diskette geladen worden ist, wird das System ganz normal gestartet, und Sie können sich danach am System anmelden und den Fehler beheben. Im Fall einer nicht funktionierenden Konfiguration von LILO loggen Sie sich als Superuser ein, beheben die Fehler in der Konfigurationsdatei und rufen anschließend das Kommando `lilo` auf.

Ein anderer Fall wäre ein Fehler im Dateisystem der Root-Partition, der ein normales Booten des Systems verhindert. In den meisten Fällen kommt man hierbei sogar ohne eine Rettungsdiskette aus. Meist hält das System beim Start an, und nach Eingabe des Root-Passworts können Sie das Dateisystem dann reparieren.

Dies lässt sich auch mit einer Rettungsdiskette ausführen. Booten Sie hierzu ganz normal mit der Diskette, als wollten Sie eine Installation durchführen. Wählen Sie noch

die gewünschte Tastatur aus und wechseln Sie dann mit der Tastenkombination `ALT+F3` auf die dritte Konsole. Nach Drücken der Eingabetaste steht dort eine Shell zur Verfügung. Sie können von dort aus das defekte Dateisystem manuell reparieren. Benutzen Sie dazu das Kommando: `e2fsck /dev/hda1`. Auch hier ist wieder das entsprechende Device anzugeben. Nach einem Neustart des Systems sollte alles wieder wie gewohnt funktionieren.

6.7 Technische Informationen zu den Bootdisketten

6.7.1 Quellcode

Das Paket `boot-floppies` enthält den gesamten Quellcode, um die Installationsdisketten herstellen zu können.

6.7.2 Die Notfalldiskette

Auf der Notfalldiskette ist ein MS-DOS-Dateisystem enthalten, auf das von jedem System zugegriffen werden kann, das DOS-Disketten mounten kann. Der Linux-Kernel liegt in der Datei `linux`. Die Datei `root.bin` ist ein mit dem Programm `gzip` komprimiertes Image eines 1.44 Mbyte großen Minix-Dateisystems. Dieses wird in die RAM-Disk im Hauptspeicher geladen und dort als Festplattenersatz für das Root-Dateisystem verwendet.

6.7.3 Kernel ersetzen

Wenn der Kernel auf der Notfalldiskette ersetzt werden soll, muss der neue Linux-Kernel mit den folgenden Eigenschaften (als fester Bestandteil, nicht als ladbare Module) konfiguriert werden:

- »Initial RAM disk«-Unterstützung
- MS-DOS-, Minix- und EXT2-Dateisystem
- Ausführbare Dateien im ELF-Format

Danach wird der neu erzeugte Kernel (`zImage` oder `bzImage`) mit dem Dateinamen `linux` auf die Notfalldiskette kopiert. Anschließend rufen Sie das Shell-Skript `rdev.sh` auf, das ebenfalls auf der Diskette zu finden ist. Dieses Programm nimmt noch ein paar kleine Einstellungen an dem neuen Linux-Kernel vor.

6.7.4 Die Basisdisketten

Die Disketten des Basis-Systems enthalten einen 512 Byte großen Kopf sowie jeweils einen Teil eines mit `gzip` komprimierten tar-Archivs. Werden jeweils die Köpfe entfernt und danach die Inhalte der Disketten zusammengefügt, so ergibt sich das vollständige komprimierte tar-Archiv.

Dieses Archiv enthält das Grundsystem, das während der Installation auf Ihr System kopiert wird. Es bietet die Basis-Funktionalität eines Debian GNU/Linux-Systems, dem jedoch noch zahlreiche Programme fehlen. Wenn das Archiv installiert ist, muss das System zunächst mit dem Menüpunkt CONFIGURE THE BASE SYSTEM konfiguriert werden. Mit Hilfe von weiteren Menüeinträgen im Installationssystem werden die Netzwerkanbindung, der Betriebssystemkern und seine Module installiert und konfiguriert. Erst danach kann das System eingesetzt werden.

6.8 Notebook-Spezialitäten

Notebooks zählen zwar nicht zu den Erweiterungen, die man sich »mal eben so« zulegt, sind aber doch recht verbreitet. Interessant sind diese Geräte in diesem Abschnitt auch aufgrund der benutzten Hardware-Komponenten.

Wenn Sie häufig mit Ihrem Notebook unterwegs sind, bietet Ihnen Debian GNU/Linux einige interessante Pakete:

An erster Stelle steht sicher das Paket `task-laptop`. Es fasst die wichtigsten Pakete für Besitzer tragbarer Computer zusammen. Es werden automatisch die Pakete `apmd` (kümmert sich um den Stromsparmodes Ihres Rechners), `pcmcia-cs` (die »allgemeinen« PCMCIA-Treiber) sowie `anacron` (eine Ergänzung zu `cron`, die auch die »Ruhezeiten« Ihres Rechners berücksichtigt und Jobs ggf. später startet) installiert.

Weitere sinnvolle Pakete (die auch von `task-laptop` vorgeschlagen werden) sind: `netenv` (hiermit können Sie beim Systemstart zwischen verschiedenen Netzwerkkonfigurationen wählen), `irda-tools` und `irda-common` (Infrarotschnittstellen-Unterstützung) sowie `toshutils` (einige Helferlein speziell für Toshiba-Notebooks).

Generelle Informationen zur Benutzung von GNU/Linux auf Notebooks finden Sie auf der Seite »UniX with Mobile Computers« von Werner Heuser unter der URL: <http://tuxmobil.org>. Konfigurationstipps, wie Sie die Laufzeit des Systems erhöhen können, finden Sie im *Battery Powered Linux Mini-HOWTO*.

Wenn Sie in den Paketen forschen, werden Sie noch weitere nützliche Programme finden. Aber einen Teil der Arbeit haben wir Ihnen schon abgenommen.

6.8.1 `divine`

`divine` wird aus den Startdateien des PCMCIA-Systems heraus aufgerufen und kann anhand der vorhandenen Konfigurationsdateien feststellen, welche Einstellungen für das Netzwerk vorzunehmen sind. Schließen Sie einfach Ihr Notebook ans Netz an, starten Sie den Rechner, und »schon sind Sie drin« ;-). Die Homepage von `divine` finden Sie unter <http://www.fefe.de/divine/>.

6.8.2 wmbattery

Um den Ladezustand des Akkus anzuzeigen, leistet `wmbattery` gute Dienste. Dieses Programm lässt sich im »Dock« des Windowmanagers WindowMaker ablegen, benötigt diesen aber nicht zwingend.



Abbildung 6.2: WMBattery

Sie können `wmbattery` auch mit jedem anderen Windowmanager benutzen. Um `wmbattery` betreiben zu können, muss das BIOS Ihres Notebooks über eine APM- (Advanced Power Management-)Unterstützung verfügen, und Sie müssen diese auch (unter Character Devices) im Kernel aktivieren. `wmbattery` zeigt Ihnen den prozentualen Ladezustand an. Zusätzlich wird der Zustand auch farblich dargestellt (grün, gelb, rot). Wenn das Notebook über das Netzteil betrieben wird und so der Akku geladen wird, wird auch dies angezeigt.

6.8.3 netenv

Mit dem Paket `netenv` können Sie beim Systemstart zwischen verschiedenen Netzwerkkonfigurationen wählen, je nachdem, wo Sie sich gerade befinden. Bitte beachten Sie, dass der Bootvorgang unterbrochen wird und Sie ein Profil auswählen müssen, wenn Sie `netenv` installiert haben. Sie können `netenv` auch dazu benutzen, einen anderen Windowmanager oder einen anderen Drucker zu verwenden.

6.8.4 cardinfo

`cardinfo` ist Bestandteil des Pakets `pcmcia-cs`. Sie können mit `cardinfo` alle Informationen über die erkannten PCMCIA-Karten sehen und auch Karten an- und abmelden oder zurücksetzen.



Abbildung 6.3: Cardinfo