Linux.Fibel.org
Eine Einführung in Linux
Version 0.5.0

Ole Vanhoefer
eMail: linux@vanhoefer.de

19. Januar 2003

Linux.Fibel.org

Eine Einführung in Linux

von Ole Vanhoefer

Gesetzt mit LaTeX unter Linux

Version 0.3	:	18. November 2001
Version 0.4	:	14. April 2002
Version 0.5.0	:	19. Januar 2003

Das Kleingedruckte

Inhaltsverzeichnis

Vorwort

In a world without walls and fences,
who needs windows and gates?

Anonymous

Dieses Skript ist als Begleitmaterial zu meinem Unterricht "Einführung in Linux" bei der Moebius GmbH in Kiel (`http://www.moebius.de`) entstanden. Schwerpunkt des Unterrichts ist eine Einführung in die Grundbedienung von Linux, wie sie dem Stoffplan der ersten Stufe der LPI-Prüfung entspricht. Leider kann dieses Skript erst einen Teil der beiden Prüfungen abdecken.

Das Skript ist einmal als Begleitmaterial zum Unterricht entstanden, da es noch kein passendes Buch zur LPI-Prüfung in Deutsch gibt, zum anderen als meine Vorbereitung für die Prüfung 101, die ich auch erfolgreich abgelegt habe.

Dieses Skript befindet sich in der Entwicklung und wird in Zukunft um einige Themen erweitert werden.

Neu hinzugekommen in der Version 0.5 ist das Thema *Shell-Scripting* sowie den grapischen Teil der SuSE-Distribution. Auch in der Kapitelaufteilung hat sich einiges geändert, sowie der Umfang mancher Kapitel.

Besonders möchte ich bei meinen bisherigen Teilnehmern danken, die mich auf Fehler und Unstimmigkeiten aufmerksam gemacht haben und sich als Beta-Tester für die Aufgaben geopfert haben.

Für Anregungen und Fehlerkorrekturen bin ich sehr dankbar. Die Aufgaben sollten zum größten Teil mit den Grundvoraussetzungen einer SuSE-Linux-Distribution ab Version 8.0 gelöst werden können.

Ich hoffe, Sie haben viel Spaß beim Lesen.

Ole Vanhoefer
Kiel, 19.01.2003

Teil I

Linux Administration

Kapitel 1

Einführung

1.1 Was ist eigentlich Linux?

Linux ist ein 32-Bit**Unix-ähnliches Betriebssystem**. Im Gegensatz zu den meisten anderen Unix-Systemen ist Linux inklusive der Programmquellcodes frei kopierbar. Inzwischen existiert eine Vielzahl von Anwendungen für Linux, so daß fast alle Aufgaben, die bisher auf Unix-, Windows- oder Apple-Systemen erledigt wurden, auch unter Linux zu meistern sind.

Ein **Betriebssystem** ist eine Sammlung von Programmen, mit denen die grundlegendsten Funktionen eines Rechners realisiert werden. Dies reicht von der Schnittstelle Mensch-Maschine über die Verwaltung der Daten bis zur Kontrolle und Steuerung der Systemressourcen. Ohne ein Betriebssystem können Sie mit Ihrem Rechner nicht arbeiten, da so wichtige Dinge wie das Starten von Programmen und die Verwaltung Ihrer Dateien davon abhängen.

Diese Funktionen stellen eigentlich alle Betriebssystem zur Verfügung. Somit werden Sie bei jeder Arbeit mit einem Betriebssystem konfrontiert, wie z. B. CP/M, DOS, Windows 9x, Windows NT, Windows 2000, Windows ME, Windows XP, OS/2, Mac-OS, Unix usw.

Linux basiert auf dem Betriebssystem UNIX, das 1969 von den Bell Laboratories entwickelt wurde. Der finnische Student Linus Benedict Thorvald entwickelte den ersten Linux Betriebssystemkern (*Kernel*) auf seinem 386er Rechner. Dabei nutzte er nicht den bestehenden Quellcode, sondern programmierte das Betriebssystem vollständig neu. Nach außen präsentiert sich Linux als UNIX-System, während es innen aus einem völlig eigenständigem Code besteht.

Thorvald stellte im September 1991 der Gemeinde der MINIX-Anhänger sein neues Betriebssystem in der Version 0.01 vor. Er entschloß sich dabei das Programm und den Quellcode frei weiterzugeben und anderen Programmierern die Arbeit an seinem System zu gestatten. Viele Programmierer begeisterten sich für Linux und schon im Januar 1992 wurde der erste stabile Kernel 0.12 ins Internet gestellt.

Eigentlich kann man nur den Kernel mit seinen Modulen und direkt dazugehörenden Daten als Linux bezeichnen. Daneben gibt es aber eine Vielzahl von Hilfsprogrammen und Applikationen, die zusammen mit dem Kernel in den sogenannten Distributionen vertrieben werden. Für diese Zusammenstellung hat sich der Name Linux nun eingebürgert.

1.1.1 Leistungsmerkmale des Kernels

Der aktuelle Kernel verfügt über folgenden Leistungsmerkmale:

- **POSIX-Konformität**: Der POSIX-Standard (*Portable Operating System Interface*) definiert eine Reihe von Bedingungen für portierbare Betriebssysteme.

- **Multitasking**: Mehrere Prozesse und damit Programme können (scheinbar) gleichzeitig ausgeführt werden.

- **Multiuser**: Mehrere Benutzer können gleichzeitig (z. B. über Telnet-Programme) auf dem Rechner arbeiten.
- **Paging**: Das Paging erlaubt die Auslagerung von Speicherinformationen auf die Festplatte, so daß der Eindruck eines wesentlich größeren Arbeitsspeichers entsteht.
- **Shared Libraries**: Die Shared Libraries sind Programm-Bibliotheken. Bei Bedarf werden sie in den Speicher geladen und mehrere Programme gleichzeitig können auf die enthaltenen Routinen zugreifen. Dies spart Speicherplatz.
- **Shared Memory**: Normalerweise werden beim Multitasking-Betrieb jedem Programm eigene exklusive Speicherbereiche zugewiesen. Die Kommunikation erfolgt über die IPC (*InterProcess Communication*). Um diesen umständlichen Weg zu umgehen, können gemeinsam genutzte Speicherbereiche definiert werden, in denen die Programme ihre Daten austauschen können.
- **Symetric Multi Processing (SMP)**: Beim Multitasking bekommen die Prozesse kleine Zeitscheiben zugewiesen, in denen sie ausgeführt werden. Da die Zeitscheiben schnell aufeinanderfolgen sieht es so aus, als ob die Programme gleichzeitig ausgeführt werden. Dies ist aber wirklich nur beim SMP der Fall. Linux ist in der Lage mehrere Prozesse auch auf mehreren Prozessoren gleichzeitig ausführen zu lassen.
- **Protected Mode**: Speicherschutzmechanismen des Prozessors verhindern, daß ein Prozeß auf den Speicherbereich eines anderen Prozesses zugreifen kann. Dadurch erscheint die von Windows noch bekannte, allseits beliebte allgemeine Schutzverletzung bei Linux nicht mehr.

1.2 Wie alles begann

Leider ist heute nicht mehr der Tag bekannt an dem Linus Benedict Torvalds[1] mit der Entwicklung des Linux Kernels begann. Ein Posting aus dem Usenet gibt aber einen Hinweis auf den möglichen Zeitpunkt, an dem er sich für das Thema zu interessieren begann.

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: Gcc-1.40 and a posix-question
Message-ID: <1991Jul3.100050.9886@klaava.Helsinki.FI>
Date: 3 Jul 91 10:00:50 GMT
Hello netlanders,
Due to a project I'm working on (in minix), I'm interested in the posix
standard definition. Could somebody please point me to a (preferably)
machine-readable format of the latest posix rules? Ftp-sites would be
nice.
```

Sicher ist, daß die erste Version des Linux Kernels am 25. August 1991 im Usenet angekündigt wurde. Schon kurze Zeit später fanden sich interessierte Programmierer, die an dem Projekt mitgearbeitet haben.

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.FI>
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki

Hello everybody out there using minix -
I'm doing a (free) operating system (just a hobby, won't be big and
professional like gnu) for 386(486) AT clones. This has been brewing
```

[1] Da ihn alle Welt eigentlich nur Linus nennt, wird hier auch so verfahren.

```
since april, and is starting to get ready. I'd like any feedback on
things people like/dislike in minix, as my OS resembles it somewhat
(same physical layout of the file-system (due to practical reasons)
among other things).
I've currently ported bash(1.08) and gcc(1.40), and things seem to work.
This implies that I'll get something practical within a few months, and
I'd like to know what features most people would want. Any suggestions
are welcome, but I won't promise I'll implement them :-)
Linus (torvalds@kruuna.helsinki.fi)
PS. Yes - it's free of any minix code, and it has a multi-threaded fs.
It is NOT protable (uses 386 task switching etc), and it probably never
will support anything other than AT-harddisks, as that's all I have :-(.
```

1991 Als erster genaue Termin steht der 3. Juli 1991 fest. Linux implementiert einige Gerätetreiber sowie den Festplattentreiber und einige User-Level Funktionen.

Linus veröffentlicht am 17. September 1991 die Version 0.01 des Kernels für einige Interessenten aus dem Usenet. Das Archiv mit der historischen Version ist auch heute noch verfügbar: http://www.kernel.org/pub/linux/kernel/Historic/linux-0.01.tar.gz.

Die erste “offizielle” Version des Linux Kernels (0.02) erscheint am 05. Oktober 1991. Mit dieser Version laufen bereits die bash, gcc, gnu-make, gnu-sed und compress.

Am 19. Dezember 1991 läuft die erste Version (0.11), die ohne die Hilfe eines anderen Betriebsystems lauffähig ist. Es gab keinen SCSI Support, so daß eine AT-Bus Festplatte Voraussetzung war. Es gab weder init noch login, nach den Systemstart landete man direkt in einer bash. Es gab Ansätze für die Implementierung von Virtual Memory, es waren aber mindestens 4 MB RAM notwendig um GNU Programme, insbesondere den `gcc` benutzen zu können. Ein einfacher Systemstart war aber auch schon mit 2 MB möglich.

Deshalb folgte für einige Personen eine Version mit virtueller Speicherverwaltung zu Weihnachten um den Kernel auch mit 2 MB RAM übersetzen lassen zu können.

1992 Bereits am 5. Januar kommt die erste Version (0.12) heraus, die mehr Funktionen hatte als unbedingt benötigt werden. Mit dieser Version wurde der Kernel unter die GPL gestellt. Die ältere Lizenz unter der der Kernel stand war in vielen Punkten deutlich strenger.

Linus verteilte diese per Anonymous-FTP im Internet, was zu einem sprunghaften Anstieg der Testerzahl führte. Da dieser Anstieg so groß wurde, daß die nötige Kommunikation nicht mehr per eMail zu bewältigen war, wurde im Usenet die Gruppe alt.os.linux geschaffen. Das Interesse an Linux wuchs stetig und wurde von Linus koordiniert.

Um die Entwicklung voranzutreiben erhöhte Linus die Entwicklungsnummer im März auf Version 0.95.

Die im April folgende Version 0.96 war die erste Version mit der es möglich war das X-Window System zu betreiben.

Man schätzt die Zahl der Anwender auf circa 1000.

Oktober: Version 0.98.2

1993 Die Zahl aller Programmierer steigt auf circa 100. Fünf von ihnen arbeiten mit Linus zusammen. Die Anwenderzahl beläuft sich auf etwa 20000.

Durch Anpassung des Linuxkernels an die GNU Umgebung der Free Software Foundation (FSF) wuchsen die Möglichkeiten von Linux erneut stark an, da man nun auf eine große Sammlung an vorhandener Software und Tools zurückgreifen konnte.

Dezember: Version 0.99.14

1994 Mit der ersten stabilen Linux Version (1.00) im März wurde der Kernel netzwerkfähig und die User-Zahl steigt auf 100.000 an. Linus stellt nun den Quelltext des Linuxkernels offiziell unter die GPL.

Ein weiterer wichtiger Schritt war die Adaption eines Graphical User Interfaces (GUI), des X-Window-Systems. Dieses wurde von dem Xfree86 Projekt beigesteuert.

April: Version 1.10

1995 Linux läuft nun auch auf DEC- und Sun Sparc-Prozessoren. Schätzungen belaufen sich auf rund eine halbe Millionen Anwender.

März: Version 1.20

Juni: Version 1.30

1996 Mit der neuen Version 2.0 des Linux-Kernels können nun mehrere Prozessoren gleichzeitig angesteuert werden. Linux verliert langsam seinen Bastlerstatus und wird zu einer ernstzunehmenden Alternative für Firmen.

Die Anwenderzahl hat sich auf rund 1,5 Millionen Benutzern erhöht.

Juni: Version 2.00

September: Version 2.10

1997 Neue Linuxversionen erscheinen fast wöchentlich. In verschiedenen Ländern existieren bereits Linuxmagazine. Die Anwenderzahl ist auf 3,5 Millionen Anwender gestiegen.

Verschiedene namhafte Firmen beginnen ihre Software auf Linux zu portieren. Netscape ihren Webbrowser, Applixware ihre Office Anwendung und die Software AG Ihre Datenbank Adabas D. Damit gibt es auch professionelle Software für Linux.

April: Version 2.1.35

1998 Das K Desktop Enviroment (kurz: KDE) wird gestartet.

Man schätzt die Anzahl der Programmierer von Linux auf 10.000, und die Anzahl der Anwender auf 7,5 Millionen.

März: Version 2.1.90

1999 Man tippt auf ungefähr 10 Millionen Anwender. Linus Torvalds kündigt den Angriff auf Windows an.

Januar: Version 2.20

Mai: Version 2.30

2000

2001 Januar: Version 2.4

2002

1.3 Was Sie vor der Installation wissen sollten

1.3.1 Hardwarevoraussetzungen

Für ein System ohne graphische Oberfläche können Sie sich an folgenden Werten orientieren.

Theoretisches Minimum	386 SX	1 MB RAM	5MB Festplatte
Brauchbares Minimum	386 DX/40 mit Coprozessor	8 MB RAM	150 MB Festplatte
Brauchbares System	486 DX/66	16 MB RAM	600MB Festplatte
Gutes System	Pentium90	32 MB RAM	1GB Festplatte

Graphische Systeme machen erst ab einem 486 Prozessor Spaß und für ein KDE-System sollten es minimal 64 MB Arbeitsspeicher sein.

1.4 Installation SuSE 8.0

Damit wir mit Linux arbeiten können ist es notwendig das Betriebssystem zu installieren. Die Installation ist stark abhängig von der verwendeten Distribution. Bei der SuSE-Distribution kann eine Installation von CD, Festplatte oder über NFS oder FTP erfolgen. Dabei können die Installations- und Administrationstools YaST 1 und Yast 2 verwendet werden.

Seit SuSE 8.0 befindet sich das für ein Textterminal optimierte Yast 1 nicht mehr auf der Distribution. Persönlich habe ich sehr gerne damit gearbeitet, da die Handhabung für Konsolensysteme optimiert war. Das für graphische Oberflächen entwickelte Yast 2 besitzt mehr Möglichkeiten. Es kann auch unter einer nichtgraphischen Oberfläche eingesetzt werden, ist dort aber nicht sehr handlich. Da auch andere Benutzer das gleiche Problem hatten, wurde in SuSE 8.1 die Bedienung von Yast im Textmodus deutlich verbessert.

1.4.1 Erstellung der Bootdisketten unter DOS

Um mit der Installation beginnen zu können, müssen Sie den Rechner mit einem rudimentären Betriebssystem starten (Booten). Dies erfolgt in der Regel durch eine speziell dafür vorbereitete Diskette oder CD. Sie brauchen die Medien nur ins CD-Laufwerk einzulegen und den Rechner dann zu starten. In der Bootreihenfolge des BIOS muß allerdings das CD-Laufwerk bzw. Diskettenlaufwerk vor der Festplatte kommen.

Die erste CD der SuSE Distribution ist bootfähig und in der Regel kann die Installation dann ohne Probleme durchgeführt werden. Manchmal unterstützen ältere Rechner oder SCSI-Systeme das Booten von CD nicht. Dort und um eine Netzwerkinstallation durchführen zu können, werden Bootdisketten benötigt, die ein Grundbetriebssystem starten. Diese Bootdisketten können von Diskettenimages erstellt werden. Für SuSE 8.0 gibt es eine Bootdiskette (`bootdisk`) und drei Moduldisketten mit Treibern (`modulesX`). Die beigefügten Textdateien enthalten die Liste der Treiber.

Unter DOS bzw. Windows können Sie das Programm `rawrite` benutzen. Sie finden das Programm auf der SuSE-Distribution auf der CD 1 im Verzeichnis `/dosutils/rawrite`. Die Bootimages (`bootdisk`, `modules` und `modules2`) befinden sich im Verzeichnis `/disks` auf der gleichen CD. Sie können also die Bootdisketten durch die Befehle

```
R:> dosutils\rawrite\rawrite disks\bootdisk
R:> dosutils\rawrite\rawrite disks\modules1
R:> dosutils\rawrite\rawrite disks\modules2
R:> dosutils\rawrite\rawrite disks\modules3
```

auf drei 3,5”-Disketten installieren, wenn die CD im Laufwerk R: liegt. Die Disketten müssen vorher formatiert werden[2].

Auf die Installation der Bootdisketten mit Linux wird in Abschnitt 4.5.5 eingegangen.

1.4.2 Booten

Installation mit Bootdisketten

1. Legen Sie die Bootdiskette ein und starten Sie den Rechner.
2. Beim jetzt erscheinenden Auswahlmenü wählen Sie Installation.
3. Nun wird der Kernel geladen. Da er sehr groß ist, benötigen Sie auch noch die Moduldiskette 1. Nachdem der Kernel vollständig geladen wurde, wird das System gestartet.
4. Nun werden Sie aufgefordert eine Moduldiskette einzulegen.
5. Sollte die Installationsroutine keine Installations-CD finden, weil der passende Treiber fehlt oder eine Netzwerkinstallation geplant ist, dann schaltet Sie nun in die “Manuelle Installation”.
6. Wählen Sie nun die Sprache für die Installation aus. (Deutsch)
7. Wählen Sie nun die Tastaturbelegung aus. (Deutsch)
8. Im nun folgenden Hauptmenü finden Sie die Menüpunkte
   - **Einstellung**
     Hier können Sie noch mal die Sprachen für Installation und Tastatur ändern.

[2]Die Disketten müssen low-level-formatiert sein. Die Datenstruktur wird durch das Image wieder überschrieben.

- **System-Information**
  Beim Booten analysiert der Kernel die Hardware des Systems. Über diesen Menüpunkt können diese und Informationen über den Kernel selber aufgerufen werden. Sie erhalten Informationen über die Kernel-Meldungen, Festplatten und CD-Laufwerke, geladene Module, PCI-Bus, Prozessor, Speicher, Ein- und Ausgabeports, Interrupts, Geräte, Netzwerkkarten und DMA-Kanäle.
- **Kernel-Module (Hardware-Treiber)**
  Fehlende Treiber für die Installation können hier eingebunden werden.
- **Installation / System starten**
  Jetzt kann die Installation gestartet werden oder ein bereits installiertes System gebootet werden.
- **Abbruch / Reboot**
  Installation abbrechen und den Rechner neu starten.
- **Power off**
  Installation abbrechen und den Rechner ausschalten.[3]

9. Sollte ein passender Treiber fehlen (z. B SCSI-Treiber), dann wechseln Sie in den Menüpunkt *Kernel-Module*.
   (a) Dort wählen Sie die passende Rubrik aus (z. B. *IDE/RAID/SCSI-Treiber laden* für SCSI-Geräte).
   (b) Legen Sie die angeforderte Moduldiskette ein. Die Daten werden in eine RAM-Disk geschrieben.
   (c) Wählen Sie den passenden Treiber aus und geben Sie eventuell benötigte Parameter für das Modul ein.[4]
   (d) Das System versucht das Modul nun zu laden. Auf Konsole 4 – zu erreichen über die Tastenkombination `<ALT>+<F4>` – können die Systemmeldungen angeschaut werden. Zurück geht es mit `<ALT>+<F1>`. Die Erfolgs- oder Mißerfolgsmeldung wird mit Return bestätigt.
   (e) Über den Button *Zurück* landen Sie wieder im Hauptmenü.
10. Nachdem nun alle Treiber geladen sind, geht es mit dem Menüpunkt *Installation / System starten* weiter.
11. Im Installations-Menü wählen Sie *Installation/Update starten*.
12. Nun muß das Quellmedium ausgewählt werden.

    CD-ROM Bei Auswahl dieses Menüpunkts versucht die Installationsroutine die Installations-CD bzw. DVD zu mounten. Klappt dies nicht, müssen Sie die Daten des Laufwerks per Hand eingeben.

    Netzwerk

    Festplatte
13. Nun erscheint das eigentliche Installationsmenü.

Installation von CD

Der Bootvorgang von CD ist wesentlich einfacher.

1. Starten Sie den Rechner und legen Sie die CD 1 oder die DVD während der POST-Phase ein.
2. Beim jetzt erscheinenden Auswahlmenü wählen Sie Installation. Sollte die Installation nicht klappen, da Treiber z. B. für die SCSI-Festplatte fehlen, muß die “Manuelle Installation” ausgewählt werden. Die wird wie oben beschrieben durchgeführt.
3. Nun wird der Kernel geladen und das System gestartet.
4. Danach erscheint das eigentliche Installationsmenü.

[3] Geht natürlich nur bei ATX-Netzteilen.
[4] Meistens brauche Sie keine einzugeben.

1.4.3 Das Installationsmenü

Da sich der Installationsassistent häufig ändert, möchte ich hier nicht auf die weitere Installation eingehen. Eine gute Beschreibung der Installationsroutinen finden Sie in den mitgelieferten Handbüchern.

1.5 Konfiguration mit YaST2

Im Gegensatz zu Windows, wo die Einstellungen des Systems in der nicht gerade übersichtlichen Registry gespeichert werden, existieren unter Linux viele Textdateien, die die Konfigurationswerte enthalten. So ist es möglich nur mit einem Texteditor das komplette System zu administrieren. Obwohl diese Konfigurationsdateien normalerweise gut dokumentiert sind, gehört doch etwas wissen und selbstvertrauen dazu, die Einstellung per Hand vorzunehmen.

SuSE hat für die Administration das Tool YaST2 für die graphische Oberfläche entwickelt. Es erlaubt eine wesentlich komfortablere aber nicht unbedingt schnellere Administration des Systems. Auch normale Benutzer können die YaST2 Kontrollzentrum oder die einzelnen Module aufrufen. Vor der Ausführung wird aber nach dem Passwort für *root* geführtl. Daher müssen Sie sich nicht extra als *root* anmelden um Ihr System administrieren zu können.

Zu den YaST2-Modulen gelangen Sie entweder über das KDE-Kontrollzentrum oder über das SuSE-Menü unter der Rubrik *System*.

Im YaST-Kontrollzentrum finden finden Sie die Module in verschiedenen Gruppen zusammengefaßt. Ich gehe im folgenden Abschnitt nur vereinzelt auf die einzelnen Module ein, weil die Funktionsweise vieler Module zum jetzigen Zeitpunkt zu viel weiteres Wissen erforden würde und damit den Umfang dieses Kapitels sprengen würde.

Software Dieser Abschnitt beschäftigt sich mit der Installation, Deinstallation und Update der Software auf dem System.

- **Installationsquelle wechseln** YaST merkt sich die letzte Installationsquelle. Erfolgt nun die Installation von einem anderen Medium, so kann dies hier eingestellt werden.
- **Patch CD-Update** Für bestimmte Kunden stellt SuSE in regelmäßigen Abständen CDs mit aktuellen Patches zur Verfügung. Hier können Sie eingespielt werden.
- **Update des Systems** Über dieses Modul können Sie das bestehende System auf einen neuen Distributionsstand bringen, z. B. von SuSE 8.0 auf SuSE 8.1.
- **Online-Update** Fehlerbehebung ist ein wichtiger Faktor in der EDV. Mit dem Online-Update können Sie kleine Programme zur Fehlerkorrektor (*Patches*) Online herunterladen und Ihr System auf den neuesten Stand bringen.
- **Software installieren/löschen** Dieses Modul ermöglicht es Ihnen Pakete von der SuSE-Distribution einzuspielen, upzudaten oder wieder vom System zu entfernen.

Hardware Wie der Namen schon erahnen läßt finden Sie hier die Möglichkeit die Hardware zu konfigurieren oder Informationen über Sie einzuholen. Spezielle für Drucker, Grafikkarte und Monitor, Festplatte, Scanner, Maus, Soundkarte und TV-Karte gibt es hier Module.

- **Automatische Druckererkennung** für die Installation lokaler Drucker.

Drucker bearbeiten Mit diesem Modul können existierende Drucker bearbeitet werden und neue lokale und Netzwerkdrucker eingerichtet werden.

Grafikkarte und Monitor Hiermit können Sie die Einstellung für die Grafikkarte ändern und die Ausgabe an den Monitor anpassen.

Hardware-Info Liefert Ihnen eine Übersicht über die eingebaute Hardware.

IDE DMA-Modus Hiermit können Sie den DMA-Modus (`Direct Memory Access`) Ihrer Laufwerke aktivieren. Dies ist z. B. für das Betrachten von DVDs wichtig.

Mausmodell wählen Was soll ich da noch viel schreiben ...

Partitionieren Hier gelangen Sie zu dem Tool, bei dem Sie schon bei der Installation die Festplatten Ihres Rechners in einzelne Abschnitte aufteilen können.

Scanner Auch Scanner wollen konfiguriert werden ...

Sound Bei der Soundkonfiguration sollten Sie die passende Lautstärke beachten und richtig einstellen. Aus Sicherheitsgründen ist die Grundeinstellung meistens sehr leise. Dies und ein sehr leise gestellter Lautsprecher haben schon oft zur falschen Annahme geführt, daß die Soundkarte unter Linux nicht funktioniert.

TV-Karten Seit SuSE 8.0 hat sich die Unterstützung von TV-Karten sehr verbessert. Hier werden Sie automatisch erkannt und konfiguriert.

Netzwerk/Basis Die Konfiguration von Netzwerkkomponenten (Netzwerkkarten, Modems, ISDN-Karten, etc.) oder Diensten (E-Mail, DSL, Serverdienste) wird durch die Module dieser Gruppe erledigt.

E-Mail

Konfiguration der Netzwerkkarte

Konfiguration des Modems

Konfiguration von DSL

Konfiguration von ISDN

Start oder Stopp von Systemdiensten

Netzwerk/Erweitert Die speziellen Netzwerkfunktionen für DNS, NFS, NIS und Routing werden hier konfiguriert.

HostName und DNS

LDAP-Client

NFS-Client

NFS-Server

NIS+-Client

NIS-Client

NIS-Server

Routing

Sicherheit & Benutzer Benutzerverwaltung und die Richtlinien zur Sicherheit (Firewall) werden hier abgehandelt.

Benutzer bearbeiten und anlegen

Einstellungen zur Sicherheit

Firewall

Gruppen bearbeiten und anlegen

Neue Gruppe anlegen

Neuen Benutzer anlegen

System Diese Module kümmern sich um die systemnahen Einstellungen, wie Backup, Bootdisketten, Starteinstellungen, Automatisches Starten von Diensten und Grundkonfigurationen.

Backup des Systems erstellen

Boot-, Rettungs- oder Moduldiskette erzeugen

Konfiguration des Bootloaders

LVM

Runlevel-Editor

Sprache wählen

Sysconfig-Editor

Tastaturlayout auswählen

Zeitzone auswählen

Sonstiges Hier finden Sie den Rest der YaST-Module, die SuSE nicht in eine der anderen Gruppen einordnen konnte.

Drucker für CUPS Konfiguriert Drucker für das Drucksystem CUPS

Drucker für LPD Konfiguriert Drucker für das Drucksystem LPD

Startprotokoll anzeigen Ermöglicht die Anzeige der einzelnen Systemlogdateien, wobei bei diesem Modul mit `/var/log/boot.msg` gestartet wird.

Support-Anfrage stellen Hier können Sie eine Anfragen an den Online-Support stellen.

Systemprotokoll anzeigen Ermöglicht die Anzeige der einzelnen Systemlogdateien, wobei bei diesem Modul mit `/var/log/messages` gestartet wird.

Treiber-Cd des Herstellers laden Ermöglicht die Installation von Treibern, die vom Hersteller der Hardware kommen.

Der folgende Abschnitt zeigt Ihnen die Administration einiger Funktionen. Diese ist im Prinzip über YaST sehr einfach. In den meisten Fällen muß man nur wissen was man will[5] und man muß lesen können.[6]

1.5.1 Wie installiere ich weitere Softwarepakete?

Das es um Software geht, wählen Sie im YaST-Kontrollzentrum die Gruppe *Software*. Ein Blick auf die Liste der Module zeigt uns, daß `Software installieren/löschen` wohl eine gute Wahl ist. Falls Sie die Installationsquelle gewechselt haben, müssen Sie diese vorher mit dem Modul *Installationsquelle wechseln* neu einstellen.

Nach dem Starten des Moduls sucht YaST nach der Paketliste. Dies kann einen Moment dauern. Findet YaST kein Installationsmedium, so behilft er sich mit der letzten gespeicherten Liste.

Achtung! Installation über NFS: Sollte sich YaST bei dieser Phase aufhängen und nicht weitermachen, dann liegt es in den meisten Fällen daran, daß der NFS-Server nicht erreichbar ist. Da es sich um eine harte Verbindung handelt, dauert es sehr lange bis es zu einem Timeout kommt und YaST den Fehler bemerkt. Um diesen Fehler zu umgehen, wählen Sie vorher ein anderes Installationsmedium aus.

Nach dem Laden der Paketlisten erscheint das Auswahlfenster für die Pakete (Abb. 1.1). Sie sehen dort drei Tabellen. Links oben zeigt Ihnen die Themengruppen oder Paketserien an, in die die Pakete einsortiert worden sind. Unten links sehen Sie, wieviel Platz noch auf der Festplatte vorhanden ist. Die rechte Tabelle zeigt die Pakete in der Gruppe an. Durch Doppelklicken oder Betätigen des Buttons *Aus-/Abwählen* können Sie den Zustand des Pakets ändern. Dieser Zustand wird durch ein Zeichen vor dem Paketnamen angezeigt. Dabei steht ein **X** für ein Paket, das zur Installation vorgesehen ist. Das kleine **i** steht für ein bereits installiertes Paket und das kleine **d** für ein installiertes Paket, daß deinstalliert werden soll. Auch ein kleines **a** kann als Markierung vorkommen. Diese Pakete wurden automatisch ausgewählt zur Installation, da diese von anderen

[5] Das ist nicht immer so einfach!

[6] Was meistens noch schwerer ist ...

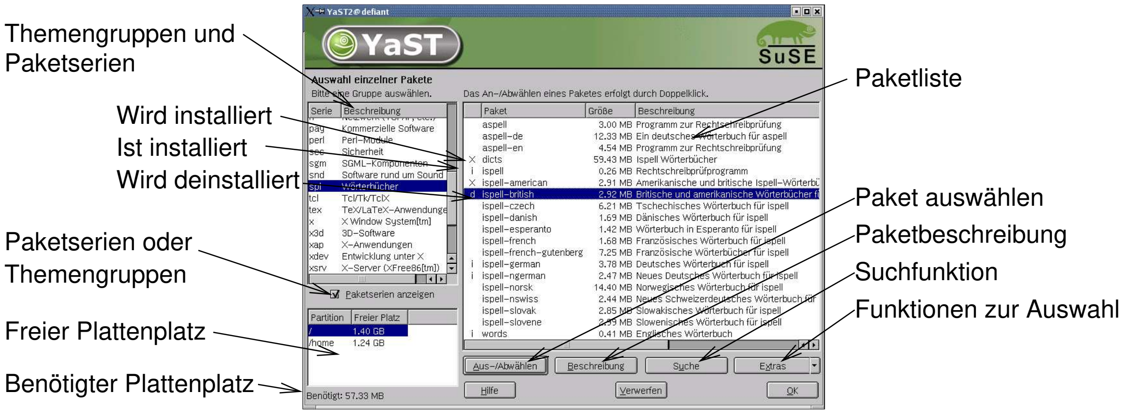

Abbildung 1.1: YaST-Kontrollzentrum: Fenster zur Paketauswahl

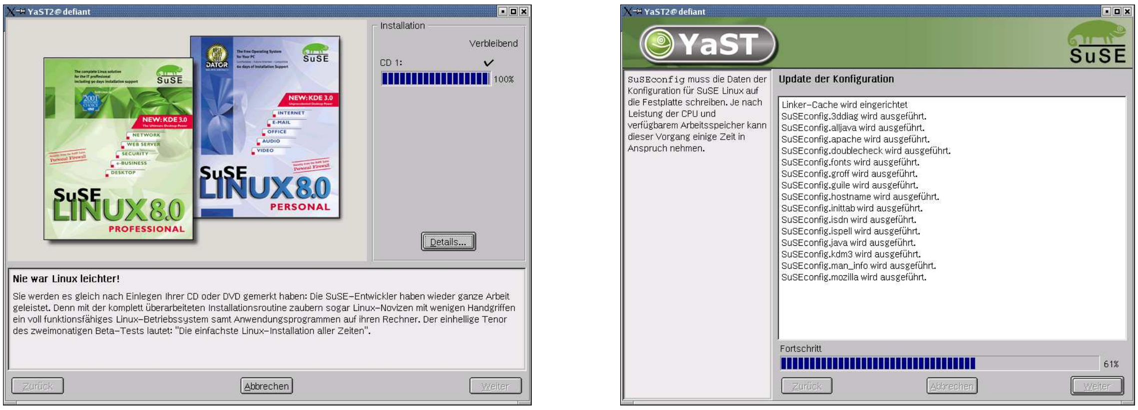

Abbildung 1.2: YaST-Kontrollzentrum: Nach der Installation der Pakete (*linkes Bild*) sorgt YaST dafür, daß die entsprechende Konfigurationsdateien angepaßt werden. (*rechtes Bild*)

Paketen benötigt werden. Keine Markierung bedeutet natürlich, daß dieses Paket nicht installiert und auch nichts für dieses Paket geplant ist.

Zu den meisten Paketen existiert neben der Kurzbeschreibung in der Tabelle noch eine etwas ausführlichere Beschreibung des Pakets. Diese erreichen Sie über den Button *Beschreibung*, was Sie natürlich nie erraten hätten. `;-)`

In vielen Fällen kennt man den Namen des Programms, aber nicht in welchem Paket oder in welcher Themengruppe/Paketserie das Paket liegt. In den meisten Fällen heißt das Programmpaket genau wie das Programm. Über eine Suchfunktion (Button *Suchen*) können die Paketnamen und sogar die Beschreibungen nach einem Stichwort durchsucht werden.

Unter *Extras* finden Sie weitere Funktionen um Pakete auszuwählen oder abzuwählen. Besonders interessant ist die Möglichkeit, die aktuelle Paketauswahl abzuspeichern oder eine gespeicherte Paketauswahl einzuladen. Dadurch ist es relativ schnell möglich Rechner mit der gleichen Softwareausstattung zu installieren.

Nach der Paketauswahl, die Sie mit dem Button *OK* abschließen, werden die Pakete installiert. Bei der CD-Version werden Sie dann und wann aufgefordert die entsprechenden CDs einzulegen. Mit einem Fortschrittsbalken werden Sie darüber informiert, wie weit Ihre Installation ist. Nach Abschluß der Installation startet YaST das Programm `SuSEConfig`, daß die Konfigurationsdateien des Systems für die neuen Programmen konfiguriert.

Und schon sind die Programme installiert.

1.5.2 Wie mache ich ein Online-Update?

Keine Software ist fehlerfrei! Diesem gerne von einer Firma aus Redmond als Entschuldigung herangezogenen Satz kann ich nur zustimmen. Wer selber mal ein längeres Programm geschrieben hat, weiß, wieviel Fehler sich einschleichen können. Und wir reden hier von Programmen, deren Sourcecode ausgedruckt mehrere Aktenordner füllen würden. Programme *sind* fehlerhaft. Finden Sie sich damit ab.

Zum Glück gibt es so Leute, die sich nicht damit abfinden und die Fehler in Programmen bereinigen. Bei OpenSource-Software kann dies theoretisch jeder. Wohlgemerkt: theoretisch. Um sich in einem großen Programm zurechtzufinden braucht es schon etwas längere Einarbeitungszeit und gute Kenntnisse in den Programmiersprachen und natürlich der Programmierung selber.

Jedenfalls werden viele Fehler bzw. Sicherheitslücken im Laufe der Zeit in Programmen entdeckt und oft auch behoben. Um nicht das ganze Programm neu einspielen zu müssen, benutzt man sogenannte Patches (Flicken), die den Binärcode des "alten" Programms auf den neuesten Stand bringen. Die Fehlerkorrekturen – manche Leute reden auch von Fehlerveränderungen – können Sie manuell einspielen. SuSE bietet Ihnen den Service eines Online-Updates. Das YOU (*YaST Online Update*) stellt Ihnen die neuesten Patches zu den SuSE-Paketen zur Verfügung. Es stellt fest, was an neuen Patches vorhanden ist. Dann lädt es die Patches für die installierten Programme runter und installiert Sie.

Das YOU finden Sie auch im Kontrollzentrum unter *Software*. Nach dem Starten des Moduls sucht das System nach verfügbaren Update-Servern. Sie müssen natürlich auch eine Verbindung zum Internet besitzen.

Sie können nun zwischen manuellem und automatischem Update wählen. Entscheiden Sie sich für das manuelle Update, denn dort können Sie besser sehen, was installiert wird. Sie können auch den Server wählen, von dem Sie die Patches beziehen möchten. Dies sollten Sie tun, falls der erste Server überlastet ist.

Nun lädt YOU die Liste der zur Verfügung stehenen neuen Updates herunter. Nachdem die Verbindung beendet wurde, geht es mit *Weiter* weiter.

Sie sehen nun die Liste der zur Verfügung stehen Patches mit Erläuterungen. Die Patches, die Sie benötigen sind schon mit einem X gekennzeichnet zum Download vorgesehen. Sie können nun einfach mit *Weiter* zum nächsten Schritt, dem Herunterladen der Pakete, gehen.

Nachdem dieses auch abgeschlossen ist und Sie mit *Weiter* zum nächsten Bildschirm gewechselt sind, kann es nun endlich mit dem Einspielen der Patches losgehen. Danach wird noch einmal SuSEconfig gestartet um die Konfigurationsdateien anzupassen und schon ist Ihr System auf dem neuesten Stand.

1.6 Benutzer- und Gruppenverwaltung

Linux ist ein Multiuser-Betriebssystem. Dadurch ist es mehreren Personen nacheinander oder sogar gleichzeitig möglich auf einem Rechner arbeiten. Damit es zu keinem Durcheinander der Daten kommt, muss sich jeder Benutzer eindeutig identifizieren, wenn er mit Linux arbeiten möchte. Außerdem gehört jeder Benutzer mindestens einer Gruppe an. Jeder Benutzer bekommt einen eindeutigen Namen und damit er sich auch authentifizieren kann ein Passwort.

Für die Benutzerverwaltung finden wir in der Gruppe *Sicherheit & Benutzer* die Module *Benutzer bearbeiten und anlegen*, *Neuen Benutzer anlegen*, *Gruppen bearbeiten und anlegen* und *Neue Gruppen anlegen*.

Diese Module sind Frontends zu der eigentlichen Benutzerverwaltung von Linux. Ein Frontend ist ein Programm, das eigentlich nichts selber macht, sondern sich als Bedienungselement zwischen den Benutzer und dem eigentlichem Programm setzt. Das Ziel eines Frontends ist es, die Bedienung von Programmen und Konfigurationsdateien zu erleichtern und damit komfortabler und fehlerfreier zu machen.

Die Konfigurationsdateien für die Benutzerverwaltung befinden sich wie die meisten Konfigurationsdateien im Verzeichnis `/etc`. Die Liste der Benutzer ist in der Datei `/etc/passwd` abgelegt. Die Passwörter befinden sich in der Datei `/etc/shadow`. Deren Aufbau und Arbeitsweise wird in einem späteren Kapitel behandelt.

1.6.1 Wie erstelle, bearbeite und lösche ich Benutzer?

Die Module *Benutzer bearbeiten und anlegen* und *Neuen Benutzer anlegen* sind für die Benutzerverwaltung verantwortlich.

Neuen Benutzer anlegen

Um einen neuen Benutzer anzulegen, starte Sie das Modul *Neuen Benutzer anlegen*. Im nun erscheinenden Fenster können Sie den Benutzernamen und das Passwort des neuen Benutzers eintragen.

Bei der Eingabe eines Passworts müssen Sie zwischen Groß- und Kleinschreibung unterscheiden. Ein Passwort sollte mindestens fünf Zeichen lang sein und darf keine Sonderzeichen (z. B. Akzente) enthalten. Sie können für die Passwörter Buchstaben, Zahlen, Leerzeichen und die Zeichen `#*,.;:._-+!$%&/|?{[()]}.` verwenden. Es ist ausreichend, ein Passwort mit der Länge von maximal acht alphanumerischen Zeichen zu wählen. Alle darüber hinausgehenden Zeichen werden bei der normalerweise eingestellten Crypt-Verschlüsselungen ignoriert. Das Passwort müssen Sie zweimal eintragen, da es aus Sicherheitsgründen nicht angezeigt wird. So können die meisten Tippfehler verhindert werden.

Für die Benutzernamen und Passwörter gibt es Regeln. So darf der Benutzername nur aus Kleinbuchstaben, Ziffern, "-" und "_" bestehen und muß mit einem Buchstaben oder "_" beginnen. Dies ist aber nur eine Vorgabe von YaST. Linux selber erlaubt auch Großbuchstaben und andere Zeichen für den Benutzernamen.

Die Felder Vorname und Nachname sind eigentlich Spaß. Die entsprechenden Daten werden nur in das Kommentarfeld des Benutzers in der Datei `/etc/passwd` eingetragen.

Hinter dem Button *Details* verbergen sich weitere Einstellungsmöglichkeiten wie Home-Verzeichnis oder die Benutzerkennung (*UID*).

Jedem Benutzer bekommt eine eindeutige Benutzerkennung (*UID*) zugewiesen. Für normale Benutzer sollte eine UID größer als 499 verwendet werden, da die kürzeren UIDs vom System für spezielle Zwecke und Pseudo-Benutzernamen verwendet werden.

Benutzer können durch eine Mitgliedschaft in einer Gruppe weitere Rechte bekommen. Zu jedem Benutzer gehört eine Standardgruppe in der er Mitglied ist. In dem Feld *Standardgruppe* können Sie eine Gruppe aus der Liste aller in Ihrem System vorhandenen Gruppen wählen.

Zusätzliche Gruppenzugehörigkeiten können sie über das entsprechende Feld dem Benutzer zuweise. Hier wird angezeigt, zu welchen Gruppen der Benutzer noch gehört.

Der Benutzer kann auch ein Heimatverzeichnis (*Home Directory*) besitzen um seine Daten zu dort zu speichern. Normalerweise ist dies `/home/BENUTZERNAME`.

Für den Betrieb auf der Konsole (Textbetrieb) wird eine Login-Shell (der sog. Kommandozeileninterpreter) benötigt. Wählen Sie im entsprechenden Feld eine Shell aus der Liste der auf Ihrem System installierten Shells aus.

Benutzer bearbeiten und anlegen

Eine Übersicht der angelegten Benutzer liefert das Modul *Benutzer bearbeiten und anlegen*. Beim Start werden nur die normalen Benutzer angezeigt. Sie können sich aber auch die für interne Zwecke (z. B. Serverdienste) angelegten Systembenutzer anzeigen lassen. Durch anklicken der Spaltenüberschriften können Sie die Sortierung der Spalten ändern.

In diesem Fenster stehen Ihnen drei Funktionen zur Verfügung:

- Einen neuen Benutzer anlegen (*Hinzufügen*),
- den markierten Benutzer bearbeiten (*Bearbeiten*) und
- den markierten Benutzer löschen (*Löschen*).

Wenn Sie den Button *Hinzufügen* betätigen landen Sie im gleichen Fenster wie in dem Module *Neuen Benutzer anlegen*.

Auch nachdem Sie den Button *Bearbeiten* landen Sie im gleichen Fenster wie in dem Module *Neuen Benutzer anlegen*. Nur daß die Felder nicht leer sondern mit den Daten des zu bearbeitenden Benutzers gefüllt sind.

Wenn Sie den markierten Benutzer löschen wollen, dann betätigen Sie einfach den Button *Löschen*. Sie werden dann noch gefragt, ob das Heimatverzeichnis des Benutzers auch mit gelöscht werden soll. Wenn Sie sich dann fürs Löschen entscheiden ist es vorbei mit dem Benutzer.

Vorbei? Nicht ganz. Die Änderungen werden erst dann ins System übertragen, wenn Sie mit dem Button *Beenden* das Modul verlassen. Das Verlassen über den Button *Abbrechen* läßt das System unberührt.

1.6.2 Wie erstelle, bearbeite und lösche ich Gruppen?

Um bestimmten Mengen von Benutzern mehr Rechte zu geben, wurden die Gruppen entwickelt. Benutzer können Mitglied in einer Gruppe sein und haben dann die Rechte, die der Gruppe zugewiesen wurde, zusätzlich zu Ihren eigenen Rechten.

Die Module *Gruppen bearbeiten und anlegen* und *Neue Gruppen anlegen* sind für die Verwaltung der Gruppen zuständig.

Neue Gruppen anlegen

Nach dem Start des Moduls *Neue Gruppen anlegen* erscheint ein Fenster mit Einträgen für den Gruppennamen, die Gruppenkennung, dem Passwort und den Mitgliedern für die Gruppe.

Der Name der Gruppe sollte nicht zu lang sein. Normalerweise werden 1-10 Zeichen verwendet. Ansonsten gelten die gleichen Regeln wie für die Benutzernamen.

Wie auch die Benutzer haben die Gruppe eine interne Kennung. Diese GID liegt irgendwo zwischen 0 und 60000. Einige der IDs werden bereits während der Installation vergeben. YaST2 gibt eine Warnmeldung aus, wenn Sie aus Versehen eine bereits vergebene verwenden.

Auch Gruppen haben ein Passwort. Falls die zugehörigen Benutzer der Gruppe sich beim Wechseln in diese neue Gruppe identifizieren sollen, können Sie dieser Gruppe ein Passwort zuteilen. Aus Sicherheitsgründen wird dieses Passwort hier nicht angezeigt. Dieses Eingabefeld muss nicht ausgefüllt werden. Um Tipfehler zu vermeiden, müssen Sie das Passwort zweimal eingeben.

Das Fenster *Mitglieder dieser Gruppe* zeigt eine Liste aller angelegten Benutzer an. Hier können Sie einstellen, wer Mitglieder dieser Gruppe sein soll.

Gruppen bearbeiten und anlegen

Das Modul *Gruppen bearbeiten und anlegen* zeigt eine große Ähnlichkeit mit dem Modul *Benutzer bearbeiten und anlegen*. Kein Wunder, es ist dasselbe Programm. Ein Klick auf *Benutzerverwaltung* und schon sind Sie in dem bereits bekannten Fenster. Dementsprechend gibt es hier auch drei Funktionen.

- Eine neue Gruppe anlegen (*Hinzufügen*),
- die markierte Gruppe bearbeiten (*Bearbeiten*) und
- die markierte Gruppe löschen (*Löschen*).

Beim Anlegen und Bearbeiten einer Gruppe landen Sie im gleichen Fenster wie im Modul *Neue Gruppen anlegen*.

Das Löschen von Gruppen gestaltet sich etwas schwieriger, da erst alle Benutzer aus der Gruppe entfernt werden müssen, bevor sich die Gruppe löschen läßt.

Auch hier gilt, alle Änderungen werden erst nach dem Verlassen des Moduls über den Button *Beenden* in das System übertragen.

Notizen:

I	Arbeitsblatt Erste Schritte	1.1

Die Ausführung der folgenden Aufgaben ist wichtig für alle folgenden Aufgaben. Führen Sie daher bitte alles aus. Eventuell haben Sie die Aufgaben schon begleitend zum Unterricht bzw. zur Lektüre gemacht.

1 Ermitteln Sie die Hardwarekonfiguration des Rechners (Graphikkarte, Festplatte, Netzwerkkarte, Monitor).

2 Ermitteln Sie die IP-Nummer, die Subnetzmaske, den Nameserver, den Namen, und das Standardgateway des Rechners an dem Sie arbeiten.

3 Beginnen Sie mit der Installation durch YaST2. Beschreiben Sie ausführlich alle Schritte der Installation und erstellen Sie damit ein "Kochrezept" für die zukünftigen Installationen. Dabei soll folgendes berücksichtigt werden:

1. Legen Sie folgende Partitionen an:
   / mit 5 GB als primäre Partition (ext3)
   /**home** mit 3 GB als logisches Laufwerk (ext3)
   swap mit 256 MB als logisches Laufwerk
2. Für die Paketauswahl wählen Sie KDE mit Office.
3. Richten Sie das Netzwerk mit den übergebenen Werten ein.

4 Sie besitzen einen Rechner, bei dem das CD-Laufwerk über einen Adaptec SCSI-Controller 2940 läuft. Auf welchem Moduldiskettenimage befindet sich der dafür benötigte Treiber?

5 Welche Installationsmöglichkeiten (Quelle) können verwendet werden?

6 Legen Sie mit `YaST2` den Benutzer *walter* mit der UID 510[7] und dem Paßwort *hallo* an.

7 Installieren Sie mit `YaST2` aus der Serie *ap* das Paket *pdksh*.

8 Installieren Sie mit `YaST2` die Pakete *stat*, *rman*, *locate* und *koffice* nach. In welchen Serien liegen sie und wofür sind sie da?

9 Speichern Sie die aktuelle Paketkonfiguration im Heimatverzeichnis von *root* (`/root`) unter dem Namen `config_kap1.usr` ab.

10 Löschen Sie die Pakete *stat*, *rman*, *locate* und *koffice*.

11 Laden Sie nun die alte Paketkonfiguration aus Aufgabe 9 ein. Überprüfen Sie, ob die gerade gelöschten Pakete nun wieder zur Installation vorgesehen sind.

12 Installieren Sie wieder die Pakete *stat*, *rman*, *locate* und *koffice*.

[7] Die UID ist die interne Benutzernummer für das System. Siehe auch Abschnitt 8.2.1.

Notizen:

Kapitel 2

X-Window und KDE

2.1 X-Window-System

Bei der Installation von SuSE-Linux in der oben genannten Konfiguration wird standardmäßig das X-Window-System mit dem KDE installiert. Das X-Window-System kann am besten durch die Tools `sax` bzw. `sax2` konfiguriert werden. Einstellungen über Graphikkarte, Monitor, Auflösung, Tastatur und Maus erfolgen hier.

Für die Bedienung und Arbeit am X-Window-System wird ein Window-Manager benötigt. Er kümmert sich in erster Linie darum, wie die Programmfenster am Bildschirm dargestellt und bedient werden. Er stattet die Fenster mit einer Titelleiste und Schaltflächen (Buttons, Menüs etc) aus und hilft bei deren Verwaltung (Verschieben, Größe ändern, Verkleinern zu Icon, Programmwechsel). Daneben stellen die meisten Windowmanager Menüs für die gängigen Programme bereit.

Diese elementaren Funktionen werden nicht vom X-Window-System zur Verfügung gestellt. Der Vorteil dieser Methode ist die große Flexibilität bei der Gestaltung der Oberfläche und damit der Benutzerschnittstelle. Das hat aber auch den Nachteil, daß jeder Anwender einen anderen Windowmanager benutzt und damit keine einheitliche Oberfläche existiert. Ein Problem, daß vor allem die Anfänger betrifft.

Die aktuelle SuSE-Distribution startet automatisch das X-Window-System für die graphische Anmeldung. Wird das System mit der Textkonsole gestartet, kann von der Textkonsole aus das X-Window-System mit dem Befehl `startx` gestartet werden.

Sollte das X-Window-System sich aufhängen, kann es mit der Tastenkombination `STRG + ALT + BACKSPACE` beendet werden.

2.2 KDE

Das KDE (KDE Desktop Enviroment) ist eine Sammlung von X-Programmen mit einem Window-Manager für das X-Window-System. Es stellt wie auch die anderen Windowmanager (z. B. Gnome oder fvwm) die Schnittstelle zwischen Benutzer und dem X-Window-System dar. Das KDE umfaßt dabei nicht nur die Konfigurationstools sondern auch eine Vielzahl von Standardprogrammen.

Noch einige Hinweise zu KDE.

- Doppelklicks sind in KDE verpönt und sind nur in Ausnahmefällen erforderlich.
- Mit der rechten Maustaste können Sie in den meisten KDE-Programmen ein Kontextmenü aufrufen.
- Fast alle KDE-Programme kommen mit Drag & Drop zurecht.
- KDE-Programme können in der Regel direkt über einen Menüpunkt konfiguriert werden. Bei einigen Programmen erfolgt die Konfiguration über das KDE-Kontrollzentrum.

2.2.1 KDE-Kontrollzentrum: KControl

Das KDE-Kontrollzentrum (KControl) ist das zentrale Konfigurationsprogramm fürs KDE. Grundlegende Eigenschaften wie Sprache, Desktopdesign, Sound u. a. können hier eingestellt werden.

2.2.2 KDE-Systemüberwachung: KSysGuard

Die KDE-Systemüberwachung zeigt Ihnen Informationen über das System und die darauf laufenden Prozesse an. Sie können im KDE-Menü über *KDE/System/Systemüberwachung* das Programm starten. Alternativ geben Sie über *KDE/Befehl ausführen* den Befehl `ksysguard` ein.

Das Programm präsentiert sich nach dem Start mit einem fast leeren Feld links und vier Graphen rechts. Warten Sie einen Moment und Sie sehen langsam von rechts die aktuellen Werte des Systems in die Graphen wandern.

Bedeutung der Graphen

CPU Load Der Graph zeigt die Auslastung der CPU an. Drei Kurven können sie erkennen. Blau ist die Auslastung des Prozessors durch Benutzeraktionen, Rot die Auslastung durch Systemprozesse und Hellbraun die Auslastung durch Programme, die mit einer veränderten Priorität gestartet wurden.

Load Average 1min Die auf eine Minute gemittelte Auslastung des Systems wird in diesem Graphen dargestellt. Da die CPU einem Programm immer die volle Leistung zur Verfügung stellt wenn nichts anderes anliegt, kommt es immer wieder zu CPU-Auslastungen von 100%. Dies können aber auch nur einzelne Peaks sein. Die über eine Minute gemittelte Auslastung gibt mehr Aufschlüsse über die Dauerbelastung eines Rechners.

Physical Memory Die Auslastung des physikalischen Speichers ist ebenso wie die CPU-Leistung ein Kriterium für die Belastung des Rechners. Auf den ersten Blick scheint es so, als ob der Speicher, selbst wenn kaum Programme auf dem System laufen, völlig belegt ist. Das stimmt auch soweit, da der Kernel den freien Speicher als Cache für Laufwerkszugriffe verwendet. Es werden nämlich auch hier drei Graphen angezeigt. Blau zeigt den Speicherbedarf der Programme an, dunkles Orange den Bedarf des Puffers, und Hellbraun ist die Farbe für den Festplattencache.

Swap Memory Neben dem physikalischen Speicher gibt es noch den Swap-Speicher, der als Auslagerungsdatei auf einem Datenträger realisiert ist. Beide zusammen ergeben den virtuellen Speicher, der dem System zur Verfügung steht. Hier heißt es genau zu beobachten. Denn wenn sich auf diesem Graphen viel tut, sollte man überlegen, ob nicht etwas mehr physikalischer Speicher dem Rechner gut tun würde.

Sensor-Browser

Im linken Fenster finden Sie für den lokalen Rechner die möglichen Sensoren, deren Werte Sie in den rechten Graphen anzeigen können. Einfach den Sensor anklicken und ihn in den entsprechenden Graphen ziehen. Einen Graphen können Sie übrigens löschen, indem Sie mit der rechten Maustaste darauf klicken und *Anzeige entfernen* betätigen.

Über die genaue Bedeutung der einzelnen Anzeige möchte ich jetzt nicht eingehen, da dies den Umfang dieses Kapitels sprengen würde.

Prozesstabelle

Neben der Systemauslastung können Sie sich auch die laufenden Prozesse (Programm bzw. Programmteile) anzeigen lassen. In einer Tabelle werden für die Prozesse der Name, die Prozessidentifikationsnummer (PID), die Systemauslastung durch Benutzer und das System, die Priorität des Prozesses, der Speicherbedarf, der ausführende Benutzer und der aufrufende Befehl angezeigt. Die Bedeutung der einzelnen Spalten ergibt sich im Laufe dieses Skriptes. Hier an dieser Stelle seien Sie nur erwähnt.

Sie können auch eine Auswahl der anzuzeigenden Prozesse treffen. Sie können sich entweder *Alle Prozesse*, die laufen, sich anzeigen lassen oder sich auf die *Systemprozesse* beschränken. Auch können Sie sich nur mit den Benutzer abgeben und sich die `Benutzerprozesse` anzeigen. Wenn Sie ganz genügsam sind, begnügen Sie sich mit *Eigene Prozesse*.

Wenn Sie wissen wollen, welcher Prozess durch welchen Prozess gestartet wurde, dann wählen Sie die *Baum-*Ansicht. Den neuesten Stand der Prozessliste können Sie mit dem Button *Neu Aufbauen* erzeugen. Der wichtigste Button ist wohl *Beenden (kill)*, der durch seinen zweiten martialischen Namen schon anzeigt was er macht: Er tötet Prozesse. So können Prozesse, die sich sonst nicht mehr beenden lassen, gestoppt werden.

Wenn Sie ganz schnell an die Prozessliste herankommen wollen, dann betätigen Sie einfach die Tasten `<STRG>+<ESC>`. Es öffnet sich der KSysGuard mit der Ansicht der Prozeßtabelle. Nun können Sie schnell einen Prozess töten oder sich über das Verhalten der Prozesse informieren.

2.3 Wie mache ich was?

2.3.1 Wie lege ich ein Programmicon auf dem Desktop an?

Am einfachsten können Sie ein Icon anlegen, wenn das Programm schon einen Eintrag im KDE-Menü besitzt. Mit der linken Maustaste auf den Punkt klicken und mit gedrückter Maustaste auf den Desktop ziehen. Dort dann im nun erscheinenden Menü auf *An diese Stelle kopieren* klicken. Fertig.

Wollen Sie ein Programm auf dem Desktop verewigen, was nicht im Menü erscheint, dann müssen Sie wie folgt vorgehen.

1. Rechte Maustaste auf den Desktop
2. Menüpunkt *Neu erstellen/Verknüpfung mit Programm ...* anklicken
3. Im Reiter *Allgemein* den Namen der Verknüpfung eingeben
4. Durck Klicken auf das Zahnrad-Symbol können Sie ein Symbol für das Programm auswählen
5. Im Register *Ausführen* geben Sie den Namen der Programmdatei an. Der Button `Durchsuchen` ermöglicht Ihnen das Dateisystem nach der Datei zu durchsuchen. Die meisten X-Window Programme finden Sie im Verzeichnis `/usr/X11R6/bin`, `/opt` und `/opt/kde3/bin`.
6. Mit *OK* bestätigen und fertig ist das Icon.

2.3.2 Wie kann ich ein Programm unter einem anderen Benutzer ausführen?

Manchmal ist es notwendig bestimmte Aufgaben unter einer anderen Benutzerkennung auszuführen, wie z. B. das Editieren von Systemkonfigurationsdateien. Auf diese Dateien können Sie aber als normaler Benutzer nicht zugreifen. Normalerweise müßten Sie sich jetzt als Benutzer ausloggen und wieder als priviligierter Benutzer (z. B. *root*) sich wieder einloggen. Das es auch anders geht, haben Sie beim Einsatz von YaST gesehen. Nach Eingabe des *root*-Passworts konnten Sie ohne Ab- und Anmeldung mit YaST arbeiten. Verantwortlich dafür ist das Programm `kdesu` (2.3.3).

Wenn Sie nun ein Programm als *root* starten wollen, müssen Sie im X-Terminal oder über *Befehl ausführen* (`<ALT>+<F2>`) einfach `kdesu PROGRAMM` eingeben. Also für den Editor NEdit z. B.
`kdesu nedit`.

Wer den Befehl häufiger braucht, kann sich einen Menüeintrag oder Icon auf dem Desktop anlegen. In den Eigenschaften des Eintrags/Icons befindet sich der Abschnitt *Als anderer Benutzer ausführen*. Hier einfach die Checkbox anklicken und den gewünschten Benutzernamen eintragen.

2.3.3 KDE su

KDE su erlaubt es X-Window-Programme unter einer anderen Benutzeridentität auszuführen als die des aktuellen Benutzers. Es ist ein graphisches Frontend für `su` (8.1.2) unter dem KDE. Dabei werden die für das X-Window benötigten Einstellungen entsprechend gesetzt.

```
kdesu [OPTIONEN] [QT-OPTIONEN] [KDE-OPTIONEN] KOMMANDO
```

Optionen	
`--help`	Grundoptionen
`--help-qt`	Spezielle Optionen zu Qt anzeigen
`--help-kde`	Spezielle Optionen zu KDE anzeigen
`--help-all`	Alle Optionen anzeigen
`-c BEFEHL`	Auszuführender Befehl
`-f DATEI`	
`-u USER`	Benutzerkennung angeben (Standard: *root*)
`-n`	Passwort nicht speichern
`-s`	Dämon anhalten (Alle Passwörter gehen verloren)
`-t`	Terminal-Ausgabe ermöglichen (keine Speicherung von Passwörtern)
`-p PRIO`	Priorität setzen: 0 <= prio <= 100, 0 ist die niedrigste. (Standard: 50)

Beispiel
Um den Editor NEdit unter dem Benutzer *wwwrun* auszuführen, müssen Sie folgenden Befehl anwenden.

```
kdesu -u wwwrun -c nedit
```

2.4 KDE-Programme

Inzwischen gibt es eine Reihe von Programmen, die speziell für das KDE geschrieben worden sind. Sie basieren auf den KDE-Bibliotheken und benötigen diese zwingend.

2.4.1 Konquerer

Der Konquerer ist nicht nur der mit dem KDE mitgelieferte Browser sondern auch gleichzeitig der Dateimanager. Da er relativ neu ist, hat er noch längst nicht alle Internet-Standards (besonders JavaScript) völlig implantiert.

2.4.2 KEdit und KWrite

Auch an Editoren wurde beim KDE gedacht. KEdit und KWrite sind hierfür entwickelt worden. KEdit ist die einfache Version des Editors, während KWrite neben Syntaxhighlighting auch Suchen und Ersetzen mit regulären Ausdrücken beherrscht.

2.4.3 KMail

Das Programm KMail ist, wie der Name es schon sagt, der Mail-Client vom KDE.

2.4.4 KSnapshot

Für Dokumentationen ist es oft sinnvoll einen Bildschirmschnappschuß machen zu können. Diese Funktion wird durch das Programm KSnapshot zur Verfügung gestellt.

2.4.5 KOffice

KOffice ist das fürs KDE entwickelte Office-Paket. Es enthält schon viele Programme, die aber entwicklungsbedingt noch nicht so viele Funktionen besitzen und teilweise auch etwas instabil sind.

KWord

KWord ist die Textverarbeitung.

KSpread

Die Tabellenkalkulation des KDE-Office ist KSpread. Allerdings beherrscht es wenige Datenformate, so daß hier das StarCalc vorzuziehen ist, daß auch Excel-Formate lesen und speichern kann.

2.4.6 Weitere KDE-Programme

KNode

Newsgroups können mit dem Programm KNode bearbeitet werden. Das Lesen und Senden von Artikeln werden von diesem Client-Programm erledigt. In Abschnitt 6.2.2 wird genauer auf das Programm eingegangen.

KAB

Das KDE-Adressbuch ist eine kleine Datenbank zur Verwaltung von Adressdaten wie Telefonnummer, eMail, Adresse u.s.w. Dabei können auch mehrere Adressen für eine Person eingegeben werden. Das KAB kann auch als Adressbuch für KMail benutzt werden.

KOrganizer

Einen Zeitplaner enthält KDE ebenfalls. Mit dem KOrganizer können Sie Ihre Termine planen und sich z. B. rechtzeitig an den Hochzeitstag erinnern lassen.

KCalc

Auch das KDE bringt einen Taschenrechner mit, der sogar die Punkt-vor-Strich-Regel beherrscht.

KJots

KJots ist ein kleines Programm um einfache Notizbücher zu verwalten. Sie können mehrere Notizbücher verwalten, die wiederum benannte Seiten enthalten in denen dann einfache Texte eingegeben werden können.

KNotes

Sie kennen doch sicherlich die gelben PostIt-Notizzettel. Das Programm KNotes ermöglicht Ihnen virtuelle Klebezettel auf den Bildschirm zu kleben und dort Ihre Erinnerungshilfen zu plazieren.

KBear

Auch einen komfortablen FTP-Client gibt es unterm KDE. Zwar kann man auch mit Hilfe des Konquerers Daten per FTP hin- und herschieben. Das Programm `KBear` ist durch die Arbeitsweise mit zwei Fenstern und dem Site-Manager wesentlich komfortabler.

2.5 Weitere X-Window-Programme

Neben den KDE-Programmen gibt es auch weitere nützliche Applikationen, die teils auch die KDE-Bibliotheken benutzen.

Eine hilfreiche Unterstützung bei der Erstellung von HTML-Seiten bieten die Programme **Bluefish** und **Quanta Plus**. Windows Nutzern wird die Ähnlichkeit der beiden Programme zum HTML-Editor Phase V auffallen.

Ein weiterer schöner Editor ist **NEdit**. Er benötigt kein KDE und stellt doch wichtige Funktionen wie Syntax-Highlighting und 'Suchen und Ersetzen' mit regulären Ausdrücken zur Verfügung.

II	Arbeitsblatt Erste Schritte	2.1

Die Ausführung der folgenden Aufgaben ist wichtig für alle folgenden Aufgaben. Führen Sie daher bitte alles aus. Eventuell haben Sie die Aufgaben schon begleitend zum Unterricht bzw. zur Lektüre gemacht.

1 Legen Sie mit `YaST2` den Benutzer *willi* (Willi Winzig) mit der UID 511 und dem Passwort *winzig* an. Geben Sie ihm zusätzlich noch die Mitgliedschaft in der Gruppe *disk*.

2 Loggen Sie sich aus und als *willi* wieder ein.

3 Konfigurieren Sie für *willi* das KDE so, daß der KDE-Menübutton angezeigt wird und das Eclipse-Design vom KDE verwendet wird.

4 Loggen Sie sich aus und als *walter* wieder ein.

5 Vor der ersten Verwendung einer Diskette muß diese formatiert werden. Dazu kann das Programm `KFloppy` verwendet werden. Unter welchem KDE-Menüpunkt ist es zu finden?

6 Formatieren Sie nun als *walter* die Diskette mit dem Dateisystem ext2 vollständig und geben Sie ihr den Namen “Disk”.

7 Installieren Sie mit `YaST2` aus der Gruppe *Produktivität/Editoren/Andere* das Paket *pdksh*. Wieviel Platz benötigt es und wofür ist es da?

8 In welchem Verzeichnis ist die Datei `nedit` installiert worden.

9 Legen Sie für das Programm `nedit` ein Menüeintrag unter *KDE/Editoren*. Das Programm soll durch die Tastenkombinations `<Windows>+<n>` gestartet werden. Testen Sie die Funktion.

10 Legen Sie für das Programm `nedit` ein Icon auf dem Desktop an. Testen Sie die Funktion.

11 Öffnen Sie mit NEdit die Datei `/etc/motd`. Tragen Sie statt des vorhandenen Spruchs ein: “It’s a good day to die ...” und speichern Sie die Datei wieder ab. Was passiert und warum.

12 Um sich nicht ständig als *root* anmelden zu müssen um Systemdateien zu ändern, soll `NEdit` auch als *root* Benutzer arbeiten können. Wie gehen Sie vor?

13 Führen Sie die Aufgabe 11 mit dem neuen Icon durch.

14 Testen Sie den Erfolg, indem Sie sich auf Konsole 2 als *walter* einloggen. Hat es geklappt?

15 Kehren Sie zum KDE-Desktop zurück.

16 Beenden Sie alle geöffneten Programme.

17 Legen Sie folgende Tastaturkürzel für Programme an.

Persönliches Verzeichnis	`<Windows>+<E>`
Konquerer (Webbrowser)	`<Windows>+<K>`
KMail	`<Windows>+<M>`
Yast2	`<Windows>+<Y>`
KCalc	`<Windows>+<T>`

Testen Sie den Erfolg.

18 Starten Sie den Konquerer als Webbrowser. Gehen Sie zur Seite von GMX (`http://www.gmx.de`) und erstellen dort ein neues eMail-Konto.

19 Tragen Sie in das KDE-Adressbuch Ihren Nachbarn mit seinem GMX-Konto ein.

II	Arbeitsblatt Erste Schritte	2.2

20 Welche Angaben benötigen Sie um mit KMail eMails von ihrem GMX-Konto lesen und versenden zu können?

21 Suchen Sie die Angaben auf der Webseite von GMX und notieren Sie sich die Daten auf einem KNotes-“Zettel”. Konfigurieren Sie KMail entsprechend den Angaben für das GMX-Konto.

22 Schicken Sie Ihrem Nachbarn als Test eine eMail unter Verwendung des KDE-Adressbuchs.

23 Rufen Sie mit KMail die von ihren Nachbarn zugesandten eMails ab.

24 Führen Sie mit `YaST2` erneut ein Online-Update durch. Sind neue Programme installiert worden?

Notizen:

Kapitel 3

Arbeiten auf der Shell

Wie kastriert man einen Windows-Benutzer?
Man schneidet einfach das Mauskabel durch...

Anonymous

Das KDE und damit auch das X-Window-System sind nur auf Linux aufgesetzt und dienen als Benutzeroberfläche. Es gibt aber auch eine wesentlich rudimentärere Möglichkeit für den Benutzer mit Linux zu arbeiten: die Shell (*engl. Muschel*). Andere Begriffe für die Shell sind Kommandozeile, Textmodus oder Prompt.

3.1 Starten der Shell

Die Shell steht auf jedem Linux-System zur Verfügung. Es gibt zwei Möglichkeiten mit der Shell zu arbeiten. Unter dem X-Window-System gibt es sogenannte X-Terminals, die eine solche Shell emulieren. Daneben steht die sogenannte Linux-Konsole zur Verfügung, die im Textmodus der Graphikkarte arbeitet und auch ohne X-Window-System funktioniert.

3.1.1 Linux-Konsole

Die Linux-Konsole erreichen Sie aus dem X-Window-System, indem Sie die Tasten <STRG>+<ALT>+<F1> gleichzeitig drücken. Unter SuSE ist diese Konsole sehr bunt und mit einem Hintergrund versehen. Dies ist der Bildschirm, den Sie auch beim Booten des Systems gesehen haben. Die letzten Meldungen des Bootvorgangs stehen noch dort.

```
Welcome to SuSE Linux 8.0 (i386) - Kernel 2.4.18-4GB (tty1)
```

Mit dieser Meldung, die Ihnen die Distribution und das eingesetzte Betriebssystem mit Versionsnummer und Kernelversion anzeigt, begrüßt Sie das System. Danach steht der Login-Prompt mit Rechnername und wartet darauf, daß Sie sich an dieser Konsole anmelden.

```
defiant login:
```

Geben Sie nun Ihren Benutzernamen ein und schließen Sie die Eingabe mit der <Return>-Taste oder <Enter> ab. Sie werden nun nach Ihrem Passwort gefragt. Dies geben Sie ein und schließen die Eingabe wieder mit <Return> oder <Enter> ab. Wundern Sie sich nicht, denn Linux zeigt bei der Eingabe eines Passworts generell nichts an, noch nicht einmal die bei Windows üblichen Sternchen.

```
defiant login: walter
Password:
```

```
Have a lot of fun ...
walter@defiant:~>
```

Ein netter Spruch wird ausgegeben und der Prompt erscheint. Der Prompt zeigt an, daß die Shell bereit ist von Ihnen ein Kommando zu bekommen.

Wenn Sie SuSE benutzen, stehen Ihnen sechs von diesen Konsolen zur Verfügung. Mit den Tasten `<ALT>+<F1>` bis `<ALT>+<F6>` können Sie zwischen den Konsolen wechseln. Sie sehen, daß die Konsolen 2 bis 6 einen schwarzen Hintergrund besitzen und die gleiche Begrüßungsmeldung enthalten. Dabei weißt das `tty` in der Klammer immer auf die jeweilige Konsole hin.

Wechseln Sie wieder zur ersten Konsole (`<ALT>+<F1>`). Geben Sie dort am Prompt `tty` ein und schließen Sie die Eingabe mit `<Return>` oder `<Enter>` ab. Der Befehl zeigt Ihnen nun an, auf welcher Konsole sie sich befinden.

```
walter@defiant:~> tty
/dev/tty1
```

Um die Shell zu beenden, geben Sie das Kommando `logout` ein. Denken Sie daran, daß jedes Kommando erst dann ausgeführt wird, wenn Sie es mit `<Return>` oder `<Enter>` abschließen.

```
walter@defiant:~> logout


Welcome to SuSE Linux 8.0 (i386) - Kernel 2.4.18-4GB (tty1)


defiant login:
```

Nun ist die Konsole wieder zur Anmeldung bereit. Jetzt müssen Sie nur noch zum X-Window-System zurückkehren. Das X-Window-System ist praktisch die siebente Konsole. Sie können also mit `<ALT>+<F7>` dorthin gelangen.

Wenn Sie schon auf der Konsole sind, können Sie alle Konsolen bzw. Terminals durch die Kombination von `<ALT>` und den Funktionstasten `<F1>` bis `<F6>` erreichen. Da die `<ALT>`-Taste unter KDE bzw. X-Window eine andere Bedeutung hat, können Sie hier die Konsolen über eine Kombination von `<STRG>+<ALT>` und den Funktionstasten `<F1>` bis `<F6>` erreichen.

3.1.2 X-Terminal

Für den Benutzer eines X-Window-Systems gibt es auch eine andere Methode um auf der Shell arbeiten zu können. Sogenannte X-Terminals emulieren die Linux-Konsole und bringen ein Textfenster auf den Bildschirm. Zum Starten des beim KDE mitgelieferten X-Terminals `konsole` brauchen Sie in der Menüleiste nur auf den Monitor mit der Muschel klicken.

An diesem Terminal brauchen Sie sich nicht anzumelden, da Sie sich durch die graphische Oberfläche schon in einer angemeldeten Sitzung befinden. Es erscheint also sofort der Prompt. Testen Sie doch mal, was der Befehl `tty` hier für ein Ergebnis zeigt.

```
ole@defiant:~> tty
/dev/pts/1
```

Es gibt eine ganze Reihe von X-Terminals mit z. T. sehr unterschiedlichen Fähigkeiten.

xterm

`xterm` ist der Klassiker unter den X-Terminals. Er ist sehr einfach. Sie können ihn starten, in dem Sie den Menüpunkt *Befehl ausführen ...* aus dem KDE-Menü wählen oder mit den Tasten `<ALT>+<F2>` das Ausführungsfenster aufrufen. Dort können Sie durch Eingabe von `xterm` das Fenster starten.

Mit dem Befehl `exit` können Sie das `xterm` dann wieder verlassen oder Sie drücken auf den X-Button oben rechts in der Titelleiste.

konsole

Dieses Terminal ist Bestandteil des KDE-Desktops. Genau wie das `xterm` kann es durch den Menüpunkt *Befehl ausführen ...* durch Eingabe von `konsole` gestartet werden.

Im Gegensatz zu `xterm` kann `konsole` einiges mehr. So können z. B. mehrere unterschiedliche Konsolen in einem Fenster geöffnet werden. Es stehen dann auch spezielle Versionen zur Verfügung, die gleich dem Midnight-Commander, einen textbasierenden Dateimanager, starten oder als Benutzer *root* vorgeben.

Eine weitere Konsole öffenen Sie über den Button unten links mit der Beschriftung *Neu*. Sie können natürlich auch über die Menüleiste über das Menü *Sitzung* gehen.

Unter dem Menü *Bearbeiten* können Sie u. a. den Verlauf bearbeiten. Der Verlauf ist die Sammlung der Zeilen, die auf der Konsole gestanden haben. Natürlich werden nicht alle Zeilen gespeichert. Die Anzahl der zu speichernden Zeilen finden Sie unter *Einstellungen/Verlaufspeicher*. Voreingestellt sind 1000 Zeilen, was auch erst einmal reichen sollte. Sehr nützlich ist es, daß Sie den Verlauf auch nach einem Stichwort durchsuchen können.

Im Menü *Ansicht* können Sie die laufende Konsole (Sitzung) umbenennen und auf Aktivität oder Nichtaktivität überwachen lassen. Sie können Ihre Eingaben auch zu mehreren Sitzungen gleichzeitig senden und die Reihenfolge der Sitzungen in der Sitzungsleiste ändern.

Das Menü *Einstellungen* erlaubt individuelle Einstellungen für die `konsole`. Schriftgröße, Signale, Tastatur, Farben und Größe können hier eingestellt werden.

Unter *Hilfe* finden Sie, wie bei eigentlich jedem KDE-Programm Anleitungen und Informationen über das Programm.

3.2 Erste Befehle

Um die ersten Befehle auszuprobieren, starten Sie jetzt einfach mal die `konsole`. Der Prompt erscheint und die Shell ist bereit Ihre ersten Kommandos entgegenzunehmen. Es gibt übrigens nicht *die* Shell. Es gibt eine Reihe von Shells für Unix und Linux-Systeme. Die im Linux-Bereich am meisten verwendet Shell ist die Bash. Auch bei SuSE ist sie für den Benutzer voreingestellt. Ich werde mich in diesem Text auch hauptsächlich mit Ihr beschäftigen.

Der Prompt zeigt uns schon ein paar wichtige Informationen[1]. Welcher Benutzer ist auf welchem Rechner eingeloggt und in welchem Verzeichnis befindet er sich gerade. Das Verzeichnis ~ ist natürlich kein Verzeichnis. Es ist ein Alias oder Synonym für das Heimatverzeichnis des jeweiligen Benutzers. Das können wir auch schnell mal vergleichen. Geben Sie doch einfach mal hinter dem Prompt den Befehl `pwd` ein.

```
ole@defiant:~> pwd
/home/ole
ole@defiant:~>
```

Bei dem Benutzer *ole* steht das ~ für sein Heimatverzeichnis `/home/ole`. Die Heimatverzeichnisse der Benutzer heißen normalerweise wie ihre Besitzer und liegen im Verzeichnis `/home`. Es ist aber auch möglich die Heimatverzeichnisse anders zu benennen.

3.2.1 Anzeigen des Verzeichnisinhalts

Was ist denn in diesem Heimatverzeichnis an Dateien vorhanden? Auch hierfür müssen Sie einen Befehl eingeben. Das Kommando `ls` zeigt Ihnen den Inhalt des Verzeichnis an.

```
ole@defiant:~> ls
Documents  public_html
```

Zwei Dateien liegen bei einem frisch eingerichteten Benutzer bei SuSE 8.0 im Heimatverzeichnis. Haben Sie auf der graphischen Oberfläche schon gearbeitet, so können hier noch mehr Dateien liegen.

[1] Der Prompt ist natürlich konfigurierbar und kann auch andere Informationen darstellen.

Ausführliche Informationen

Die Informationen, die der Befehl ls gibt, sind sehr spärlich. Wir können aber dem Kommando `ls` sagen, daß es mehr Informationen ausgeben soll. Dies erfolgt über eine *Option*. Andere Namen dafür sind *Schalter* oder *Attribut*. Die Option für eine ausführliche Ansicht heißt `-l`. Das Minus vor dem Buchstaben `l` (wie "long") zeigt an, daß es sich um eine Option handelt.

```
ole@defiant:~> ls -l
insgesamt 8
drwxr-xr-x    2 ole    users        4096 Sep 25 12:25 Documents
drwxr-xr-x    2 ole    users        4096 Sep 25 12:25 public_html
```

Wenn Sie sich nun die Ausgabe anschauen, dann wird Ihnen sicherlich nicht die Bedeutung jeder dieser Spalten klar sein. Schauen wir uns dazu die Ausgabe noch einmal genauer an.

Typ und Rechte	Links	Besitzer	Gruppe	Größe	Änderungsdatum	Name
drwxr-xr-x	2	ole	users	4096	Sep 25 12:25	Documents
drwxr-xr-x	2	ole	users	4096	Sep 25 12:25	public_html

Typ und Rechte Die in dieser Spalte stehende Kombination aus Buchstaben und Minus-Zeichen gibt Auskunft über den Dateityp und die Rechte, die für diese Datei gelten. Das erste Zeichen steht für den Dateityp. Das `d` in dem Beispiel steht für ein Verzeichnis. Würde ein Minuszeichen dort stehen, dann handelt es sich um eine normale Datei. Die Tabelle unten zeigt ihnen die anderen möglichen Typen. Auf die Bedeutung der einzelnen Typen möchte ich hier nicht eingehen.

Zeichen	Bedeutung
-	gewöhnliche Datei
d	Verzeichnis (directory)
c	zeichenorientierte Gerätedatei (character device)
b	blockorientierte Gerätedatei (block device)
p	FIFO-Pipeline(named pipe)
l	symbolischer Link (symbolic link)
s	Netzwerksocket (socket)

Die restlichen neun Zeichen geben in Gruppen zu jeweils drei Zeichen die Rechte auf die Datei an. Dabei steht das `r` für "Lesen" (read), das `w` für "Schreiben" (write) und das `x` für "Ausführen" (execute). Die ersten drei Zeichen geben die Rechte für den Besitzer der Datei an. Die zweiten drei Zeichen geben Auskunft über die Rechte der Gruppe, die dieser Datei zugeordnet wurde und die letzten drei Zeichen geben dann an, was der Rest der Welt mit dieser Datei darf.

Links Im Gegensatz zu Windows kann eine Datei bei Linux mehrere Namen besitzen. Bei Verzeichnissen sind das mindestens zwei. Einmal der Name selbst und dann im Verzeichnis die Datei mit dem Namen ".". Eine Datei wird erst dann gelöscht, wenn auch der letzte Name, bzw. Hardlink, wie die Dateinamen auch genannt werden, entfernt wurde.

Besitzer Hier wird der Besitzer der Datei angegeben. Wenn ein Benutzer eine neue Datei anlegt, wird er automatisch der Besitzer der Datei.

Gruppe Jeder Datei wird eine Gruppe von Benutzern zugeteilt, die extra Rechte bekommen können. Ist ein Benutzer in dieser Gruppe, so gelten die Rechte der Gruppe für ihn, wenn er nicht gerade der Besitzer der Datei ist.

Größe Die Größe der Datei wird in dieser Spalte in Byte angegeben.

Änderungsdatum Normalerweise zeigt der Befehl `ls` an dieser Stelle das Datum der letzten Änderung der Datei an. Durch bestimmte Attribute kann der Befehl auch dazu bewegt werden das Datum der Erstellung oder das Datum des letzten Zugriffs anzuzeigen.

Name Hier steht der Name der Datei. Die Sortierung der aufgelisteten Dateien erfolgt lexikalisch. Beachten Sie dabei, daß die Shell zwischen Groß- und Kleinschreibung unterscheidet und die Großbuchstaben vor den Kleinbuchstaben einsortiert werden.

Versteckte Dateien

Wie Sie oben gesehen haben, enthält das Heimatverzeichnis scheinbar nur zwei Dateien bzw. Verzeichnisse. Das dem nicht so ist, zeigt der Einsatz der Option `-a`.

```
ole@defiant:~> ls
Documents  public_html
ole@defiant:~> ls -a
.           .Xresources    .emacs   .profile   .xim               .xtalkrc
..          .bash_history  .exrc    .urlview   .xinitrc           Documents
.Xdefaults  .bashrc        .kermrc  .xcoralrc  .xserverrc.secure  public_html
.Xmodmap    .dvipsrc       .muttrc  .xemacs    .xsession
```

Anstatt zwei werden nun 23 Dateien angezeigt. Diese neu angezeigten Dateien bezeichnet man als *Versteckte Dateien*. Sie werden bei normalen Dateioperationen nicht berücksichtigt. Diese versteckten Dateien finden Sie auch in anderen Betriebssystemen. Dort wird durch ein Attribut der Datei angezeigt, ob eine Datei versteckt ist oder nicht. Wenn Sie sich die Liste der Dateien im obigen Beispiel noch einmal genau anschauen, finden Sie recht schnell die Gemeinsamkeit der versteckten Dateien heraus.

Dateien, deren Name mit einem Punkt beginnt, bezeichnet man als Versteckte Dateien. Sie werden bei normalen Dateioperationen nicht berücksichtigt.

Eine besondere Bedeutung haben die Dateien '`.`' und '`..`'. Sie sind Bestandteil des Verzeichnisses. Der Name '`.`' ist ein weiterer Link für das Verzeichnis in dem sich diese Datei befindet. Dagegen verweist der Name `..` auf das Elternverzeichnis des aktuellen Verzeichnis. Wenn Sie das obige Beispiel als Grundlage nehmen, so ist `.` gleichzusetzen mit `/home/ole` und `..` mit `/home`.

Zwei Optionen können auch miteinander kombiniert werden. Eine ausführliche Liste aller Dateien bekommen Sie durch die Kombination der Optionen `-l` und `-a`.

```
ole@defiant:~> ls -a -l
insgesamt 92
drwxr-xr-x    5 ole    users        4096 Sep 25 12:25 .
drwxr-xr-x    9 root   root         4096 Sep 25 12:25 ..
-rw-r--r--    1 ole    users        5742 Sep 25 12:25 .Xdefaults
-rw-r--r--    1 ole    users        1305 Sep 25 12:25 .Xmodmap
lrwxrwxrwx    1 root   root           10 Sep 25 12:25 .Xresources -> .Xdefaults
-rw-------    1 ole    users           0 Sep 25 12:25 .bash_history
-rw-r--r--    1 ole    users        1691 Sep 25 12:25 .bashrc
-rw-r--r--    1 ole    users         208 Sep 25 12:25 .dvipsrc
-rw-r--r--    1 ole    users        1637 Sep 25 12:25 .emacs
-rw-r--r--    1 ole    users        1124 Sep 25 12:25 .exrc
-rw-r--r--    1 ole    users         164 Sep 25 12:25 .kermrc
-rw-r--r--    1 ole    users        2286 Sep 25 12:25 .muttrc
-rw-r--r--    1 ole    users         952 Sep 25 12:25 .profile
-rw-r--r--    1 ole    users         311 Sep 25 12:25 .urlview
-rw-r--r--    1 ole    users        7913 Sep 25 12:25 .xcoralrc
drwxr-xr-x    2 ole    users        4096 Sep 25 12:25 .xemacs
-rw-r--r--    1 ole    users        3407 Sep 25 12:25 .xim
-rwxr-xr-x    1 ole    users        2324 Sep 25 12:25 .xinitrc
```

```
-rw-r--r--    1 ole    users        1101 Sep 25 12:25 .xserverrc.secure
-rwxr-xr-x    1 ole    users        2804 Sep 25 12:25 .xsession
-rw-r--r--    1 ole    users         119 Sep 25 12:25 .xtalkrc
drwxr-xr-x    2 ole    users        4096 Sep 25 12:25 Documents
drwxr-xr-x    2 ole    users        4096 Sep 25 12:25 public_html
```

Wer es kürzer mag, kann die Buchstaben auch zusammenziehen. Dabei ist es völlig egal, ob Sie `-al` oder `-la` schreiben.

```
ole@defiant:~> ls -al
insgesamt 92
drwxr-xr-x    5 ole    users        4096 Sep 25 12:25 .
drwxr-xr-x    9 root   root         4096 Sep 25 12:25 ..
-rw-r--r--    1 ole    users        5742 Sep 25 12:25 .Xdefaults
-rw-r--r--    1 ole    users        1305 Sep 25 12:25 .Xmodmap
...
```

Parameter und Jokerzeichen

Der `ls` Befehl kann nicht nur den Inhalt des aktuellen Verzeichnis anzeigen. Wenn Sie z. B. den Inhalt des Verzeichnis `Documents` sehen wollen, geben Sie folgendes an.

```
ole@defiant:~> ls -al Documents
insgesamt 12
drwxr-xr-x    2 ole    users        4096 Sep 25 12:25 .
drwxr-xr-x    5 ole    users        4096 Sep 25 12:25 ..
-rw-r--r--    1 ole    users        1106 Sep 25 12:25 .directory
```

Der Name des Verzeichnis `Documents` bezeichnet man als Parameter des Befehls `ls`. Die Angabe des Verzeichnisnamens erfolgte hier relativ zum Arbeitsverzeichnis. Um das Elternverzeichnis des Arbeitsverzeichnis anzuzeigen können wir den Dateinamen `..` verwenden.

```
ole@defiant:~> ls ..
conny  lost+found  ole  perl  vnc  walter  willi
```

Um zur Wurzel zu gelangen müssen Sie sogar noch ein Verzeichnis weiter im Verzeichnisbaum hochgehen.

```
ole@defiant:~> ls ../..
bin   cdrom  dvd  floppy  lib         media  opt     proc  sbin  usr  windows
boot  dev    etc  home    lost+found  mnt    pcmcia  root  tmp   var
```

Sie können die Verzeichnisse auch direkt adressieren, wenn Sie den Verzeichnisnamen mit einem Slash `/` beginnen lassen. Jetzt wird nämlich nicht vom aktuellen Arbeitsverzeichnis aus gesehen, sondern von der Dateiwurzel.

```
ole@defiant:~> ls /
bin   cdrom  dvd  floppy  lib         media  opt     proc  sbin  usr  windows
boot  dev    etc  home    lost+found  mnt    pcmcia  root  tmp   var
ole@defiant:~> ls /home
conny  lost+found  ole  perl  vnc  walter  willi
ole@defiant:~> ls /home/ole
Documents  public_html
```

Neben den Verzeichnissen können Sie auch eine Auswahl unter den Dateien treffen, die Sie sich anzeigen lassen wollen. Hilfreich sind dabei die sogenannten *Jokerzeichen* oder *Wildcards*.

Dazu schauen Sie sich doch mal das Verzeichnis `/etc` an, in dem sich die Konfigurationsdateien des Linux-Systems befinden.

```
ole@defiant:~> ls /etc
DIR_COLORS          inittab             php.ini
HOSTNAME            inputrc             pnm2ppa.conf
Muttrc              insserv.conf        powerd.conf
SuSE-release        ioctl.save          ppp
...
```

Es werden sehr viele Dateien angezeigt. Sie können die Anzahl der angezeigten Dateien verringern, wenn Sie andere Kriterien für die Auswahl stellen. So könnten Sie sich nur die Dateien anzeigen lassen, die mit dem Buchstaben "a" beginnen.

```
ole@defiant:~> ls /etc/a*
/etc/a2ps-site.cfg  /etc/aliases       /etc/at.deny
/etc/a2ps.cfg       /etc/aliases.db    /etc/auto.master
/etc/adjtime        /etc/asound.state  /etc/auto.misc

/etc/alsa.d:
emu10k1  sbawe
```

Der Asterisk "*" steht für eine beliebige Anzahl beliebiger Zeichen und das Fragezeichen "?" steht für *ein* beliebiges Zeichen. Auffällig an dem Beispiel oben ist, daß nicht nur der Name des Verzeichnis angegeben wird, sondern auch der Inhalt. Dies ist eine der Eigenschaften von `ls`. Ist ein Verzeichnis ein Ziel, dann wird der Inhalt des Verzeichnis angezeigt. Mit der Option `-d` kann dies unterbunden werden.

```
ole@defiant:~> ls -d /etc/a*
/etc/a2ps-site.cfg  /etc/aliases     /etc/asound.state  /etc/auto.misc
/etc/a2ps.cfg       /etc/aliases.db  /etc/at.deny
/etc/adjtime        /etc/alsa.d      /etc/auto.master
ole@defiant:~> ls -ld
drwxr-xr-x    5 ole    users        4096 Sep 25 12:25 .
```

Dies wirkt, wie Sie oben sehen, auch beim aktuellen Arbeitsverzeichnis. Es wird nur die Information über das aktuelle Verzeichnis angezeigt und nicht über die enthaltenen Dateien.

Inodes

Wie schon oben erwähnt, kann eine Datei mehrere Namen (Links) besitzen. Dies wird dadurch ermöglicht, daß die Benennung der Daten im System über eine sogenannte Inode-Nummer erfolgt. Als Inode bezeichnet man den Platz, an dem die Informationen über die Datei gespeichert werden. Das Betriebssystem merkt sich also nicht den Namen der Datei sondern die Inode-Nummer. Die Namen sind nur für uns "dumme" Benutzer, die sich mehr unter dem Namen "amaretti.rezept.txt" vorstellen können, als unter einer Inode-Nummer "45312". Die Verzeichnisse sind im Prinzip nichts anderes als Tabellen, die den Zusammenhang zwischen Dateinamen und der Inode-Nummer herstellen.

Die Inode-Nummer können sie sich mit `ls` unter Verwendung der Option `-i` anzeigen lassen.

```
ole@defiant:~> ls -i
   1095 Documents     1097 public_html
ole@defiant:~> ls -i /home
   1074 conny         31873 ole    112382 vnc      286867 willi
     11 lost+found   223209 perl   207194 walter
```

Hilfe

Es gibt noch viel mehr Optionen für `ls`, als die ich Ihnen gerade vorgestellt habe. Um eine kurze Übersicht über die Funktion und vorhandenen Optionen eines Befehls zu erhalten, können Sie den Shellbefehl mit der Option `--help` aufrufen. Mit dem doppelten Minuszeichen wird dem Befehl signalisiert, daß es sich nicht um die einzelnen Optionen `-h`, `-e`, `-l` und `-p` handelt, sondern um eine Option mit dem langen Namen `help`.

```
ole@defiant:~> ls --help
Benutzung: ls [OPTION]... [DATEI]...
Auflistung von Informationen der DATEIen (Standardvorgabe ist das momentane
Verzeichnis). Alphabetisches Sortieren der Einträge, falls weder -cftuSUX
noch --sort angegeben.

  -a, --all                  Einträge, die mit . beginnen, nicht verstecken.
  -A, --almost-all           Keine Anzeige implizierter . und ..
  -b, --escape               Ausgabe octaler Repräsentation für nicht-druck-
...
```

Natürlich ist der Hilfetext etwas länger als die Konsolenhöhe. Auf der Konsole können Sie ohne Probleme scrollen (wenn es nicht mehr als 1000 Zeilen sind). Damit die Anzeige nicht durchscrollt, sondern seitenweise anzeigt, können Sie einen sogenannten Pager benutzen. Ein solcher Pager ist der Befehl `more`. Sie leiten einfach die Ausgabe des Befehls mit dem Zeichen `|` an den Pager weiter.

```
ole@defiant:~> ls --help | more
ole@defiant:~> ls /etc | more
ole@defiant:~> ls -l / | more
ole@defiant:~> ls /dev | more
```

Mit der Leertaste blättern Sie die Seiten des Pagers `more` durch. Erreicht der Pager das Ende des Textes bzw. der Ausgabe, so beendet er sich automatisch.

Zusammenfassung

Der Befehl `ls` zeigt die in den Verzeichnissen enthaltenen Dateien an. Mit Optionen kann man die Art und Weise beeinflussen, wie Befehle funktionieren. Die Parameter geben an, womit gearbeitet werden soll.

Die Optionen `-aldi` sollten sie sich für den Befehl `ls` merken.

`-a`	Zeigt auch die versteckten Dateien an (`-all`)
`-l`	Zeigt ausführliche Liste zu den Dateien an (*long*)
`-d`	Zeigt Verzeichnis anstatt Inhalt an (`-directory`)
`-i`	Zeigt Inodes an (`-inode`)
`--help`	Zeigt die Hilfe für den Befehl an

3.2.2 Erstellen und Löschen von Verzeichnissen und Dateien

Ein Verzeichnissystem macht nur Sinn, wenn es Dateien und Verzeichnisse enthält. Fürs Erstellen und Löschen von Dateien und Verzeichnissen stehen auf der Shell viele Programme zur Verfügung.

Erstellen

Eine einfache Art und Weise eine leere Datei zu erstellen ist der Befehl `touch`.

```
ole@defiant:~> touch testdatei
ole@defiant:~> ls -l
insgesamt 8
drwxr-xr-x    2 ole    users        4096 Sep 25 12:25 Documents
drwxr-xr-x    2 ole    users        4096 Sep 25 12:25 public_html
-rw-r--r--    1 ole    users           0 Sep 26 14:37 testdatei
```

Eigentlich soll der Befehl `touch` die Zeit der letzten Änderung und des letzten Zugriffs auf die aktuelle Zeit stellen. Existiert die angegebene Datei nicht, dann wird einfach eine neue Datei erstellt.

Ein neues Verzeichnis zu erstellen ist ebenso einfach. Der Befehl heißt hier `mkdir`.

```
ole@defiant:~> mkdir test
ole@defiant:~> ls
Documents  public_html  test  testdatei
ole@defiant:~> mkdir beruf hobby
ole@defiant:~> ls
Documents  beruf  hobby  public_html  test  testdatei
```

Sie müssen sich beim Befehl `mkdir`, wie auch bei `touch`, nicht auf einen Parameter beschränken. Sie können ruhig mehrere Dateien bzw. Verzeichnisse angegeben, die erstellt werden sollen.

Auch eine Angabe des Verzeichnis mit seinen Elternverzeichnissen ist möglich. Allerdings müssen diese vorher existieren.

```
ole@defiant:~> cd /
ole@defiant:/> mkdir /home/ole/test/neuerOrdner
ole@defiant:/> ls /home/ole/test
neuerOrdner
```

Löschen

Was man erstellt, muß man auch löschen können. Dies gilt für Verzeichnisse wie auch für Dateien. Der Befehl `rmdir` (*remove directory*) sorgt für das Löschen von Verzeichnissen.

```
ole@defiant:~> rmdir beruf
ole@defiant:~> rmdir test
rmdir: »test«: Das Verzeichnis ist nicht leer
```

Allerdings löscht `rmdir` nur leere Verzeichnisse. Sobald sich nur eine Datei oder ein Verzeichnis darin befindet, wird das Löschen des Verzeichnis verweigert.

Dateien entledigt man sich dagegen durch den Befehl `rm` (*remove*).

```
ole@defiant:~> rm testdatei
ole@defiant:~> rm test
rm: »test« ist ein Verzeichnis
```

Mit Verzeichnissen tut sich `rm` dagegen schwer und verweigert das Löschen mit einem entsprechenden Hinweis.

Dies ist auch die normale Arbeitsweise der Shellkommandos. Klappt ein Befehl, so erscheint in der Regel kein Kommentar, sondern der Prompt fordert den Benutzer auf wieder tätig zu werden. Nur wenn Fehler aufgetreten sind, melden sich die Kommandos noch einmal beim Benutzer.

Sie können aber `rm` überreden auch Verzeichnisse zu löschen. Mit der Option `-r` (*recursive*) löscht `rm` Verzeichnisse *mit* ihrem Inhalt.

```
ole@defiant:~> rm -r test
```

Manchmal fragt `rm` auch nach, ob eine Datei gelöscht werden soll. Ist einem das zu lästig, gerade wenn ganze Verzeichnisstrukturen gelöscht werden sollen, dann sorgt der Schalter `-f` für Ruhe. Er sorgt dafür, daß die Dateien ohne Nachfrage gelöscht werden.

Sie sollten aber mit dem Befehl vorsichtig sein. Stellen Sie sich doch einfach mal vor was passiert, wenn der Administrator folgendes eingibt:

```
defiant:~$ rm -rf /
```

Der Editor `vi`

Der Editor, den Sie auf jedem Linux/Unix-System vorfinden ist der `vi` oder einer seiner Klone. Seine Benutzerführung ist etwas eigenwillig und gewöhnungsbedürftig. Da er aber auf jedem System vorhanden ist, sollten Sie sich doch mit ihm etwas auskennen.

Eine Datei zum Bearbeiten rufen Sie ganz einfach auf, indem Sie den Namen hinter dem Befehl `vi` schreiben.

```
ole@defiant:~> vi meinedatei
```

Existiert die Datei nicht, so geht der `vi` von einer leeren neuen Datei aus, wie sie auch in der Statuszeile ganz unten sehen können.

```
"meinedatei" [Neue Datei]                                  0,0-1        Alles
```

Der `vi` besitzt verschieden Modi für die Bearbeitung von Dokumenten. Beim Start landen Sie im Bearbeitungsmodus, der spezielle Funktionen zur Bearbeitung der Dokumente zur Verfügung stellt. Diese Funktionen werden nicht wie von den meisten Editoren gewohnt über ein Menü zur Verfügung gestellt, sondern durch Tastaturkürzel.

Sie müssen, wenn Sie Text schreiben wollen, in den Eingabemodus wechseln. Dazu drücken Sie im Bearbeitungsmodus die Taste `<I>`. Die Statuszeile zeigt nun `EINFÜGEN` bzw. `INSERT` an.

```
-- EINFÜGEN --                                             0,1          Alles
```

Hier können Sie nun Text eingeben, mit den Richtungstasten sich durch den Text bewegen und mit der Taste `<ENTF>` bzw. `<DEL>` Zeichen rechts vom Cursor löschen.

Um wieder in den Bearbeitungsmodus zu wechseln, drücken Sie einfach die Taste `<ESC>`. Um sicher zu gehen, können Sie die Taste auch mehrfach drücken. `vi` quittiert das dann mit einem Piepton.

Im Bearbeitungsmodus können Sie das Zeichen unter dem Cursor entweder wie im Eingabemodus mit der Taste `<DEL>` löschen oder indem Sie die Taste `<X>` betätigen. Es gibt aber auch weiter reichende Löschkommandos im Zusammenhang mit der Taste `<D>`. So löscht `<D> <W>` die Zeichen bis zum Wortende inklusive des folgenden Leerzeichens. Mit `<D> <E>` löscht er ebenfalls den Rest vom Wort, ohne aber das Leerzeichen mitzunehmen. Mit der Kombination `d$` (`<D> <SHIFT>+<4>`) löscht er den Text vom Cursor bis zum Zeilenende. Und um die ganze Zeile, in der sich der Cursor befindet, zu löschen, betätigen Sie einfach zweimal die Taste `<D>` (`<D> <D>`). Sie können auch durch das Voranstellen einer Zahl vor das Kommando angeben, wie oft das Kommando ausgeführt werden soll. So können Sie mit `<4> <D> <D>` vier Zeilen löschen oder mit `<3> <D> <W>` die drei Worte rechts vom Cursor.

Die Mehrfachausführung klappt auch mit dem Eingabemodus. Einfach vorm Einschalten des Eingabemodus eine Zahl eintippen. Nach Beendigung des Eingabemodus wird das frisch Eingegebene mehrfach ausgegeben. So können Sie z. B. mit der Tastenkombination `<7> <5> <I> <-> <ESC>` 75 Minus-Zeichen in den Text einfügen.

Vom Bearbeitungsmodus können Sie auch einen Kommandomodus aufrufen. Dies erfolgt durch den Doppelpunkt (`<SHIFT>+<:>`). Der Doppelpunkt erscheint in der unteren Statuszeile und Sie können dann weitere Befehle eintippen, die Sie mit der `<RETURN>`-Taste abschließen.

`:q`	Beendet den `vi`
`:q!`	Beendet den `vi` ohne Nachzufragen
`:wq`	Beendet den `vi` und speichert das Dokument vorher

Bei Linux wird der `vi`-Klon `vim` mitgeliefert und durch den Befehl `vi` ausgeführt. Mit dem Befehl `vimtutor` erhalten Sie ein kleines Tutorial mit Übungen zur Bedienung des `vim`.

3.2.3 Kopieren, Verschieben und Umbenennen von Dateien

Für das Kopieren, Verschieben und Umbenennen von Dateien brauchen wir hauptsächlich zwei Befehle: `cp` (*copy*) und `mv` (`move`).

Kopieren

Um eine Kopie einer Datei zu erstellen benutzen Sie den Befehl `cp`. Der Befehl braucht eine Datei, die es zu kopieren gilt, (Quelldatei) und den neuen Namen der Datei (Zieldatei).

```
ole@defiant:~/test> vi themenvorschlag.txt
ole@defiant:~/test> ls -l
insgesamt 4
-rw-r--r--    1 ole      users        2123 Okt  9 10:25 themenvorschlag.txt
ole@defiant:~/test> cp themenvorschlag.txt themenvorschlag.txt.alt
ole@defiant:~/test> ls -l
insgesamt 8
-rw-r--r--    1 ole      users        2123 Okt  9 10:25 themenvorschlag.txt
-rw-r--r--    1 ole      users        2123 Okt  9 10:26 themenvorschlag.txt.alt
ole@defiant:~/test>
```

So kann einfach eine Sicherungskopie einer Datei erstellt werden. In den meisten Fällen werden aber eine oder mehrere Dateien in ein anderes Verzeichnis kopiert, wie z. B. in `/floppy` für die Diskette. In diesem Fall wird anstatt der Zieldatei ein Zielverzeichnis angegeben. Erkennt der Befehl `cp`, daß das Ziel ein existierendes Verzeichnis ist, dann kopiert er die Datei ohne den Namen zu ändern in das Verzeichnis. Existiert kein Verzeichnis, so wird die Angabe als Dateiname verstanden und die Datei dementsprechend kopiert. Übrigens: `cp` überschreibt in der normalen Einstellung existierende Dateien gnadenlos.

```
ole@defiant:~/test> mkdir backup
ole@defiant:~/test> ls -l backup/
insgesamt 0
ole@defiant:~/test> cp themenvorschlag.txt backup
ole@defiant:~/test> ls -l backup/
insgesamt 4
-rw-r--r--    1 ole      users        2123 Okt  9 10:40 themenvorschlag.txt
```

Sie können auch mehrere Dateien zum Kopieren angegeben, wenn Ihr Ziel ein Verzeichnis ist.

```
ole@defiant:~/test> cp /etc/enscript.cfg /etc/esd.conf backup
ole@defiant:~/test> ls -l backup
insgesamt 16
-rw-r--r--    1 ole      users        5983 Okt  9 10:43 enscript.cfg
-rw-r--r--    1 ole      users          77 Okt  9 10:43 esd.conf
-rw-r--r--    1 ole      users        2123 Okt  9 10:40 themenvorschlag.txt
```

Wenn Sie das aktuelle Verzeichnis als Ziel angeben wollen, machen Sie das einfach durch den Punkt '`.`'. Genauso können Sie das Elternverzeichnis mit '`..`' als Ziel angeben. Um zu sehen, welche Dateien kopiert worden sind, benutzen Sie den Befehl `cp` einfach mit dem Schalter `-v` (*verbose*). Diese auch als "Blubberschalter" bekannte Option veranlaßt den Befehl dazu ausführlich über seine Tätigkeit zu berichten. Sie werden bei vielen Shell-Befehlen die Option `-v` wiederfinden. Sie können natürlich auch für die Liste der zu kopierenden Dateien die Jokerzeichen wie den Asterisk "`*`" verwenden.

```
ole@defiant:~/test> cp -v /etc/f* .
»/etc/fam.conf« -> »./fam.conf«
»/etc/fb.modes« -> »./fb.modes«
»/etc/fdprm« -> »./fdprm«
»/etc/filesystems« -> »./filesystems«
»/etc/fstab« -> »./fstab«
»/etc/ftpusers« -> »./ftpusers«
```

Verschieben und Umbenennen

Der Befehl `mv` arbeitet im Prinzip genau wie der Befehl `cp`. Allerdings existiert die Quelldatei bzw. Quelldateien danach nicht mehr. Im Prinzip ist ein Verschieben und Umbenennen nichts anderes als eine Änderung des Verzeichniseintrags. Nur wenn zwischen zwei Partitionen verschoben werden soll, muß die Datei erst kopiert und dann die alte Datei gelöscht werden. Dadurch ist Verschieben im Normalfall schneller als Kopieren.

Um also eine Datei umzubennen benutze ich den Befehl `mv` unter Angabe von altem und neuem Namen.

```
ole@defiant:~/test> ls -l
insgesamt 12
drwxr-xr-x    2 ole      users        4096 Okt  9 10:43 backup
-rw-r--r--    1 ole      users        2123 Okt  9 10:25 themenvorschlag.txt
-rw-r--r--    1 ole      users        2123 Okt  9 10:26 themenvorschlag.txt.alt
ole@defiant:~/test> mv themenvorschlag.txt.alt themenvorschlag.old
ole@defiant:~/test> ls -l
insgesamt 12
drwxr-xr-x    2 ole      users        4096 Okt  9 10:43 backup
-rw-r--r--    1 ole      users        2123 Okt  9 10:26 themenvorschlag.old
-rw-r--r--    1 ole      users        2123 Okt  9 10:25 themenvorschlag.txt
```

Um die Syntax eines Shell-Kommandos kurz und knapp darzustellen, werden Syntaxregeln mit einer bestimmten Symbolik verwendet.

Der `mv` Befehl kann als Syntaxregel so ausgedrückt werden:

```
mv ALTERNAME NEUERNAME
```

Die groß geschriebenen Worte stehen für den Wert, der dort eingetragen werden muß.

Ähnlich wie bei `cp` führt die Angabe eines existierenden Verzeichnis als Ziel dazu, daß die Datei nicht umbenannt, sondern in das Verzeichnis unter Beibehaltung des Namens verschoben wird.

```
ole@defiant:~/test> mv themenvorschlag.old backup
ole@defiant:~/test> ls -l
insgesamt 8
drwxr-xr-x    2 ole      users        4096 Okt  9 11:09 backup
-rw-r--r--    1 ole      users        2123 Okt  9 10:25 themenvorschlag.txt
oole@defiant:~/test> ls -l backup
insgesamt 8
-rw-r--r--    1 ole      users        2123 Okt  9 10:26 themenvorschlag.old
-rw-r--r--    1 ole      users        2123 Okt  9 10:40 themenvorschlag.txt
```

Wenn Sie ein Verzeichnis als Ziel angeben, können Sie nicht nur eine Datei verschieben, sondern eine ganze Liste von Dateien.

```
ole@defiant:~/test> touch datei1.txt datei2.txt datei3.txt
ole@defiant:~/test> ls -l
insgesamt 8
drwxr-xr-x    2 ole      users        4096 Okt  9 11:10 backup
-rw-r--r--    1 ole      users           0 Okt  9 11:12 datei1.txt
-rw-r--r--    1 ole      users           0 Okt  9 11:12 datei2.txt
-rw-r--r--    1 ole      users           0 Okt  9 11:12 datei3.txt
-rw-r--r--    1 ole      users        2123 Okt  9 10:25 themenvorschlag.txt
ole@defiant:~/test> mv d*.txt backup
ole@defiant:~/test> ls -l backup
insgesamt 8
-rw-r--r--    1 ole      users           0 Okt  9 11:12 datei1.txt
```

```
-rw-r--r--    1 ole      users           0 Okt  9 11:12 datei2.txt
-rw-r--r--    1 ole      users           0 Okt  9 11:12 datei3.txt
-rw-r--r--    1 ole      users        2123 Okt  9 10:26 themenvorschlag.old
-rw-r--r--    1 ole      users        2123 Okt  9 10:40 themenvorschlag.txt
```

Die Kommandosyntax kann durch folgende Regeln ausgedrückt werden:

```
mv DATEILISTE ZIELVERZEICHNIS
```

oder auch als

```
mv DATEI1 [DATEI2 [DATEI3 [...]]] ZIELVERZEICHNIS
```

Die eckigen Klammern zeigen an, daß der entsprechende Teil optional ist und nicht unbedingt notwendig.

Auch bei `mv` gibt es den “Blubberschalter” `-v` und noch weitere Optionen. Die Syntaxregel unter Berücksichtigung der Optionen lautet dann:

```
mv [-v] DATEILISTE ZIELVERZEICHNIS
```

oder auch

```
mv [OPTIONEN] DATEILISTE ZIELVERZEICHNIS
```

3.3 Informationen und Hilfe

Im Gegensatz zu der graphischen Oberfläche kann auf der Shell nicht einfach durch durchgucken der Menüpunkte der richtige Befehl gefunden werden. Linux stellt Informationen und Hilfen zur Verfügung.

3.3.1 Hilfe im Befehl

Fast jeder Befehl und somit auch die Shell, die ja auch nichts anderes als ein Programm ist, stellt eine eigene Kurzhilfe zur Verfügung.

In der Bash gibt es eingebaute Befehle, die nicht als eigenständige Programme vorliegen. Eine Übersicht der Befehle gibt der Befehl `help` . Genauere Informationen zu einem Befehl bekommen Sie dann durch Eingabe des Befehls hinter `help`.

```
ole@defiant:~/test> help pwd
pwd: pwd [-PL]
    Print the current working directory.  With the -P option, pwd prints
    the physical directory, without any symbolic links; the -L option
    makes pwd follow symbolic links.
```

Bei den meisten anderen Befehlen kommen Sie mit dem Schalter `--help` weiter. Diese Option veranlaßt den Befehl dazu, eine kurze Erklärung seiner Funktion und eine Übersicht über seine Optionen zu geben.

```
ole@defiant:~/test> touch --help
Benutzung: touch [OPTION]... DATEI...
   oder:   touch [-acm] MMDDhhmm[YY] DATEI... (veraltet)
Aktualisieren der Zugriffs- und Modifikationszeiten jeder DATEI auf die
momentane Zeit.

  -a                        Nur die Zugriffszeit ändern.
  -c, --no-create           Keine Dateien erzeugen.
```

```
  -d, --date=ZEICHENKETTE   Lesen der ZEICHENKETTE und statt der momentanen
                              Zeit verwenden.
  -f                        (ignoriert)
  -m                        Nur Modifikationszeit ändern.
  -r, --reference=FILE      Die Zeiten dieser Datei anstatt der momentanen Zeit
                              verwenden.
  -t MARKE                  verwenden von [[HH]JJ]MMTTSSmm[.ss] statt der
                              momentanen Zeit.
      --time=WORT           Setzen der Zeit die von WORT angegeben wird:
                              access, atime, (wie -a), mtime, modify (wie -m).
      --help                Anzeigen dieser Hilfe und beenden.
      --version             Ausgabe der Versionsinformation und beenden.

Beachten Sie das die drei Zeitformate die von den Optionen -d, -t und dem
veraltete Argument erkannt werden alle verschieden sind.

Berichten Sie Fehler an <bug-fileutils@gnu.org>.
```

Andere Befehle, wie z. B. der Packer `zip` liefern diese Erklärungen schon, wenn Sie ohne Parameter aufgerufen werden.

```
ole@defiant:~/test> zip
Copyright (C) 1990-1999 Info-ZIP
Type 'zip "-L"' for software license.
Zip 2.3 (November 29th 1999). Usage:
zip [-options] [-b path] [-t mmddyyyy] [-n suffixes] [zipfile list] [-xi list]
  The default action is to add or replace zipfile entries from list, which
  can include the special name - to compress standard input.
  If zipfile and list are omitted, zip compresses stdin to stdout.
  -f   freshen: only changed files  -u   update: only changed or new files
  -d   delete entries in zipfile    -m   move into zipfile (delete files)
  -r   recurse into directories     -j   junk (don't record) directory names
  -0   store only                   -l   convert LF to CR LF (-ll CR LF to LF)
  -1   compress faster              -9   compress better
  -q   quiet operation              -v   verbose operation/print version info
  -c   add one-line comments        -z   add zipfile comment
  -@   read names from stdin        -o   make zipfile as old as latest entry
  -x   exclude the following names  -i   include only the following names
  -F   fix zipfile (-FF try harder) -D   do not add directory entries
  -A   adjust self-extracting exe   -J   junk zipfile prefix (unzipsfx)
  -T   test zipfile integrity       -X   eXclude eXtra file attributes
  -y   store symbolic links as the link instead of the referenced file
  -R   PKZIP recursion (see manual)
  -e   encrypt                      -n   don't compress these suffixes
```

3.3.2 Manual-Pages

Für weitergehende Erläuterungen wurden die Manual Pages, oder auch nach Ihrem Verwaltungsprogramm Man-Pages genannt, entwickelt. Die Man-Pages sind praktisch die Bedienungsanleitungen für die Shell-Befehle.

Nach einem `man cp` wird das Dokument entpackt und dann mit dem passenden Pager angezeigt.

Mit den Richtungstasten können Sie nun hin- und herscrollen. Die Betätigung der Taste `<Q>` beendet dann das Man-Programm.

Die Programme `whatis` und `apropos` basieren auf `man`. `whatis` liefert eine Kurzbeschreibung für einen Befehl, während `apropos` diese Kurzbeschreibungen nach einem Begriff durchsucht.

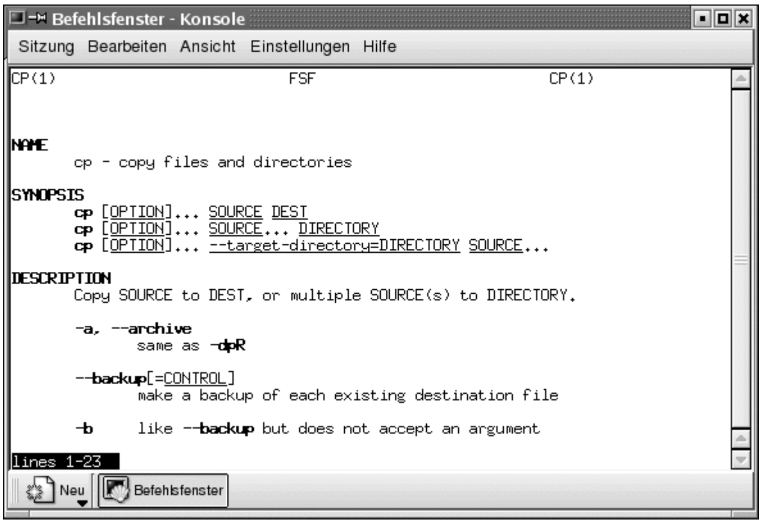

Abbildung 3.1: Ausgabe des Befehls `man cp`

```
ole@defiant:~/test> apropos login
logname (1)          - print user's login name
/etc/login.defs (5) [login.defs] - Login configuration
login_tty (3)        - tty utility functions
getlogin (3)         - get user name
sulogin (8)          - Single-user login
slogin (1)           - OpenSSH SSH client (remote login program)
nologin (8)          - politely refuse a login
faillog (8)          - examine faillog and set login failure limits
nologin (5)          - prevent non-root users from logging into the system
ssh (1)              - OpenSSH SSH client (remote login program)
utmp (5)             - login records
wtmp (5)             - login records
      ....

ole@defiant:~/test> whatis ssh
ssh (1)              - OpenSSH SSH client (remote login program)
```

Die Manual-Pages enthalten nicht nur Informationen zu Befehlen, sondern auch zu Konfigurationsdateien, Kernelfunktionen u.a.

Notizen:

Notizen:

III | **Arbeitsblatt Einführung in die Shell** | 3.1

Sollte eine Aufgabe zu einer Fehlermeldung führen, kann das von mir gewollt sein! Prüfen Sie aber dennoch, ob Sie keinen Tippfehler gemacht haben, und ob die Voraussetzungen wie in der Aufgabenstellung gegeben sind. Auch sollten keine Verzeichniswechsel ausgeführt werden, wenn dies nicht ausdrücklich in der Aufgabe verlangt wird! Notieren Sie die Ergebnisse auf einem seperaten Zettel.

1 Wechseln Sie vom X-Window-System auf die Konsole 2.

2 Loggen Sie sich dort als *walter* ein.

3 In welchem Verzeichnis befinden Sie sich?

4 Auf welchem Terminal arbeiten Sie gerade?

5 Führen Sie den Befehl `rm -rf *` aus. Was bewirkt der Befehl?

6 Legen Sie eine leere gewöhnliche Datei mit dem Namen `meiohmei` an.

7 Legen Sie drei leere gewöhnlichen Dateien mit den Namen `achdugrueneneune`, `achduliebermeinvater` und `achdukannstmichmal` mit einem Befehl an.

8 Lassen Sie sich nun den Verzeichnisinhalt anzeigen.

9 Lassen sich sich ausführlichere Informationen über den Verzeichnisinhalt anzeigen.

10 Lassen Sie sich nun für die vier Dateien die Inode-Nummern anzeigen.

11 Legen Sie mit dem `vi` die Datei `wasichschonimmersagenwollte` mit dem folgenden Inhalt an.

Linux Unser, Der Du bist im Kernel,
Geöffnet sei Dein Swap-File,
Dein KDE komme,
Dein Bash-Command geschehe,
Wie in Red Hat so auch in Suse.
Unser täglich Login gib uns heute,
Und vergib uns unsere zu kleine Festplatte,
Wie auch wir vergeben Dir Deiner Installation.
Und führe uns nicht in die Kernel-Panik,
Sondern erlöse uns von Microsoft.
Denn Dein ist das Netz Und die Festplatte Und die Stabilität
In Ewigkeit
Enter!
Anonymous

12 Legen Sie von der Datei `wasichschonimmersagenwollte` eine Kopie namens `ichhabsgesagt` an.

13 Lassen Sie sich die Inode Nummern der Dateien anzeigen.

14 Benennen Sie nun `ichhabsgesagt` nach `ichsagte` um.

15 Lassen Sie sich die Inode Nummern der Dateien wiederum anzeigen und vergleichen Sie das Ergebnis mit Aufgabe 13. Was sehen sie?

16 Legen Sie das Verzeichnis `abfall` an.

17 Lassen Sie sich den Inhalt des aktuellen Verzeichnis ausführlich anzeigen. Wie unterscheidet sich das Verzeichnis von den gewöhnlichen Dateien?

18 Lassen Sie sich ausführlich den Inhalt des Verzeichnis `abfall` anzeigen.

III	Arbeitsblatt Einführung in die Shell	3.2

19 Kopieren Sie die Datei `ichsagte` in das Verzeichnis `abfall`.

20 Kopieren Sie die Dateien, die mit `achdu` beginnen in das Verzeichnis `abfall`. Lassen Sie sich dabei die Aktionen des Programms anzeigen.

21 Lassen Sie sich wieder ausführlich den Inhalt des Verzeichnis `abfall` anzeigen.

22 Kopieren Sie alle Dateien aus dem Verzeichnis `etc`, die mit `c` beginnen, in das Verzeichnis `abfall`. Welche Fehler treten dabei auf?

23 Versuchen Sie über die Hilfe herauszubekommen, ob mit dem `cp` es prinzipiell möglich ist Verzeichnisse zu kopieren.

24 Wechseln Sie in das Verzeichnis `abfall` und lassen Sie sich das aktuelle Verzeichnis anzeigen.

25 Wechseln Sie zur Verzeichniswurzel und lassen Sie sich den Inhalt des Verzeichnis ausführlich anzeigen. Um was für Dateien handelt es sich hier?

26 Wechseln Sie in das Verzeichnis `root`. Was passiert?

27 In welchem Verzeichnis sind Sie nun?

28 Wechseln Sie in das Verzeichnis `etc`.

29 Wann wurde das Verzeichnis `cron.daily` erstellt?

30 Was bewirkt der Schalter `-G` beim Befehl `ls`

31 Testen Sie den Schalter `-G` bei der ausführlichen Anzeige des Inhalts des Verzeichnis `cron.daily`.

32 Öffen Sie die Datei `do_mandb` aus `cron.daily` im `vi`. Welchen Hinweis bekommen Sie.

33 Beenden Sie den `vi` ohne zu speichern.

34 Wechseln Sie wieder in das Heimatverzeichnis von `walter`.

35 Öffnen Sie die Datei `wasichschonimmersagenwollte` mit dem `vi`.

36 Löschen Sie das Wort KDE.

37 Schreiben Sie an der Stelle jetzt `Gnome` hinein.

38 Gehen Sie an den Anfang des Dokuments und fügen Sie dort eine Leerzeile ein.

39 Fügen Sie nun in dieser Zeile 75 mal den Asterisk ein.

40 Löschen Sie die Zeile "Dein Bash-Command geschehe,"

41 Verlassen Sie den `vi` ohne das Dokument zu speichern.

42 Wenn Sie noch Zeit haben, dann starten Sie das Lernprogramm für den `vi` und arbeiten es durch, soweit Sie kommen.

Kapitel 4

Die Shell I

4.1 Was ist eine Shell?

Die Shell ist nur ein Linux-Programm. Sie dient als Verbindungsstelle zwischen dem Benutzer und dem Betriebssystem. Ihre Aufgabe ist es, die eingegebene Kommandozeile so umzuformen, daß das Betriebssystem sie interpretieren kann. Daher gibt es auch nicht nur eine Shell, sondern eine Anzahl von Programmen, die diese Vermittlungsaufgabe übernehmen können. Eine Übersicht über die verschiedenen Shells liefert Tabelle 4.1.

Im Normalfall, wenn man sich auf einem Linux-Rechner einloggt, benutzt man die `bash`. Die Standard-Shell eines Benutzers wird durch einen Eintrag in der `/etc/passwd` festgelegt. Der Name der Login-Shell wird in der Umgebungsvariablen `SHELL` gespeichert. Die Variable enthält nicht zwingend, was viele glauben, den Namen der aktuellen Shell.

Um eine andere Shell zu starten, geben sie einfach das Shellkommando mit seinem Pfad ein. Dies startet eine Kindprozess in dem die neue Shell läuft. Um die Shell zu beenden geben Sie `exit` ein.

Im folgenden Beispiel wird durch die Eingabe `ksh` zur *Public Domain Korn Shell* [1] gewechselt. Dies können Sie an der veränderten Prompt-Einstellung sehen, da die Korn-Shell die Abkürzung ~ fürs Heimatverzeichnis nicht kennt. Trotzdem zeigt die Variable `SHELL` immer noch die Bash als Shell an.

```
ole@enterprise:~> echo $SHELL
/bin/bash
ole@enterprise:~> ksh
ole@enterprise:/home/ole> echo $SHELL
/bin/bash
ole@enterprise:/home/ole> exit
ole@enterprise:~>
```

Es gibt drei verschiedene Situationen, in denen eine Shell arbeiten kann. Als *interaktive Login-Shell*, als *interaktive Shell* und als *nichtinteraktive Shell*.

Eine interaktive Login-Shell, meistens nur Login-Shell genannt, wird direkt nach dem Einloggen gestartet und Sie können direkt mit ihr Arbeiten.

Eine interaktive Shell ermöglicht ebenfalls das direkte Arbeiten. Allerdings wurde sie nicht durch einen Login-Vorgang gestartet. Dies kann z. B. durch die Eingabe des Shellprogramms in einer anderen Shell geschehen, wie Sie im obigen Beispiel sehen konnten, oder z. B. durch das Starten einer Terminalemulation auf der graphischen Oberfläche.

Mit einer nichtinteraktiven Shell können Sie nicht am Prompt arbeiten. Sie wird nur zum Ausführen eines Shell-Skripts gestartet und beendet sich nach dem Ende des Skripts automatisch. Damit dient diese Shell nur als eigenständige Umgebung für die Ausführung von vordefinierten Shell-Befehlen ohne Möglichkeiten für den Benutzer direkt einzugreifen.

[1] Das Paket *pdksh* muß eventuell nachinstalliert werden.

Shell	Built-In Befehle	Kommandozeilenoptionen
ash	24	10
bash (Bourne Again Shell)	48	12
ksh (Public Domain Korn Shell)	42	20
rcsh (Emulation der csh)	53	18
zsh	84	50

Tabelle 4.1: Übersicht über die häufigsten Shells

4.1.1 chsh

Das Kommando **chsh** ändert dauerhaft die Login-Shell.

`chsh [-s LOGINSHELL] [USER]`

Der Benutzer kann nur die Shell für sich selbst ändern, während **root** das für jeden Benutzer machen kann. Dabei ist aber der Benutzer auf die Shells eingeschränkt, die in der Datei `/etc/shells` aufgelistet werden. Wird die Option `-s` nicht angegeben, so erfolgt die Änderung der Daten im interaktiven Modus. Vor jeder Änderung wird das Kennwort des Benutzers zur Sicherheit abgefragt.

Optionen

`-s LOGINSHELL`	Die neue Login-Shell

Beispiel
Der Benutzer will in Zukunft die Korn-Shell verwenden. Er ändert dies mit
`chsh -s /usr/bin/ksh`

4.2 Die Bash

Die erste programmierbare Shell wurde von Steve Bourne entwickelt und war die erste UNIX-Shell. Als aufwärtskompatibler Nachfolger wurde später die Bourne-Again-Shell oder kurz Bash entwickelt. Sie wurde als Bestandteil des GNU-Betriebssystems entworfen, für das allerdings bisher nur der Compiler, Tools wie z.B. Editoren und die Shell existieren. Der wichtigste Teil des Betriebssystems, der Betriebssystemkern, fehlt allerdings noch. Da alle Teile freie Software sind, sind sie Bestandteile der freien UNIX-Varianten Linux und Free-BSD, die im Prinzip hauptsächlich aus dem Betriebssystemkern mit Treibern bestehen und die anderen Teile, die für ein funktionierendes System notwendig sind, wie eben Compiler, Shell, Editoren, Tools zum Dateihandling oder ähnliches, aus anderen Quellen beziehen. So ist die Bash die Standard-Shell von Linux, wobei auch hier dem Benutzer die Wahlfreiheit gelassen wird, und man jede andere UNIX-Shell verwenden kann.

Im Prinzip ist die Bash nichts weiter als ein normales Programm, was die Kommandos des Benutzers an das Betriebssystem und andere Programme weiterleitet und deren Ausgaben wieder dem Benutzer zur Verfügung stellt.

Als allerersten sollten Sie bei der Bash beachten, daß zwischen Groß- und Kleinschreibung peinlich genau unterschieden wird. So sind mit den Namen `ahrschlecker.txt`, `AHRSCHLECKER.TXT` und `AhrSchlecker.TXT` im Gegensatz zu DOS/Windows drei verschiedene Dateien gemeint. Da die meisten Befehle auch nur ausführbare Dateien sind, gilt es natürlich auch für diese.

Der Prompt zeigt nicht nur die Bereitschaft der Shell an ein neues Kommando zu empfangen. Er kann auch mit Informationen ausgestattet werden. Bei SuSE wird voreingestellt der eingeloggte Benutzer, der Rechner und das Verzeichnis angezeigt. Verantwortlich für das Aussehen des Prompts sind die Variablen `PS1` und `PS2`.
`tapico@defiant:/etc/news >`

Unvollständige Befehlszeilen quittiert die Shell (Bourne und Bash) durch Ausgabe des Hilfsprompts, der durch die Variable `PS2` festgelegt wird. In der Regel handelt es sich um ein Größer-Zeichen gefolgt von einem

Leerzeichen. Häufigste Ursache dafür ist ein fehlendes zweites Anführungszeichen. Durch Eingabe dieses Zeichens wird die Befehlszeile vervollständigt und der Befehl kann ausgeführt werden.

```
ole@enterprise:~> echo "Hallo Welt
> "
Hallo Welt
```

Häufig ist es nötig einen Befehl zu wiederholen oder einen Befehl leicht abzuwandeln. Unter Linux steht eine Eingabewiederholung ähnlich dem `DOSKEY` unter Windows zur Verfügung: die History. Diese wird im wesentlichen über die Cursortasten Auf und Ab gesteuert.

Sollten Sie mal mit dem Platz in einer Zeile nicht auskommen, so kann eine Eingabezeile problemlos verlängert werden, indem man anstatt <RETURN> zu drücken unmittelbar vor <RETURN> einen Backslash \ eingibt.

Gerade am Anfang können Sie ungewollt in Programme raten, mit denen Sie sich nicht auskennen. Hier ein paar Tips, wie Sie solche Programme beenden können.

- Die Taste `<Q>` oder `<Q><RETURN>` für "quit" führen oft zum Erfolg. (z. B. `less` (4.5.10), `more` (4.5.9) und `top` (12.6.3))
- Die Kombination `<STRG>+<D>` steht für Ende-der-Datei. Damit können Sie vor allem Programme beenden, die auf eine Eingabe warten. (z. B. `cat` (4.5.2), wenn es von der Standardeingabe liest, und `at` (13.1.1)). Sollten Sie sich aber direkt in der Shell befinden, dann werden Sie damit ausgeloggt.
- Eine etwas härtere Methode ein Programm zu beenden ist die Kombination `<STRG>+<C>`. Hiermit wird das laufende Programm aufgefordert, sich sofort zu beenden.
- Sollten Sie sich im Editor `vi` verfangen haben, kommen sie durch die Tastenfolge `<ESC><:><Q><!><RETURN>` heraus.
- Der Editor `joe` ist da viel leichter zu beenden. Er reagiert auf die Kombination `<STRG>+<C>` als normalen Ende-Befehl.

4.2.1 Kommandosyntax

Ein Kommando ist eine Folge von Zeichenketten, die durch ein oder mehrere Leerzeichen oder Tabulatoren getrennt werden. Die ersten Zeichenkette legt den Namen des Kommandos fest. Die weiteren Zeichenketten werden als Parameter bezeichnet.

```
KOMMANDO [PARAMETER1] [PARAMETER2] ... [PARAMETERn]
```

Dabei gehören die Leerzeichen und Tabulatoren nicht zu den Parametern, sondern werden nur als Trennzeichen gewertet. Soll der Parameter hingegen Leerzeichen oder Tabulatoren enthalten, so muß der Ausdruck in einfache (') oder doppelte (") Anführungszeichen gesetzt werden.

Beispiel

```
echo Ich          bin          da!
echo "Ich          bin          da!"
```

Bei den Parametern kann man noch zwischen normalen Parametern und Optionen unterscheiden. Optionen werden durch ein `-` in der Kurzform[2] oder `--` in der langen Form eingeleitet.

Merkregel

Das Kommando legt fest **was** gemacht werden soll. Die Parameter bestimmen **womit** und die Optionen **wie**.

[2] Nur ein Buchstabe lang.

4.3 Arbeiten mit Verzeichnissen

Nach dem Einloggen befinden Sie sich immer in Ihrem Heimatverzeichnis[3]. Dies wird generell abgekürzt durch das Tilde-Zeichen `~`. Die folgenden Arbeitsweisen des Tilde-Zeichens werden als Tilde-Ausdehnung (engl. *tilde expansion*) bezeichnet.

`~`	Heimatverzeichnis des aktuellen Benutzers
`~walter`	Heimatverzeichnis von dem Benutzer *walter*
`~+`	Das aktuelle Arbeitsverzeichnis
`~-`	Das alte Arbeitsverzeichnis

Ihre Funktionsweise können Sie im folgenden Beispiel verfolgen.

```
ole@enterprise:~/test> echo ~
/home/ole
ole@enterprise:~/test> echo ~walter
/home/walter
ole@enterprise:~/test> echo ~+
/home/ole/test
ole@enterprise:~/test> cd ../Documents/
ole@enterprise:~/Documents> echo ~-
/home/ole/test
```

4.3.1 pwd

Der Befehl `pwd` (*Print Working Directory*) zeigt das Verzeichnis an, in dem Sie gerade arbeiten.

```
pwd
```

4.3.2 cd

Um in ein anderes Verzeichnis zu wechseln benutzt man das Kommando `cd` (*Change Directory*).

```
cd [DIRECTORY]
```

Das Zielverzeichnis kann dabei auf absolute oder relative Weise angegeben werden. Bei der absoluten Darstellung geht man immer von der Wurzel (*root*) aus. Diese Pfadangaben beginnen immer mit einem Slash `/`. Bei der relativen Pfadangabe wird dagegen von dem aktuellen Verzeichnis als Startpunkt ausgegangen. Jedes Verzeichnis enthält zwei besondere Verzeichnisse. Das Verzeichnis `.` steht für das Verzeichnis selber, während `..` für das übergeordnete Verzeichnis (Elternverzeichnis) steht.

Um wieder zurück ins Heimatverzeichnis zu kommen gibt es mehrere Wege. Sie können zum einen den absoluten oder relativen Pfad zum Verzeichnis angeben. Kürzer geht es aber mit den Befehlen `cd ~` oder ganz kurz nur mit `cd`.

4.3.3 ls

Eine Reihe von Befehlen ermöglicht es, sich den Inhalt eines Verzeichnisses anzusehen. Der am häufigsten benutzte Befehl ist `ls` (*LiSt*).

```
ls [OPTIONEN] [DATEINAME]
```

Für `DATEINAME` können Namen von Dateien oder Verzeichnissen[4] verwendet werden. Dabei kann, um eine Menge von Dateinamen zu bilden, Metazeichen (Joker) verwendet werden. Die bekanntesten Joker sind dabei

[3] "Homedirectory". Verwechseln Sie dieses Verzeichnis nicht mit dem Verzeichnis `/home`, in dem sich die Heimatverzeichnisse befinden.

[4] In Linux sind Verzeichnisse im Prinzip auch nur Dateien.

das Fragezeichen `?`, das für ein einzelnes beliebiges Zeichen steht, und der Asterisk `*`, der für eine beliebige Anzahl beliebiger Zeichen steht.

Optionen

`-a`	Anzeige aller Dateien, auch derjenigen, die mit . beginnen.
`-A`	Anzeige aller Dateien außer . und ..
`-c`	Sortiert zusammen mit `-t` Dateien nach dem Datum der letzten Änderung der Inode (Verwaltungsinformationen). Standardsortierfolge ist nach ASCII-Zeichen. Die Anzeige des entsprechenden Datums erfolgt zusammen mit `-l`. (`ls -ctl`)
`-d`	Verzeichnisse in der Liste der Argumente werden wie andere Dateien behandelt. Unterdrückung der Durchsuchung des Inhalts von Verzeichnissen.
`-l`	Anzeige ausführlicher Dateiinformationen. Viele Schalter haben nur im Zusammenhang mit `-l` eine Bedeutung, da sie nicht automatisch die Information anzeigen sondern nur vorbereiten (z. B. `-e` oder `-k`).
`-r`	Ausgabe in umgekehrter Sortierreihenfolge
`-t`	Sortiert Dateien nach dem Datum der letzten Änderung. Standardsortierfolge ist nach ASCII-Zeichen. Die Anzeige des Datums erfolgt zusammen mit `-l`.
`-x`	Spaltenweise Ausgabe der Dateinamen, im Gegensatz zu `-C` jedoch waagerecht geordnet.
`-m`	Ausgabe der Dateinamen als durch Komma getrennte Liste.
`-F`	Kennzeichnung diverser Dateitypen durch Anhängen von Sonderzeichen: `/` für Verzeichnisse, `*` für ausführbare Dateien, `@` für Links, `|` für FIFOs und `=` für Sockets.
`-R`	Rekursive Anzeige, d. h. es werden nicht nur Verzeichnisse nach ihrem Inhalt durchsucht, sondern auch darin enthaltene (Unter-) Verzeichnisse. Nicht zusammen mit `-d` anwendbar.

Eine Auflistung von weiteren Schaltern finden Sie in Tabelle 4.2. Es gibt eine Reihe von alternativen Optionen. Z. B. ist `--tabsize=zahl` eine Alternative zu `-T zahl`. Diese Optionen sind selbsterklärend (wenn man Englisch kann und Phantasie hat) und werden stets mit doppeltem Minuszeichen eingeleitet, da sie sonst wegen ihrer Länge mit Zusammenfassungen der herkömmliche Schalter verwechselt werden können. Man kann sie sich mit `ls --help` anzeigen lassen.

Über die Ausgabe des `ls`-Befehls sollten sie noch wissen:

- Die standardmäßig vorgegebene Reihenfolge der Ausgabe beruht auf einer lexikalischen Sortierung, wobei erst nach Ziffern, dann nach Großbuchstaben und zuletzt nach Kleinbuchstaben sortiert wird.
- Die Benutzung von `ls` ohne Schalter ergibt eine Liste in vertikal sortierten Spalten.
- Dateien, die mit einem Punkt `.` beginnen, werden als versteckte Dateien bezeichnet. Sie werden nur unter Verwendung der Schalter `-a` und `-A` angezeigt. Im Allgemeinen handelt es sich hier um Konfigurationsdateien.
- Die Farben bei der Ausgabe von `ls` werden in der Datei `/etc/DIR_COLORS` festgelegt. Um die Farben selbst bestimmen zu können, muß die Datei als `.dir_colors` in das Heimatverzeichnis kopiert werden.

Beispiel
Ausgabe des Befehls mit der Option -Fl

```
tapico@defiant:~/test > ls -Fl
total 4
lrwxrwxrwx   1 ole      users          9 May  9 22:00 hallo -> hallowelt*
-rwxr-xr-x   1 ole      users         36 May  9 21:56 hallowelt*
-rw-r--r--   2 ole      users        401 May  9 22:04 liste
-rw-r--r--   2 ole      users        401 May  9 22:04 myliste
drwxr-xr-x   2 ole      users       1024 May  9 21:55 mytest/
prw-r--r--   1 ole      users          0 May  9 21:54 testpipe|
```

Dabei enthalten die Spalten von links nach rechts gelesen: Dateityp und Rechte, Anzahl der Hardlinks, Besitzer, Gruppe, Größe in Byte, Monat, Tag, Uhrzeit, Dateiname und Dateityp (durch `-F`). Den Dateityp aus der ersten Spalte können Sie aus der Tabelle 4.3 entnehmen.

`-b`	Oktale Darstellung nichtdruckbarer Zeichen im Dateinamen (wichtiger Notnagel, wenn `ls` ein merkwürdiges Verhalten zeigt).
`-e`	Zusammen mit `-l` Ausgabe von Datum und Uhrzeit. (Nicht S.u.S.E.)
`-f`	Unsortierte Ausgabe (Dateien werden in der Reihenfolge angezeigt, in der sie im Verzeichnis eingetragen sind). Beinhaltet automatisch `-a`.
`-g`	existiert nur aus Kompatibilitätsgründen. Wird ignoriert.
`-i`	Anzeige der Inode-Nummer vor dem Dateinamen.
`-k`	Zusammen mit `-l` Anzeige der Dateigröße in Kilobyte (aufgerundet) vor dem Dateinamen (siehe auch `-s`)
`-n`	Ersetzt (in Verbindung mit `-l`) die Namen von Benutzer und Gruppe durch die entsprechenden Nummern (UID und GID).
`-o`	Keine farbliche Kennzeichnung der Dateitypen
`-p`	Kennzeichnung diverser Dateitypen durch Anhängen von Sonderzeichen. Ausführbare (gewöhnliche) Dateien werden im Gegensatz zu `-F` nicht gesondert gekennzeichnet.
`-q`	Nichtdruckbare Zeichen im Dateinamen werden als ? dargestellt (vgl. auch `-b`).
`-s`	Zusammen mit `-l` Angabe der Dateigröße in Blocks vor dem Dateinamen. Ist die Umgebungsvariable `POSIXLY_CORRECT` definiert, ist die Blockgröße 512 Byte, ansonsten (i. a.) ein Kilobyte (vgl. auch `-k`).
`-u`	Sortiert zusammen mit `-t` Dateien nach dem Datum der letzten Benutzung. Standardsortierfolge ist nach ASCII-Zeichen. Die Anzeige des entsprechenden Datums erfolgt zusammen mit `-l`. (`ls -utl`)
`-w zahl`	Legt die maximale Breite (*width*) der Ausgabezeile auf `zahl` fest. Wird durch `-l` aufgehoben.
`-B`	Keine Anzeige von Backup-Dateien (Enden mit ~)
`-C`	Spaltenweise Ausgabe der Dateinamen, senkrecht geordnet (dies ist im übrigen die Standardeinstellung. Sie wird durch `-l` aufgehoben).
`-D`	Erzeugt eine für einen bestimmten emacs-Modus geeignete Ausgabe.
`-G`	Unterdrückt bei `-l` die Anzeige der Gruppe
`-I muster`	Unterdrückt die Ausgabe von Informationen über Dateien, deren Name `muster` entspricht. `muster` kann Joker enthalten. Der Schalter kann mehrfach (mit unterschiedlichen Mustern) benutzt werden. Beachten Sie, daß auch hier Verzeichnisse besonders behandelt werden.
`-L`	Zeigt im Zusammenhang mit `-l` und `-I` bei symbolischen Links die Informationen über die Originaldatei und nicht die Daten des symbolischen Links an.
`-N`	Dateien mit Sonderzeichen im Namen werden nicht durch Anführungszeichen gekennzeichnet. (Voreinstellung)
`-Q`	Dateien mit Sonderzeichen im Namen werden durch Anführungszeichen gekennzeichnet. Die Sonderzeichen selbst werden in der aus der Programmiersprache C bekannten Form dargestellt.
`-S`	Sortierung der Dateinamen nach der Größe der Datei
`-T zahl`	Definiert den Abstand von Tabstopps (Tabulatoren).
`-U`	Unsortierte Ausgabe (im Gegensatz zu `-f` wird nicht der Schalter `-a` aktiviert).
`-X`	Sortiert die Ausgabe nach Dateiendung! Erstes Sortierkriterium ist der Namensteil nach dem letzten . im Namen. Z. B. werden Dateien, die mit .c enden, vor Dateien angezeigt, die mit .o enden. Dateien "ohne Endungen" werde zuerst angezeigt.
`-1`	Ausgabe der Dateinamen in nur einer Spalte (war in älteren UNIX-Systemen voreingestellt).

Tabelle 4.2: Weitere Optionen des Befehls `ls`

Kennzeichen	Bedeutung
-	gewöhnliche Datei
d	Verzeichnis (directory)
c	zeichenorientierte Gerätedatei (character device)
b	blockorientierte Gerätedatei (block device)
p	FIFO-Pipeline(named pipe)
l	symbolischer Link (symbolic link)
s	Netzwerksocket (socket)

Tabelle 4.3: Die Kennzeichnung der Dateiarten durch ls -l

4.3.4 dir

Der Befehl `dir` wird in den einzelnen Distributionen unterschiedlich behandelt. Er ist entweder ein Alias für `ls` oder für `ls -l`.

4.3.5 vdir

Das Kommando `vdir` ist fast identisch mit dem Kommando `ls`. Ohne Schalter eingegeben wirkt es wie `ls -l`.

4.3.6 mkdir

Der Befehl `mkdir` wird dazu benutzt neue Verzeichnisse zu erstellen.

```
mkdir [OPTIONEN] [VERZEICHNISPFADLISTE]
```

Wird die Option `-p` nicht mit angegeben, so müssen die Elternverzeichnisse für das neue Verzeichnis existieren.

Optionen	
`-p`	Erzeugt auch die nötigen Elternverzeichnisse, wenn diese nicht existieren.
`-m RECHTE`	Erlaubt gleich das Setzen der Rechte für das neue Verzeichnis wie durch `chmod`

Beispiele
Dieses Kommando legt das Verzeichnis `tex/linux` an. Dabei muß das Verzeichnis `tex` existieren.
`mkdir tex/linux`

Dieser Befehl legt einen kompletten Verzeichnispfad an. Dabei werden alle Verzeichnisse erzeugt, wenn sie noch nicht existieren.
`mkdir -p tex/linux/kurs/material`

Auf die neuen Verzeichnisse bekommt nur der Besitzer Rechte.
`mkdir -m 700 tagebuch adressen`

4.3.7 rmdir

Um leere Verzeichnisse wieder zu löschen, wird der Befehl `rmdir` verwendet.

```
rmdir [OPTIONEN] [VERZEICHNISPFADLISTE]
```

Dabei wird immer das letzte Verzeichnis in der angegebenen Verzeichnishierachie gelöscht. Der Schalter `-p` ermöglicht das Löschen ganzer Hierachien.

Optionen	
`-p`	Löschen aller leeren Verzeichnisse in der Verzeichnishierachie.

Beispiele
Dieses Kommando löscht alle Verzeichnisse des Elternverzeichnisses `linux`.
`rmdir tex/linux/*`

Soll die ganze Hierachie gelöscht werden, also auch die Verzeichnisse `tex/linux` und `tex` gelöscht werden, so lautet der Befehl:
`rmdir -p tex/linux/*`
Dabei dürfen die Verzeichnisse aber keine weiteren Verzeichnisse oder Dateien enthalten.

4.4 Der Linux-Dateibaum

Den Linux-Dateibaum gibt es eigentlich nicht. Jede Distribution hat einen individuellen Dateibaum für sich entwickelt. Allerdings gibt es einige Verzeichnisse, die eigentlich überall vorhanden sein sollten. Als Leitlinie für den Baum gilt der sogenannte Dateisystemstandard (*Filesystem-Hierachy-Standard*) **FHS**. Eine Übersicht über die wichtigsten Verzeichnisse liefert Abbildung 4.1.

- `/` Enthält alle anderen Verzeichnisse. Es ist die Wurzel des Dateisystems und wird auch Wurzelverzeichnis genannt.
- `/bin` Enthält die wichtigsten, für alle Benutzer verfügbaren LINUX-Befehle.
- `/boot` Enthält den Kernel und für das Booten wichtige Dateien.
- `/dev` Das Verzeichnis für die Gerätedateien.
- `/etc` Konfigurations- und Informationsdateien, darunter `magic`, `DIR_COLORS`, `input_rc` u. v. m.
- `/etc/rc.d` Enthält die für den Bootvorgang wichtigen Skripte und die `rc`-Verzeichnisse.
- `/etc/skel` Aus diesem Verzeichnis werden die Konfigurationsdateien (*skeleton user files*) für einen neuen Benutzer in dessen Heimatverzeichnis kopiert.
- `/etc/X11` Die Konfigurationsdateien des X-Window-Systems liegen hier.
- `/home` beherbergt die Heimatverzeichnisse, in denen normalerweise keine Rechte die Arbeit des Benutzers einschränken. Es ist somit für den normalen Benutzer von existentieller Bedeutung. In größeren LINUX-Systemen sollte für dieses Verzeichnis entsprechender Platzbedarf einkalkuliert werden, da hier die Benutzerdaten abgelegt werden. Häufig wird es dann auf einen Extra-Datenträger ausgelagert, der beim Systemstart separat gemountet wird.
- `/lib` Hier findet man die Bibliothek des C-Compilers, sowie sogenannte Shared Libraries, Programmbibliotheken (vereinfacht gesagt Programmteile), die von verschiedenen Anwendungen benutzt werden können. Diese Programmteile werden erst zur Laufzeit in den Arbeitsspeicher geladen, wenn das betreffende Programm sie benötigt. Dadurch und durch die Tatsache, daß verschiedene Programm darauf zugreifen können (shared) wird der Arbeitsspeicher wesentlich entlastet. Sind Sie Windows-Benutzer kennen Sie sicherlich .DLL-Dateien. Diese werden unter Windows entsprechend genutzt.
- `/Lost+found` Dieses Verzeichnis wird für wiederhergestellte Dateien verwendet.
- `/mnt` Hilfsverzeichnis zur Aufnahme von Dateisystemen, die in den LINUX-Verzeichnisbaum gemountet werden sollen.
- `/proc` ist ein virtuelles Dateisystem für Informationen über die laufenden Prozesse. Der Befehl `ps` (process status) liefert seine Angaben auf der Basis des Inhalts dieses Verzeichnisses. Jeder Prozeß erhält ein eigenes Unterverzeichnis in dem die Informationen über Ihn gespeichert werden.
- `/root` ist das Heimatverzeichnis der Superusers (**root**).
- `/sbin` Hier liegen die Befehle zur Systemverwaltung wie `adduser`, mit dessen Hilfe neue Benutzer definiert werden, `fsck` zur Überprüfung von Dateisystemen oder `shutdown`, mit dem das System heruntergefahren werden kann. Programme für Systemstart wie `init`, mit dem diverse Systemdienste (z. B. Drucker- und Zeitdienste) initiiert werden, darunter auch `getty` (ebenfalls in `/sbin` zu finden). Über `getty` erfolgt das Anmelden an der Konsole oder an Terminals, die via serieller Leitung mit dem LINUX-Rechner verbunden sind.
- `/tmp` Verzeichnis zum Zwischenlagern von Dateien. Hier darf jeder schreiben und lesen.
- `/usr` Enthält alle wichtigen Programme und Daten wie etwas das Online-Manual, die nicht zum Booten des Systems benötigt werden. Es kann ein Verzeichnis sein, das sich physikalisch auf einem anderen Datenträger befindet als das Bootmedium, auf einer CD-ROM etwa oder an einem entfernten NFS-Server. In dieser Situation müßte dieser Teil des Dateisystems nach dem Booten erst verfügbar gemacht werden. Für den Fall, daß es dabei zu Problemen kommen sollte, stehen die allernötigsten Befehle in verschiedenen anderen Verzeichnissen, z. B. `/bin`, zur Verfügung.

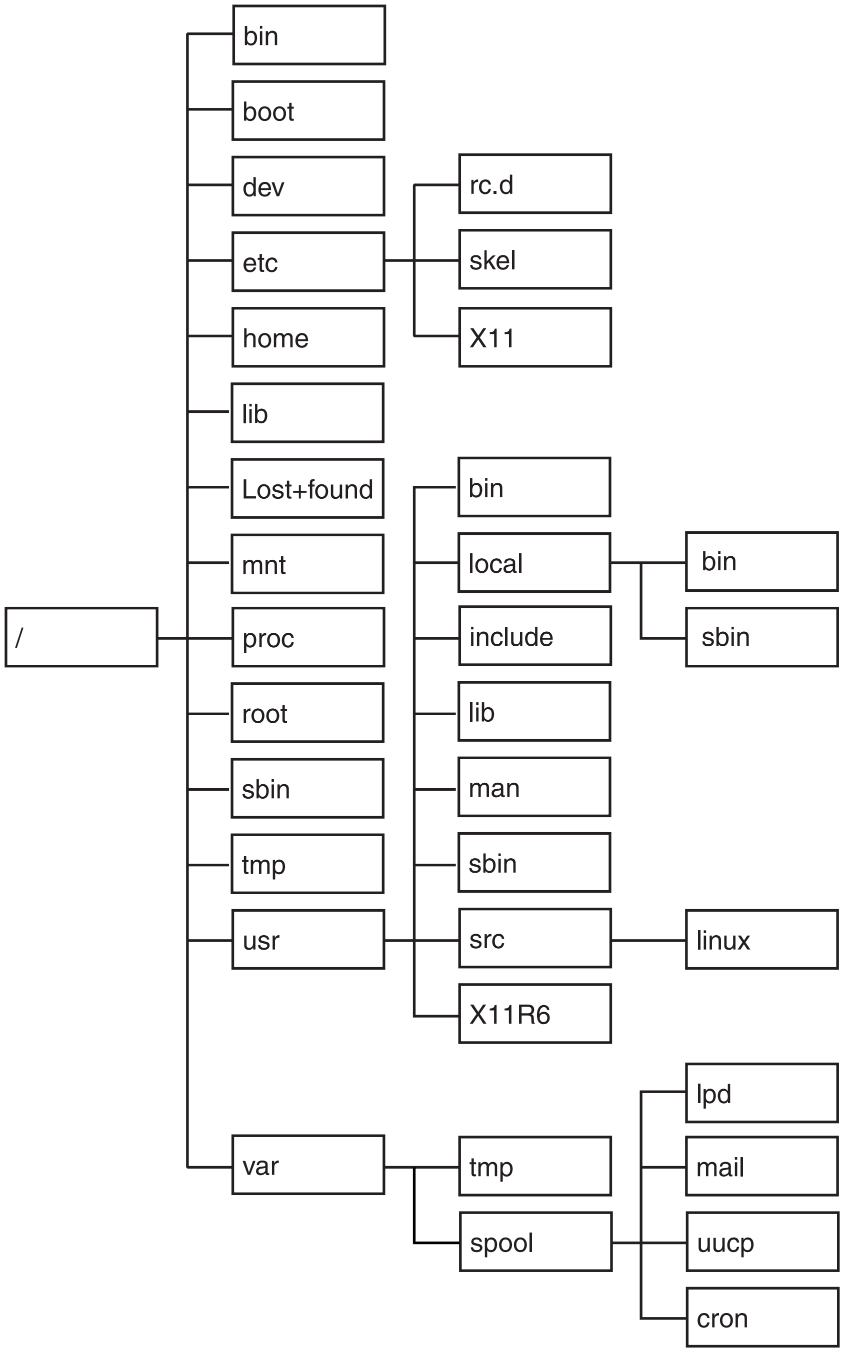

Abbildung 4.1: Der Linux-Dateibaum

- `/usr/bin` enthält die Befehle, die allen Teilnehmern zur Verfügung stehen sollten. Ein Teil dieser Benutzer-Befehle befindet sich aus den oben genannten Gründen (vgl. Eintrag bei `/usr`) in `/bin`.
- `/usr/local` wird für Programme verwendet, die nicht Bestandteil des Betriebssystems sind, wie z. B. das Dokumentenverzeichnis für den Webserver.
- `/usr/local/bin` Hier werden die Binärdateien von diesen Programmen gespeichert. Dieses Verzeichnis sollte im Suchpfad des Benutzers enthalten sein.
- `/usr/local/sbin` ist für lokal installierte Administrationstools zuständig.
- `/usr/include` enthält die Standard-C/C++-Header-Dateien.
- `/usr/lib` Die statischen Programmbibliotheken sowie die Unterverzeichnisse für Bibliotheken verschiedener Programmiersprachen sind hier zu finden. Enthält auch Links zu X-Window-Dateien.
- `/usr/man` ist eins der Verzeichnisse für das Online-Manual. Eine unerschöpfliche Informationsquelle für den Benutzer.
- `/usr/sbin` Analog zu `/usr/bin` gibt es dieses Verzeichnis für den Superuser (**root**).
- `/usr/src` enthält die Quellcodes für die Systemprogramme.
- `/usr/src/linux` In diesem Unterverzeichnis ist der Quellcode des Systemkerns zu finden.
- `/usr/X11R6` Enthält Dateien für das X-Window-System. Oft auch nur ein Link auf ein anderes Verzeichnis.
- `/var` Informationsdateien für das System sind hier abgelegt. Dies sind Tabellen, die von LINUX laufend verändert werden, z. B. die Dateien `utmp` und `wtmp`, die Informationen über die eingeloggten Benutzer enthalten. (Standard-UNIX: Die Tabellen befinden sich zum größten Teil in `/etc`.)
- `/var/tmp` ist das Ziel der temporären Dateien, die von Anwendungen aus verschiedenen Gründen während eines Programmlaufs angelegt und verändert, zum Programmende aber wieder gelöscht werden. Editoren oder Textverarbeitungen speichern unter Umständen den vom Anwender geschriebenen Text zur Sicherheit in einer temporären Datei. Temporäre Dateien sind kurzlebig und ständiger Änderung unterworfen. Sie sind ein sprechendes Beispiel für die allgemeine Bedeutung des Verzeichnisses `/var`.
- `/var/spool` ist der Ort der Warteschlangen (*Queues*) diverser Spoolprogramme. Dies sind Programme, die im Hintergrund arbeitend Aufträge der verschiedenen Benutzer entgegennehmen und in einer vernünftigen Reihenfolge abwickeln. Da mehrere Benutzer (beinahe) gleichzeitig z. B. einen Druckauftrag an den Druckspooler geben können, sprich drucken wollen, muß dieser dafür sorgen, daß die einzelnen Aufträge sich nicht vermischen und etwa am Drucker zwei Texte zu einem - bunt gemischten - werden. als Hilfsmittel dienen ihnen Warteschlangen, in die die einzelnen Aufträge nacheinander eingereiht und (nach dem Wirkungsprinzip einer FIFO) nach und nach abgearbeitet werden.
- `/var/spool/lpd` Warteschlangen des Druckerspoolers
- `/var/spool/mail` Warteschlange für die elektronische Post
- `/var/spool/uucp` Warteschlange für den Kommunikationsdienst uucp (unix-to-unix-copy)
- `/var/spool/cron` ist für die Abwicklung von Zeitdiensten über den cron-Dämonenprozeß wichtig, der in regelmäßigen Abständen Tabellen in `/var/spool/crontabs` daraufhin überprüft, ob sie Anweisungen enthalten, die zum gegenwärtigen Zeitpunkt auszuführen sind.

4.5 Arbeiten mit Dateien

4.5.1 touch

Das Kommando `touch` ändert die Zeit des letzten Zugriffs und der letzten Änderung auf die aktuelle Zeit. Existiert die Datei nicht, so wird eine neue Dateien erstellt.

```
touch [OPTIONEN] [DATEILISTE]
```

4.5.2 cat

Das Kommando `cat` (`concatenate file`) kann dazu benutzt werden neue Dateien zu erstellen. Es wird aber hauptsächlich dazu verwendet Dateien zusammenzufügen und sie auf dem Bildschirm oder einem anderen Gerät auszugeben.

```
cat [OPTIONEN] [DATEILISTE]
```

`cat` wird vor allem als Tool zum Anzeigen und Zusammenfügen von Dateien verwendet und als schneller Mini-Editor.

Optionen

`-b`	Nummeriert alle nichtleeren Zeilen durch
`-e`	Gleich mit `-vE`
`-n`	Nummeriert alle Zeilen durch (`-b` hat Vorrang)
`-E`	Fügt am Ende jeder Zeile ein $ ein
`-s`	Faßt aufeinanderfolgende Leerzeilen zu einer Leerzeile zusammen
`-v`	Zeigt alle nichtdruckbaren Zeichen durch Metazeichen an
`-T`	Zeigt Tabulatoren als Î an
`-A`	Zeigt alle nichtdruckbaren Zeichen, Tabulatoren als Î und das Zeilenende als $ an (vgl. `-vET`)

Beispiel
Fügt die Dateien `t1` und `t2` zusammen und schreibt sie in die Datei `t3`.
```
cat t1 t2 > t3
```
Ein einfacher Editor kann mit `cat` realisiert werden.
```
cat > text.txt
```
Dieser Befehl liest die Daten nicht aus einer Datei sondern von der Standardeingabe und schreibt sie dann in die Datei `text.txt`. Die Eingabe wird mit <CTRL>+<D> (EOF-Zeichen) beendet.

4.5.3 cp

Um Kopien von einer Datei oder einem Verzeichnis zu erstellen wird das Kommando `cp` verwendet.

```
cp [OPTIONEN] QUELLDATEI ZIELDATEI
cp [OPTIONEN] QUELLDATEILISTE ZIELVERZEICHNIS
```

Wird ein Dateiname für die Quelle angegeben, so kopiert `cp` die Datei in eine zweite Datei (`ZIELDATEI`). Ist der letzte Parameter ein Verzeichnis, dann wird die Datei in das Verzeichnis kopiert. Listen von Quelldateien können nur in ein Verzeichnis kopiert werden. Allerdings kann `cp` in der Normaleinstellung keine Verzeichnisse kopieren.

Optionen

`-a`	Kopiert die Dateien unter Beibehaltung von Struktur und Attributen
`-b`	Erzeugt von jeder überschriebenen Datei eine Sicherheitskopie
`-f`	Überschreiben von vorhandenen Zieldateien
`-i`	Nachfragen vorm Überschreiben von vorhandenen Zieldateien
`-l`	Legt Hardlinks anstatt Kopien der Dateien an
`-s`	Legt Softlinks anstatt Kopien der Dateien an
`-p`	Überträgt Besitzer, Gruppe, Rechte und Zeitmarken an die neue Datei
`-P`	Kopieren der Dateien inklusiver ihrer Verzeichnisstruktur
`-r`	Kopiert rekursiv Dateien aus Verzeichnissen
`-R`	Wie `-r`, aber anstatt die Inhalte der Dateien zu kopieren, werden die Datei wie sie sind kopiert
`-u`	Kopiert nur Dateien, die jünger sind als die Zieldateien
`-v`	Zeigt die Namen der kopierten Dateien an

Beispiel
Die folgende Kommandosequenz kopiert alle Dateien des aktuellen Verzeichnis unter Beibehaltung ihrer Strukturen.

```
cp -dpRP * \backup
```

4.5.4 dd

Das Programm `dd` (*Device to Device copy*) ist ein spezielles Kopierprogramm. Es wird in aller erster Linie dazu genutzt um Dateien von einem Gerät zu einem anderen zu kopieren.

```
dd [OPTIONEN]
```

Optionen

`if=DATEI`	Die Eingabedatei
`of=DATEI`	Die Ausgabedatei
`bs=BLOCKGRÖSSE`	Anzahl der Bytes, die auf einmal gelesen bzw. geschrieben werden.

Beispiel
Dieser Befehl legt eine Kopie einer Diskette in der Datei BackupDisk an.
`dd if=/dev/fd0 bs=512 of=BackupDisk`

4.5.5 Erstellung der Bootdisketten mit `dd`

Unter Linux können sie ebenfalls, wie unter DOS (Abschnitt 1.4.1) die Bootdisketten erstellen. Hierzu werden die gleichen Images im Verzeichnis `/disks` der CD 1 verwendet. Die Disketten müssen low-level-formatiert sein. Dazu kann der Befehl `fdformat` (Abschnitt 10.3.2) verwendet werden.

Nachdem die CD auf das Verzeichnis `/cdrom` gemountet worden ist, können mit den Befehlen

```
dd if=/cdrom/disks/bootdisk of=/dev/fd0 bs=8k
dd if=/cdrom/disks/modules1 of=/dev/fd0 bs=8k
dd if=/cdrom/disks/modules2 of=/dev/fd0 bs=8k
dd if=/cdrom/disks/modules3 of=/dev/fd0 bs=8k
```

die Bootdisketten erzeugt werden.

4.5.6 mv

Um Dateien zu verschieben oder umzubenennen wird das Kommando `mv` verwendet.

```
mv [OPTIONEN] ALTERDATEINAME NEUERDATEINAME
mv [OPTIONEN] QUELLDATEILISTE ZIELVERZEICHNIS
```

Werden zwei Dateinamen als Parameter vergeben, dann wird die Datei umbenannt. Ist der letzte Parameter ein Verzeichnis, dann werden die angegebenen Dateien in dieses Verzeichnis verschoben.

Optionen

`-b`	Erzeugt von jeder überschriebene Datei eine Sicherheitskopie
`-f`	Überschreiben von vorhandenen Zieldateien
`-i`	Nachfragen vorm Überschreiben von vorhandenen Zieldateien
`-u`	Verschiebt nur Dateien, die jünger sind als die Zieldateien

Beispiel
Die folgende Kommandosequenz benennt die Datei `megabox.txt` in die Datei `MEGABOX.txt` um.
`mv megabox.txt MEGABOX.txt`

Mit dem folgenden Kommando werden alle Dateien aus dem aktuellen Verzeichnis in das Verzeichnis `/home/harald` verschoben und eventuell vorhandene Zieldateien überschrieben.
`mv -f * /home/harald`

4.5.7 rm

Der Befehl `rm` löscht die angegebenen Dateien. In seiner Standardeinstellung löscht er keine Verzeichnisse.

```
rm [OPTIONEN] DATEILISTE
```

Um eine Datei zu löschen, benötigt man das Schreibrecht (`w`) auf das Verzeichnis, aber nicht auf die Datei. Sollte ein Datei nicht das Schreibrecht besitzen, so erbittet `rm` nur eine Bestätigung des Löschbefehls, wenn kein `-f` oder wenn ein `-i` gesetzt ist.

Optionen

`-d`	Löscht ein Verzeichnis durch Entfernen des Hardlinks. Volle Verzeichnisse werden auch gelöscht. Da die enthaltenen Dateien nicht mehr referenziert werden, ist es ratsam ein `fsck` danach auszuführen. (Nur Superuser)
`-f`	Löscht alle Dateien ohne explizites Nachfragen. Überlagert die Option `-i`
`-i`	Fragt vor jedem Löschen einer Datei um Bestätigung
`-r`	Löscht Verzeichnisse und deren Inhalt rekursiv

Beispiel
Der folgende Befehl löscht alle Dateien, die mit `.temp` enden.
`rm *.temp`

Vorsicht! Dieser Befehl versucht die gesamte Dateistruktur ohne Nachfragen zu löschen. Als **root** ausgeführt ist das Ergebnis fatal.
`rm -rf /`

4.5.8 Dateibezeichnung mit Jokerzeichen

Bei der Arbeit mit der Shell kommen Sie oft in Situationen, in denen Sie nicht nur einzelne Dateien oder alle Dateien in einem Verzeichnis ansprechen wollen, sondern nur eine Gruppe von Dateien (z. B. alle HTML-Dateien) bearbeiten möchten. Für diesen Zweck besitzt die Shell die sogenannten Jokerzeichen (*Wildcards*).

Wenn diese Jokerzeichen eingesetzt werden, gibt die Shell nicht einen einzelnen Dateinamen zurück, sondern eine Liste von Dateinamen, auf die das Muster zutrifft. So erhalten Sie mit dem Muster `*.jpg` eine Liste aller Dateinamen, die mit `.jpg` enden.

```
tapico@defiant:~> echo *.jpg
helm35.jpg helm37.jpg helm38a.jpg helm59.jpg
tapico@defiant:~> cp *.jpg ~/bilder
```

Denken Sie daran. Die Shell und nicht der Befehl interpretiert das Kommando. Deshalb müssen die Jokerzeichen in manchen Fällen maskiert werden, damit sie unbeschadet an den Befehl übergeben werden können. Probieren Sie den unteren Befehl mal ohne Backslash aus.

```
tapico@defiant:~> echo \* Hallo \*
* Hallo *
```

Eine Reihe von Jokerzeichen steht Ihnen zur Verfügung.

Asterisk *

Der Asterisk bedeutet im Prinzip alles oder nichts. Er steht für eine beliebige Anzahl beliebiger Zeichen. Wobei beliebige Anzahl auch kein Zeichen bedeuten kann. Z. B. stimmt das Muster `lk*` mit folgenden Dateinamen überein.

```
tapico@defiant:~> ls lk*
lk-aufbau-ext2.eps        lk-shell.aux       lk.log
```

```
lk-dateibaum.eps           lk-shell.tex         lk.pdf.gz
lk-dateisystem.aux         lk-shell.tex.bck     lk.ps
lk-dateisystem.tex         lk-vorwort.aux       lk.tex
lk-dateisystem.tex.bck     lk-vorwort.tex       lk.tex.bck
lk-installation.aux        lk-vorwort.tex.bck   lk.toc
lk-installation.tex        lk.aux               lktex
lk-installation.tex.bck    lk.dvi
```

Fragezeichen `?`

Das Fragezeichen steht für *genau ein* beliebiges Zeichen.

```
tapico@defiant:~> ls lk.*
lk.aux  lk.idx  lk.ind  lk.pdf.gz  lk.tex      lk.toc
lk.dvi  lk.ilg  lk.log  lk.ps      lk.tex.bck
tapico@defiant:~> ls lk.???
lk.aux  lk.dvi  lk.idx  lk.ilg  lk.ind  lk.log  lk.tex  lk.toc
tapico@defiant:~> ls lk.??
lk.ps
```

Menge `[ZEICHEN]`

Mit den eckigen Klammern ist es möglich eine Menge von Zeichen vorzugeben, die an dieser Stelle stehen können. So steht `[Aa]` für ein Zeichen, daß entweder ein großes oder ein kleines A sein kann. Ein Ausrufezeichen in der eckigen Klammer negiert die Bedeutung der Menge. So steht `[!aeiou]` für ein Zeichen, daß kein Vokal ist.

Innerhalb der Klammern können nicht nur einzelne Zeichen angegeben werden, sondern auch Bereiche. So steht z. B. `[a-z]` für alle Kleinbuchstaben und `[0-9]` für alle Zahlen.

```
tapico@defiant:~> ls lk.???
lk.aux  lk.dvi  lk.idx  lk.ilg  lk.ind  lk.log  lk.tex  lk.toc
tapico@defiant:~> ls lk.[it]??
lk.idx  lk.ilg  lk.ind  lk.tex  lk.toc
tapico@defiant:~> ls lk.[!it]??
lk.aux  lk.dvi  lk.log
tapico@defiant:~> ls [Hh]*
Home.txt  Hurra.jpg  hans.txt  himmel.jpg
tapico@defiant:~> ls [A-Z]*
Amerika.jpg  Home.txt  Hurra.jpg  Qualle.jpg
tapico@defiant:~> ls *[0-9][0-9].jpg
helm35.jpg  helm37.jpg  helm38.jpg  helm59.jpg
```

Klammerexpansion `{WORT1,WORT2,WORT3,...}`

Die Klammerexpansion (*brace expansion*) ist im eigentlichen Sinne kein Jokerzeichen. Sie funktioniert nämlich auch ohne existierende Dateien. Die Funktion ist einfach. Der Ausdruck wird jeweils mit dem in den geschweiften Klammern angegebenen und durch Kommata getrennten Zeichenketten ausgegeben.

```
tapico@defiant:~> echo "Ich bin "{gut,besser,super}"."
Ich bin gut. Ich bin besser. Ich bin super.
tapico@defiant:~> echo {1,2,3}{1,2,3}
11 12 13 21 22 23 31 32 33
```

Trotzdem können Sie die Klammerexpansion natürlich auch auf Dateioperationen anwenden.

```
tapico@defiant:~> ls lk-{kernel,shell}.*
```

```
lk-kernel.aux  lk-kernel.tex.bck  lk-shell.tex
lk-kernel.tex  lk-shell.aux       lk-shell.tex.bck
tapico@defiant:~> ls lk-*.{tex,eps}
lk-administration.tex  lk-dateibaum.eps    lk-installation.tex  lk-shell.tex
lk-aufbau-ext2.eps     lk-dateisystem.tex  lk-kernel.tex        lk-vorwort.tex
```

4.5.9 more

Der Befehl `more` gehört zur Gruppe der Befehle, mit denen man sich den Inhalt von Dateien anschauen kann (*pager*). Im Gegensatz zu `cat` (4.5.2) zeigt er die Daten aber seitenweise an.

```
more [OPTIONEN] DATEILISTE
```

Befehle	
`<LEERTASTE>`	Eine Seite nach unten scrollen
`<f>`	Eine Seite nach unten scrollen
`<b>`	Eine Seite nach oben scrollen
`<RETURN>`	Eine Zeile nach unten scrollen
`<s>`	Eine Zeile nach unten scrollen
`</>MUSTER`	Durchsuchen des Textes nach dem regulären Ausdruck `MUSTER`
`<n>`	Weitersuchen nach unten
`<q>`	Beenden

4.5.10 less

Der Befehl `less` ist die Weiterentwicklung von `more`[5].

```
less [OPTIONEN] DATEILISTE
```

Neben den Möglichkeiten von `more` bietet `less` die Möglichkeit mit den Richtungstasten zu scrollen und mit Lesezeichen, Zeilennummern und prozentualen Textpositionen zu arbeiten. Auch wird `less` nicht am Ende der Datei beendet.

Zusätzliche Befehle	
`<j>`	Eine Zeile nach unten scrollen
`<k>`	Eine Zeile nach oben scrollen
`<g>`	An den Anfang der Datei scrollen
`<G>`	Ans Ende der Datei scrollen
`</>MUSTER`	Durchsuchen des Textes nach dem regulären Ausdruck `MUSTER` nach unten
`<?>MUSTER`	Durchsuchen des Textes nach dem regulären Ausdruck `MUSTER` nach oben
`<n>`	Weitersuchen nach unten
`<N>`	Weitersuchen nach oben

4.5.11 lesskey

Das Programm `lesskey` erlaubt eine individuelle Konfiguration der Steuerbefehle von `less`.

4.5.12 file

Der Befehl `file` analysiert Dateien und gibt ihren Typ aus.

```
file DATEILISTE
```

`file` prüft zunächst anhand der Inode-Informationen, um welchen Datentyp es sich handelt, prüft aber zusätzlich den Anfang der Datei anhand der Informationen, die in der Datei `/etc/magic` enthalten sind.

[5] Und weil er mehr kann als `more` wird er `less` genannt `;-)`

Optionen

`-b`	Ausgabe ohne Angabe der Dateinamen
`-f DATEINAME`	Die Dateiliste wird aus der Datei `DATEINAME` entnommen (Ein Dateiname pro Zeile).
`-z`	Bearbeitet auch den Inhalt von gepackten Dateien

Beispiel

```
ole@defiant:~ > ls -Fl
total 5
-rw-r--r--   1 ole      users         468 May  9 22:28 194.195.155.105
lrwxrwxrwx   1 ole      users           9 May  9 22:00 hallo -> hallowelt*
-rwxr-xr-x   1 ole      users          36 May  9 21:56 hallowelt*
-rw-r--r--   2 ole      users         401 May  9 22:04 liste
-rw-r--r--   2 ole      users         401 May  9 22:04 myliste
drwxr-xr-x   2 ole      users        1024 May  9 21:55 mytest/
prw-r--r--   1 ole      users           0 May  9 21:54 testpipe|
ole@defiant:~ > file *
194.195.155.105: ASCII text
hallo:           symbolic link to hallowelt
hallowelt:       perl commands text
liste:           ASCII text
myliste:         ASCII text
mytest:          directory
testpipe:        fifo (named pipe)
```

4.5.13 /etc/magic

Die Datei `/etc/magic` ist eine Liste von sogenannten “Magic Numbers”. Dies Magic Numbers sind die Erkennungsmelodien verschiedener Dateiformate, die meistens am Anfang einer Datei stehen.

Die Datei wird von verschiedenen Programmen gebraucht, wie z. B. `file` (4.5.12).

Der Eintrag z. B. für eine Bilddatei im PNG-Format ist:

```
# PNG [Portable Network Graphics, or "PNG's Not GIF"] images
# (Greg Roelofs, newt@uchicago.edu)
# (Albert Cahalan, acahalan@cs.uml.edu)
#
# 137 P N G \r \n ^Z \n [4-byte length] H E A D [HEAD data] [HEAD crc] ...
#
0       string          \x89PNG         PNG image data,
>4      belong          !0x0d0a1a0a     CORRUPTED,
>4      belong          0x0d0a1a0a
>>16    belong          x               %ld x
>>20    belong          x               %ld,
>>24    byte            x               %d-bit
>>25    byte            0               grayscale,
>>25    byte            2               \b/color RGB,
>>25    byte            3               colormap,
>>25    byte            4               gray+alpha,
>>25    byte            6               \b/color RGBA,
#>>26   byte            0               deflate/32K,
>>28    byte            0               non-interlaced
>>28    byte            1               interlaced
1       string          PNG             PNG image data, CORRUPTED
```

Die Datei kann sich auch im Verzeichnis `/usr/share/misc/magic` befinden. Dann ist `/etc/magic` nur ein Link auf diese Datei. Weitere Informationen können Sie den Manualpages (`man 4 magic`) entnehmen.

4.6 Weitere Befehle

4.6.1 clear

Dieser Befehl löscht den Bildschirm.

```
clear
```

4.6.2 cal

Das Kommando `cal` zeigt einen Kalender an.

```
cal [[MONAT] JAHR]
```

Für `JAHR` sind Werte zwischen 1 und 9999 erlaubt und für `MONAT` zwischen 1 und 12.

Optionen

`-m`	Zeigt Montag als ersten Tag der Woche an
`-j`	Benutzt das Julianische Datum
`-y`	Zeigt einen Kalender für das aktuelle Jahr

4.6.3 date

Diese Befehl zeigt und setzt das aktuelle Datum und die Uhrzeit. Allerdings kann nur *root* die Zeit setzen.

Zeit anzeigen

Die Anzeige der aktuellen Zeit erfolgt über

```
date [+FORMATSTRING]
```

`FORMATSTRING` ist ein beliebiger Text, der, falls er Sonderzeichen enthält, in Anführungszeichen zu setzen ist. Sinn macht dieser Text allerdings erst, wenn er Platzhalter enthält, die Datum oder Zeit anzeigen. Einige Beispiele für solche Platzhalter sind:

`%m`	Monat	`%H`	Uhrzeit
`%d`	Tag	`%M`	Minuten
`%y`	Jahr (zweistellig)	`%S`	Sekunden
`%Y`	Jahr (vierstellig)	`%T`	Zeit (hh:mm:ss)

Zeit stellen

Der Systemverwalter kann auch Zeit und Datum über diesen Befehl ändern.

```
date MMTTSSmm[[HH]JJ][.ss]
```

Wobei `MM` für Monat, `TT` für Tag des Monats, `SS` für Stunde, `mm` für Minute, `HH` fürs Jahrhundert, `JJ` fürs Jahr und `ss` für die Sekunden stehen.

Die Änderung der Systemzeit hat keine Auswirkungen auf die Hardware-Uhr. Nach dem Neustart des Systems sind die Einstellungen wieder verloren.

Für weitere Informationen zu Schaltern und Platzhaltern geben Sie `man date` ein.

Beispiele

```
tapico@defiant:~> date +"Es ist der %d.%m.%y"
```

```
Es ist der 19.01.03
tapico@defiant:~> date +"Es ist %T Uhr"
Es ist 13:47:35 Uhr
```

Um die Zeit zu setzen brauchen Sie die Berechtigung des Systemadministrators. Wenn Sie als normaler Benutzer eingeloggt sind, wechseln Sie zuerst mit dem Befehl `su` (8.1.2) ihre Identität zu *root*. Danach können Sie das Datum einstellen.

```
tapico@defiant:~> su
Password:
root@defiant:/home/tapico # date 10221735
Mon Okt 22 17:35:00 CEST 2001
```

4.6.4 hwclock

Der Befehl `date` beschäftigt sich nur mit der Systemzeit. Zum Auslesen und Setzen der Hardware-Uhr können Sie den Befehl `hwclock` verwenden. Dieser Befehl steht nur dem Systemadministrator zur Verfügung.

```
hwclock [OPTIONEN]
```

Für die Anwendung und Möglichkeiten dieses Befehls konsultieren Sie bitte die Manualpage (`man 8 hwclock`).

Um neben der Systemzeit auch die Hardware-Uhr zu setzen, gehen Sie bitte wie folgt vor.

1. Wechseln Sie ihre Identität zu *root*.
2. Setzen Sie mit dem Befehl `date` (4.6.3) die Systemzeit.
3. Setzen Sie nun mit den Befehl `hwclock` die Systemzeit und Hardware-Uhr gleich.

Das kann dann so aussehen:

```
ole@defiant:~> su
Password:
defiant:/home/ole # date 01191418
Son Jan 19 14:18:00 CET 2003
defiant:/home/ole # hwclock --systohc
```

4.6.5 echo

Der Befehl `echo` wirkt wie ein Echo, indem er einfach den angegebenen Text wieder auf den Bildschirm ausgibt. Seine Haupteinsatz findet er daher in den Shellskripten, der Ausgabe von Texten in Dateien und dem Anzeigen der Inhalte von Variablen.

```
echo [OPTIONEN] [TEXT]
```

Optionen

`-e`	erlaubt die Benutzung von Sonderzeichen im Text, z. B: `\n` Zeilenumbruch `\a` Piepton (ASCII-Zeichen 7) `\b` Backspace `\f` Seitenvorschub `\ooo` beliebiges ASCII-Zeichen in oktaler Notation
`-n`	unterdrückt Zeilenumbruch nach Ausgabe des Textes

Texte mit Sonderzeichen (z. B. Leerzeichen) sollten in Anführungszeichen stehen. `echo` ohne Angabe eines Textes bewirkt die Ausgabe einer Leerzeile.

{attr}		{fg}		{bg}	
0	Normaler Modus	30	Schwarz	40	Schwarz
1	Hell oder Fett	31	Rot	41	Rot
2	Dunkel	32	Grün	42	Grün
3	Unterstrichen	33	Gelb	43	Gelb
5	Blinkend	34	Blau	44	Blau
7	Invertiert	35	Margenta	45	Margenta
8	Versteckt	36	Türkis	46	Türkis
		37	Weiß	47	Grau
				48	Weiß

Tabelle 4.4: Farbcodes fürs Terminal

4.6.6 Farbcode fürs Terminal

Um Terminals zu konfigurieren werden Codesequenzen benutzt. Sie werden als Escape-Sequenzen bezeichnet, da sie mit dem Escape-Zeichen (0x1B)[6] beginnen. So kann die Farbe der Ausgabe festgelegt werden. Mit der Sequenz <ESC>[{attr};{fg};{bg}m kann nun die Farbe vorgegeben werden.

So bewirkt der Befehl
echo "^[[0;31;40mIch sehe rot"
daß der Text 'Ich sehe rot' in roter Farbe ausgegeben wird. Das erste Zeichen ist das Escape-Zeichen. Es wird als ^[dargestellt. Um es zu schreiben drücken Sie <STRG>+<V> und dann <ESC. Um wieder den Normalzustand herzustellen, müssen Sie für die Konsole den Befehl
echo "^[[0;37;40m"
eingeben.

Eine Liste der möglichen Farben und Attribute liefert Tabelle 4.4. Diese Werte werden auch in den Dateien /etc/DIR_COLORS und .dir_colors für die Farbcodierung verwendet. Eine vollständige Liste der Konsolen-Codes liefert die Manpage *console_ codes (4)*.

```
ole@enterprise:~> man 4 console_codes
```

4.6.7 logout

Dieses bash-Kommando beendet die Sitzung eines Benutzers.

```
logout
```

4.6.8 script

Der Befehl script sorgt dafür, daß alle Ausgaben auf den Bildschirm gleichzeitig in einer Datei mitgeloggt werden. Dabei wird alles mitgeschrieben bis der Benutzer sich wieder ausloggt. Dieses Kommando eignet sich sehr gut zum Protokollieren von Aktionen.

```
script [OPTIONEN] [DATEI]
```

Wird kein Dateiname angegeben, so wird automatisch die Datei typescript angelegt.

4.6.9 telnet

Natürlich gibt es auch eine Möglichkeit eine Linux-Rechner als Terminal zu verwenden. Das Programm telnet ermöglicht den Zugang zu anderen Rechnern. Im Gegensatz zu Windows sehen Sie im Textterminal keine Unterschied ob Sie direkt an der Konsole oder über telnet arbeiten.

[6] Zahlen, die mit 0x beginnen, sind in hexadezimaler Schreibweise dargestellt. 0x1B ist somit die Zahl 27 $(16 + 11)$.

```
telnet HOST [PORT]
```

Achtung! Aus Sicherheitsgründen können Sie sich nicht als *root* über Telnet an einem Rechner anmelden. Sie müssen sich erst als normaler Benutzer einloggen und können dann einen Identitätswechsel zum Superuser vollziehen.

Weitere Informationen über das Programm `telnet` finden Sie in den Manual-Pages (`man telnet`).

4.6.10 tty

Die Ausgabe eines Befehls landet immer auf der Konsole oder dem Terminal, auf dem es gestartet wurde. Aber auf welcher Konsole bin ich denn überhaupt? Da hilft der Befehl `tty` weiter.

```
tty
```

Beispiel

```
tapico@defiant:~ > tty
/dev/pts/0
```

Notizen:

IV	Arbeitsblatt Shell 1	4.1

Sollte eine Aufgabe zu einer Fehlermeldung führen, kann das von mir gewollt sein! Prüfen Sie aber dennoch, ob Sie keinen Tippfehler gemacht haben, und ob die Voraussetzungen wie in der Aufgabenstellung gegeben sind. Auch sollten keine Verzeichniswechsel ausgeführt werden, wenn dies nicht ausdrücklich in der Aufgabe verlangt wird! Notieren Sie die Ergebnisse auf einem seperaten Zettel.

1 Loggen Sie als Benutzer *walter* auf Konsole 1 ein!

2 Führen Sie den Befehl `rm -rf *` aus. Was bewirkt der Befehl?

3 Legen Sie eine leere gewöhnliche Datei mit dem Namen `ogottogottogott` an.

4 Wechseln Sie ins Elternverzeichnis (Nach .. also)

5 In welchem Verzeichnis sind Sie jetzt?

6 Stellen Sie fest, welche Dateien hier sind!

7 Löschen Sie den Bildschirm. (Nicht mit Wasser!)

8 Wechseln Sie in das Verzeichnis mit Ihrem Benutzernamen!

9 Lassen Sie sich den Text “Moin, Moin” begleitet von einem Piepton auf dem Bildschirm ausgeben.

10 Welche Dateien gibt es im aktuellen Verzeichnis?

11 Wechseln Sie zur Wurzel! Welche Dateien gibt es hier?

12 Wechseln Sie in das Verzeichnis boot!

13 Welche Dateien finden Sie hier?

14 Wechseln Sie in das Verzeichnis `/dev`?

15 Prüfen Sie den Inhalt des Verzeichnisses! Kommen Ihnen einige Namen bekannt vor?

16 Wechseln Sie ins übergeordnete Verzeichnis (nach .. also).

17 Wechseln Sie ins Verzeichnis `/bin` !

18 Stellen Sie fest, welche Dateien mit dem Anfangsbuchstaben l (wie Ludwig) vorhanden sind.

19 Lassen Sie sich ausführliche Informationen über die Dateien im Verzeichnis `/etc` geben!

20 Stellen Sie fest, in welchem Verzeichnis Sie sind! Wechseln Sie, falls notwendig, ins Verzeichnis `/bin` !

21 Legen Sie ein Verzeichnis mit Ihrem Benutzernamen in `/bin` an!

22 Wechseln Sie in Ihr Heimatverzeichnis!

23 Legen Sie im Heimatverzeichnis die Verzeichnisse `test`, `org`, `org/hh` und `org/ki` an!

24 Stellen Sie fest, wem die Verzeichnisse `test` und `org` in Ihrem Heimatverzeichnis gehören!

25 Erstellen Sie (mittels `echo text > datei`) eine Datei namens `keks` in ihrem Heimatverzeichnis (Inhalt beliebig)!

26 Wechseln Sie nach `org/hh` !

27 Stellen Sie fest, in welchem Verzeichnis Sie sind!

28 Legen Sie (wie in Aufgabe 25) eine Datei `bonbon` im aktuellen Verzeichnis an!

29 Lassen Sie sich den Inhalt des aktuellen Verzeichnisses anzeigen!

IV	Arbeitsblatt Shell 1	4.2

30 Lassen Sie sich den Inhalt Ihres Heimatverzeichnisses anzeigen!

31 Wechseln Sie ins Heimatverzeichnis!

32 Löschen Sie die Verzeichnisse `org` und `test` mit dem Befehl `rmdir`!

33 Stellen Sie fest, welche Dateien in `/usr/bin` sind!

34 Lassen Sie sich alle Dateien/Verzeichnisse des Heimatverzeichnisses anzeigen außer den der Standardeinträgen `.` und `..` !

35 Lassen Sie sich die Inode-Nummern der Dateien des Verzeichnisses `/usr/bin` anzeigen!

36 Wann wurde zuletzt auf die Datei `/usr/share/doc/packages/shadow/HOWTO` zugegriffen?

37 Wie groß (Byte) ist die Datei `/etc/passwd` ?

38 Wem gehört die Datei `/boot` ?

39 Wieviele Links gibt es für Ihr Heimatverzeichnis?

40 Lassen Sie sich die gesamte Verzeichnisstruktur (Unterverzeichnisse) von `/usr` anzeigen!

41 Welche Dateien in dem Verzeichnis `/boot` beginnen mit `:`?

42 Gibt es in `/dev` ausführbare Dateien?

43 Vergleichen Sie die Inode-Nummern der Dateien `.` und `..` in der Wurzel!

44 Welche Inode-Nummern haben die Dateien `mv`, `cp` und `ln` in `/bin` ?

45 Lassen Sie sich ausführliche Informationen über das aktuelle Verzeichnis anzeigen, nicht aber über die darin enthaltenen Dateien!

46 Kopieren Sie alle Dateien aus `/usr/share/doc/packages/shadow` ins aktuelle Verzeichnis!

47 Prüfen Sie die Zeit der letzten Änderung von `HOWTO` im aktuellen Verzeichnis und in `/usr/share/doc/packages/shadow`!

48 Erstellen Sie die Verzeichnisse PRIMA, PrimA und prima im aktuellen Verzeichnis!

49 Verschieben Sie `HOWTO` aus dem aktuellen Verzeichnis nach `prima` !

50 Kopieren Sie die Dateien `README` und `ogottogottogott` aus dem aktuellen Verzeichnis in einem Arbeitsgang nach `prima` !

51 Wechseln Sie nach `prima` !

52 Benennen Sie `ogottogottogott` um in `meingottwalter` !

53 Benennen Sie `README` um in `.juhe` !

54 Kopieren Sie alle Dateien des aktuellen Verzeichnisses in einem Zug ins Heimatverzeichnis! Wechseln Sie danach ins Heimatverzeichnis !

55 Wann wurde die Datei `ogottogottogott` zum letzten Mal geändert?

56 Setzen Sie die Zeit des letzten Zugriffs und der letzten Änderung auf die aktuelle Zeit.

57 Löschen Sie das komplette Verzeichnis `prima`, unterdrücken Sie dabei Rückfragen!

58 Lassen Sie sich den Text “Fertig” in roter blinkender Schrift auf dem Bildschirm ausgeben. Achten Sie dabei darauf, daß der Prompt wieder dem ursprünglichem Schema entspricht.

59 Löschen Sie den Bildschirm.

Kapitel 5

Die Shell II

5.1 Die gewissen Extras

Um dem Benutzer die Arbeit mit dem Rechner zu erleichtern, besitzt die Shell einige sehr nützliche Funktionen. Das fängt mit dem Editor für die Eingabe der Befehle an, umfaßt Hilfen zur Eingabe langer Dateinamen und vieles mehr.

5.1.1 Readline Library

Wenn Sie auf der Eingabezeile arbeiten, benutzen Sie den Editor Readline Library der Free Software Foundation. Standardmäßig arbeitet er im `emacs`-Modus. D. h. er verwendet zur Steuerung die gleichen Befehle wie der Editor `emacs`.

Eine Übersicht über die Steuerungsbefehle finden Sie in Tabelle 5.1.

Anstatt der Readline Library kann man auch einen anderen Editor für die Eingabezeile benutzen. Um z. B. den `vi` zu benutzen, geben Sie ein:

```
set -o vi
```

Zurück kommen Sie mit

```
set -o emacs
```

5.1.2 inputrc

Die Konfigurationsinformationen für die Eingabezeile befinden sich in der Datei `/etc/inputrc`. Um diese Einstellungen individuell zu ersetzen, kann im Heimatverzeichnis die Datei `.inputrc` erstellt werden. Die Informationen dieser Konfigurationsdatei werden nur beim Einloggen ausgelesen.

So kann z. B. die Tastaturbelegung selbst definiert werden. Die Codesequenzen, die der Shell für die Tasten übermittelt werden, können von der Eingabeoberfläche abhängen. So steht die Sequenz `\e[11~` für die Taste `<F1>` bei einem X-Window-Terminal (xterm, kvt) oder einem Telnet-Terminal. Dagegen übermittelt die normale Linuxkonsole (linux) die Codesequenz `\e[[A` für das Drücken der Taste `<F1>`.

Um die Taste `<F1>` mit dem Befehl `ls -l` zu belegen, muß folgender Eintrag in die `.inputrc` gemacht werden.

```
"\e[11~": "ls -l"
```

oder

```
"\e[[A": "ls -l"
```

Jedesmal, wenn die Taste `<F1>` gedrückt wird, erscheint nun `ls -l` am Prompt. Soll der Befehl dann auch noch sofort ausgeführt werden, dann muß nur noch ein `\n` hinten angehängt werden.

```
"\e[11~": "ls -l\n"
```

Soll eine andere Datei für die Konfiguration verwendet werden, so muß diese mit ihrem Pfad in der Umge-

Tasten	Aktion
`<CTRL>+<b>`	Ein Zeichen nach links
`<CTRL>+<f>`	Ein Zeichen nach rechts
`<ESC>+<b>`	Anfang des nächsten Wortes links
`<ESC>+<f>`	Anfang des nächsten Wortes rechts
`<CTRL>+<a>`	Anfang der Zeile
`<CTRL>+<e>`	Ende der Zeile
`<DEL>`	Löscht Zeichen links vom Cursor
`<CTRL>+<d>`	Löscht Zeichen rechts vom Cursor
`<ESC>+<DEL>`	Löscht das Wort links vom Cursor
`<ESC>+<d>`	Löscht das Wort rechts vom Cursor
`<CTRL>+<k>`	Löscht vom Cursor bis zum Ende der Zeile

Tabelle 5.1: Steuerbefehle für den `emacs`
Hinweis: Bei manchen Editoreinstellungen löscht die Taste `<DEL>` das rechte Zeichen und `<BACKSPACE>` das linke Zeichen.

bungsvariablen `INPUTRC` angegeben werden. Bei der SuSE-Distribution ist existiert schon eine fertige Datei `/etc/inputrc`. Wenn Sie für Benutzer eine individuelle Konfigurationsdatei anlegen wollen, dann kopieren Sie einfach die Datei aus dem Verzeichnis `/etc` als `.inputrc` in das jeweilige Heimatverzeichnis und führen dann die Änderungen dort aus.

5.1.3 Automatische Kommandoergänzung

Die Shell ist konzipiert worden, um dem Benutzer eine komfortable Umgebung zum Arbeiten zu schaffen. Ein Tool dafür ist die automatische Kommandoergänzung. Oft reicht es aus, nur die ersten Zeichen eines Befehls oder Dateinamens einzugeben um diesen eindeutig zu identfizieren. Durch Betätigen der Tabulator-Taste (`<TAB>`)versucht die Shell diesen Namen zu vervollständigen.

So erweitert die Shell nach Betätigung der Tabulator-Taste die Buchstaben `ec` zu dem Befehl `echo`.

Ist der Befehl der Dateiname nicht eindeutig über die ersten Zeichen definiert, so gibt die Shell ein Signalton zurück. Wird die Tabulator-Taste noch einmal betätigt, zeigt die Shell eine Liste von Kommandos, die mit den entsprechenden Zeichen beginnen.

So erhalten Sie beispielsweise nach Eingabe von `mk` und zweimaligem Betätigen der Tabulator-Taste folgende Liste[1].

```
mkdir       mkfontdesc  mkmanifest     mktemp         mktexpk
mkdirhier   mkfontdir   mknod          mktexlsr       mktexftm
mkfifo      mkindex     mksusewmrc     mktexmf
```

Auf diese Art und Weise können Sie sich auch eine Liste aller zur Verfügung stehenden Befehle ausgeben lassen. Drücken Sie einfach am leeren Prompt zweimal die Tabulator-Taste. Da es nicht gerade wenig Befehle sind, werden Sie dann erst einmal von der Bash gefragt, ob Sie auch wirklich alle Treffer anzeigen lassen möchten.

```
tapico@defiant:~><TAB><TAB>
Display all 2192 possibilities? (y or n)
```

Genauso kann man sich auch schnell durch das Dateisystem bewegen. Durch die Eingabe von
`cd /h<TAB>t<TAB>mo<TAB>l<TAB>`
erhält man z. B. die Kommandosequenz `cd /home/tapico/moebius/linux`.

Auch wenn Sie den Pfad nicht kennen, hilft die Tabulatortaste weiter. Einfach zweimal die Tabulatortaste drücken und alle in Frage kommenden Verzeichnisse werden angezeigt.

[1] Die Liste ist natürlich von System zu System unterschiedlich, je nachdem welche Programme installiert sind.

```
tapico@defiant:~> cd /var/<TAB><TAB>
X11R6     games     log       opt       squid     yp
adm       lib       mail      run       tmp
cache     lock      named     spool     ucd-snmp
tapico@defiant:~> cd /var/
```

Und was für Verzeichnisse gilt, gilt auch für alle anderen Dateien. Dann natürlich nicht mit `cd`, sondern einem dateibezogenen Befehl wie z. B. `less`.

```
tapico@defiant:~> less /etc/cron<TAB><TAB>
cron.d        cron.hourly   cron.weekly   crontab.old
cron.daily    cron.monthly  crontab
tapico@defiant:~> less /etc/cron
```

5.1.4 Gruppierung von Kommandos

Unter Linux ist es möglich, mehrere Kommandos in einer Zeile einzugeben. Dabei werden die einzelnen Kommandos durch das Zeichen '`;`' getrennt.

```
tapico@defiant:~/test/ > date; echo "Homeverzeichnis"; ls ~
Die Sep 19 23:55:00 MEST 2000
Homeverzeichnis
Desktop     datum       public_html             test.pl         who
Mail        datum.old   public_html_old         test1.pl        whoami
cgi-bin     manpages    test                    test2.pl        zulu.txt
```

Damit die Ausgabe aller Befehle in eine Datei umgeleitet werden kann, müssen die Befehle gruppiert werden. Denn die Befehlssequenz

```
date; echo "Homeverzeichnis"; ls ~ > dir.txt
```

würde nur die Ausgabe des `ls`-Befehls in die Datei umlenken. Richtig muß die Kommandosequenz lauten:

```
(date; echo "Homeverzeichnis"; ls ~) > dir.txt
```

Die Klammern gruppieren die Kommandos zu einem Befehl. Deshalb wird dies als Kommandogruppierung bezeichnet.

5.1.5 Bedingte Ausführung

Die Zeichen `&&` und `||` verknüpfen zwei Befehle miteinander, wobei der zweite Befehl in Abhängigkeit vom Erfolg des ersten Befehls ausgeführt wird.

```
less index.htm || less index.html
```

bewirkt, daß nur wenn der erste Befehl keinen Erfolg hat, weil z. B. die Datei nicht existiert, der zweite Befehl ausgeführt wird.

```
cat index.html && cp index.html /backup
```

bewirkt, daß nur wenn der erste Befehle erfolgreich war, der zweite Befehl ausgeführt wird.

5.1.6 Substituierung von Kommandos

Oft ist es sinnvoll die Ausgabe eines Kommandos in einem anderen Kommando zu verwerten. Das Einschließen eines Kommandos in `$(...)` oder in `` `...` `` bewirkt, daß die Ausgabe des Kommandos verwendet wird.

```
tapico@defiant:~ > echo $(pwd)
/home/tapico
tapico@defiant:~ > echo `pwd`
/home/tapico
```

Die obere Methode ist die neue Syntax der `bash`. Die zweite zeigt die Syntax der Bourne Shell. Mindestens eine der beiden Methode wird in der `bash` funktionieren.

Beispiel Sie wollen den Kalender des aktuellen Monats in eine gleichnamigen Datei speichern.

```
tapico@enterprise:~/cal> cal -m > kalender-$(date +"%Y-%m").txt
tapico@enterprise:~/cal> ls -l kalender-*
-rw-r--r--    1 tapico   users         129 Apr 14 12:57 kalender-2002-04.txt
```

5.2 Variablen

Um bestimmte, immer wiederkehrende Zeichenfolge zu speichern, werden in der Shell sogenannte Variablen verwendet. Jede Variable besitzt einen Namen, über den auf diese Zeichenfolge zugegriffen werden kann. Einer *Benutzervariable* kann ein Name und ein Wert zugewiesen werden, während die Namen der *Umgebungsvariablen* festgelegt sind. Eine besondere Form der Benutzervariablen sind die *Aliase*.

5.2.1 Benutzervariablen

Benutzervariablen werden meistens in Skripten verwendet. Die Variablennamen dürfen aus Buchstaben, Ziffern und dem Unterstrich bestehen. Dabei darf das erste Zeichen keine Ziffer sein.

Den Variablen wird ein Wert durch folgende Sequenz zugewiesen.
`VARIABLENNAME=WERT`

Um eine Variable zu löschen, reicht es ihr einen Null-Wert zu geben.
`VARIABLENNAME=`

Die Variablen können nur von dem erzeugenden Prozess und seinen Kinderprozessen verwendet werden. Es sind lokale Variablen.

5.2.2 export

Um eine Variable überall zugängig zu machen, wird das bash-Kommando[2] `export` verwendet.

```
export [OPTIONEN] [NAME[=WERT]]
```

`export` ohne die Angabe von Optionen zeigt eine Liste aller exportierten Variablen.

Optionen	
-f	Die Variable wird als Funktion interpretiert
-n	Löschen einer Variablen aus der Liste
-p	Liste aller exportierten Variablen (standard)

Beispiel
Folgende Kommandos erstellen eine Variable und machen sie global zugänglich.
`ILove=Linux; export ILove`

[2] built-in command: Befehl, der direkt in die Shell eingebaut ist und nicht als eigenständiges Programm vorliegt.

Name	Bedeutung
?	Fehlercode des letzten ausgeführten Kommandos (0 für “OK”)
CDPATH	Suchpfad für das `cd`-Kommando
EDITOR	Name des Editors, den man verwenden möchte
EUID	Effektive Benutzerkennung (ro)
FCEDIT	Editor für den Befehl `fc`
HISTFILE	Name und Pfad für die History-Datei
HISTFILESIZE	Max. Größe der History-Datei
HISTSIZE	Max. Anzahl von Befehlen zur Speicherung in der History-Datei
HOME	Heimatverzeichnis des Benutzers
HOSTNAME	Name des Rechners
INPUTRC	Konfigurationsdatei für Kommandozeileneditor
PATH	Suchpfad für Kommmandos
PS1	Aktueller Prompt
PS2	Prompt für die Eingabe fehlender Parameter
PWD	Name des aktuellen Verzeichnisses
RANDOM	Zufallszahl
SECONDS	Sekunden nach Starten der Shell
SHELL	Name der benutzten Login-Shell
TERM	Art des Linux-Terminals
UID	Benutzerkennung (ro)
USER	Login-Name des Users

Tabelle 5.2: Auszug aus den Umgebungsvariablen

5.2.3 Umgebungsvariablen

Aussehen und Funktionsweise der Shell werden durch die Umgebungsvariablen festgelegt. Die Festlegung der Umgebungsvariablen erfolgt in verschiedenen Dateien, die in `/etc` oder im Heimatverzeichnis `~` liegen. So findet man z. B. die Umgebungsvariablen der `bash` in der Datei `/etc/profile` wieder. Einen Auszug von Umgebungsvariablen gibt Tabelle 5.2 wieder. Ausführlichere Informationen über die Umgebungsvariablen liefert das Kommando `man bash`.

Beispiel
Schauen wir uns als Beispiel die Variable `PWD` an, die den Namen des aktuellen Verzeichnisses enthält.

```
tapico@defiant:~/test > echo $PWD
/home/tapico/test
```

Der Variablen `PWD` kann ein neuer Wert zugewiesen werden.

```
tapico@defiant:~/test > PWD=/home/tapico
tapico@defiant:~/ > echo $PWD
/home/tapico
tapico@defiant:~/ > pwd
/home/tapico/test
tapico@defiant:~/ > echo $PS1
\u@\h:\w >
```

Nach der Änderung der Umgebungsvariable `PWD` ändert sich auch das Aussehen des Prompts, da dieser die Umgebungsvariable nutzt. Wie aber der Befehl `pwd` zeigt, hat sich das aktuelle Verzeichnis nicht geändert.

5.2.4 printenv

Um sich die angelegten Umgebungsvariablen ansehen zu können, wird der Befehl `printenv` verwendet.

```
printenv [VARIABLE]
```

Wird keine spezielle Variable angegeben, so zeigt `printenv` eine Liste aller Variablen an.

5.2.5 env

`env` startet ein Kommando mit anderen Umgebungsvariablen.

```
env [OPTIONEN] [VARIABLE=WERT] [BEFEHL [ARG1] ...]
```

Wird kein `BEFEHL` angegeben, so listet `env` die resultierende Umgebung auf. Es arbeitet dann wie `printenv` (5.2.4).

5.2.6 Suchpfad

Die Umgebungsvariable `PATH` enthält eine Liste von Verzeichnissen, die nach einem Kommando durchsucht werden. Die einzelnen Verzeichnisnamen werden durch einen Doppelpunkt '`:`' getrennt. Im Gegensatz zu DOS wird das aktuelle Verzeichnis `.` nicht immer durchsucht, sondern muß auch in der `PATH`-Variable angegeben werden.

```
tapico@defiant:~/ > echo $PATH
/bin:/usr/bin:/usr/local/bin:.
```

Im Normalfall ist das aktuelle Verzeichnis sogar das letzte in der Suchreihenfolge. Das kann zu Problemen führen. So wird beim Ausprobieren von Skripten z. B. sehr gerne der Name `test` als Dateiname verwendet. `test` ist aber ein Programm im Verzeichnis `/usr/bin`. Daher wird beim Aufruf von `test` nicht das Skript im aktuellen Verzeichnis, sondern das Programm in `/usr/bin` gestartet, weil es dort zuerst gefunden wurde.

Die Variable `PATH` wird in der Datei `/etc/profile` definiert. Um die Datei für sich selber zu ändern, sollte die Änderung in den Datei `~/.bash_profile` , `~/.profile` oder `~/.bashrc` erfolgen. Welche Datei als Beste dafür geeignet ist, ist von Distribution zu Distribution unterschiedlich.
Bsp.: `PATH=$PATH:$HOME/bin`

5.2.7 Prompt

Der Prompt steht immer am Anfang einer Kommandozeile. Er zeigt an, daß die Shell auf eine Eingabe wartet. Das Aussehen des Prompts wird durch die Umgebungsvariable `PS1` festgelegt.

Beispiel

```
tapico@defiant:~ > echo $PS1
\u@\h:\w >
```

`\u`, `\h` und `\w` sind besondere Zeichen, die ausführliche Informationen enthalten. Für eine Liste der speziellen Zeichen siehe Tabelle 5.3.

Wie der Name `PS1` schon vermuten läßt, existiert noch eine Umgebungsvariable `PS2`. Diese steuert das Aussehen des zweiten Prompts. Dieser Prompt taucht immer dann auf, wenn Sie einen Befehl noch nicht vollständig eingegeben und doch die Eingabetaste gedrückt haben. Er zeigt an, daß noch etwas fehlt.

Im folgenden Beispiel fehlt das schließende Anführungszeichen.

```
ole@enterprise:~> echo $PS2
>
ole@enterprise:~> PS2="ergänzen >"
ole@enterprise:~> echo $PS2
ergänzen >
```

Zeichen		Bedeutung
`\a`	*alert*	Piepton
`\d`	*date*	Datum Format: Wochentag Monat Tag
`\e`	*escape*	ESC-Zeichen
`\h`	*host*	Hostname bis zum ersten `.`
`\H`	*host*	Der ganze Hostname
`\n`	*newline*	Neue Zeile
`\r`	*return*	Wagenrücklauf
`\s`	*shell*	Name der Shell
`\t`	*time*	Zeit Format: 24 Stunden HH:MM:SS
`\T`	*time*	Zeit Format: 12 Stunden HH:MM:SS
`\@`		Zeit Format: 12 Stunden a.m./p.m.
`\u`	*user*	Username des aktuellen Benutzers
`\v`	*version*	Version der `bash`
`\V`	*version*	Ausgabenummer der `bash` (Version + Patchstufe)
`\w`	*working*	Aktuelle Arbeitsverzeichnis
`\W`		Der Basisname des aktuellen Verzeichnisses
`\!`		Die Nummer in der History für den aktuellen Befehl
`\#`		Die Nummer des Kommandos für den aktuellen Befehl
`\$`		Wenn die effective UID 0 ist, dann #, sonst $
`\nnn`		Zeichen mit dem oktalen ASCII-Code `nnn`
`\\`		Backslash
`\[`		Beginn einer Folge von nichtdruckbaren Zeichen
`\]`		Ende einer Folge von nichtdruckbaren Zeihen

Tabelle 5.3: Spezielle Zeichen für den Prompt. Die kursiven Worte sind Eselsbrücken zum Merken

```
ole@enterprise:~> echo "Hallo
ergänzen >Welt "
Hallo
Welt
ole@enterprise:~> echo $PS2
ergänzen >
```

5.2.8 alias

Für häufig wiederkehrende Befehle kann ein Alias angelegt werden. Dieses Alias ist vergleichbar mit einem kleinen Skript, wird aber wesentlich schneller ausgeführt. Um z. B. für den Befehl `cd ..` den Alias `..` zu setzen, wird folgendes Kommando eingegeben.

```
alias ..="cd .."
```

Die allgemeine Syntax für den Befehl `alias` lautet:

```
alias [NAME[=KOMMANDO]]
```

Dabei gibt das Kommando `alias` unter Angabe des Aliasnamen das dazugehörige Kommando aus. Der Aufruf ohne Parameter gibt eine Liste aller definierten Aliase mit ihren Kommandosequenzen aus.

Der Befehl `alias` sowie der Gegenbefehl `unalias` sind fester Bestandteil der Bash und direkt eingebaut.

5.2.9 unalias

Um die ganze Alias-Geschichte rückgängig zu machen, wird der Befehl `unalias` verwendet. Damit kann der normale Benutzer Aliase wieder entfernen, die der Systemverwalter für alle Benutzer durch Eintragen in die `/etc/.profile` angelegt hat.

```
unalias [-a] [NAMENSLISTE]
```

Wenn Sie den Schalter `-a` verwenden, arbeitet `unalias` sehr gründlich: Alle Alias-Definitionen werden gelöscht.

5.3 History-Liste

Die History-Liste umfaßt die letzten eingegeben Kommandos (*events*). Diese Liste kann dazu benutzt werden um alte Befehle wieder zu benutzen, sie zu bearbeiten oder sie zu analysieren. Dabei wird jedes Element von der Shell mit einer Nummer versehen.

Beim Starten der Shell wird die History-Liste aus der Datei `.bash_history` im Heimatverzeichnis erstellt. Die Umgebungsvariable `HISTFILE` enthält den Namen dieser Datei. Die Größe der Datei in Zeilen wird durch `HISTFILESIZE` bestimmt. Beim Ausloggen wird eine Anzahl von Zeilen, die in der Variablen `HISTSIZE` definiert worden ist, an die Datei `.bash_history` angehängt oder die komplette Datei wird durch diese ersetzt, je nachdem ob das `histappend`-Attribut der Shell gesetzt ist oder nicht. Mit den Shellvariablen `HISTCONTROL` und `HISTIGNORE` kann in Abhängigkeit von im Befehl enthaltenen Zeichen bestimmt werden, welcher Befehl in der History gespeichert wird und welcher nicht.

5.3.1 history

Das `bash`-Kommando `history` zeigt den Inhalt der History-Liste an.

```
history [OPTIONEN] [ZAHL]
```

Ohne Angabe von Parametern wird die gesamte History-Liste angezeigt. Durch Angabe einer Zahl werden die letzten `ZAHL` Zeilen der History-Liste angezeigt.

Optionen	
`-c`	Löschen der History-Liste

5.3.2 fc

Wie auch `history` ist `fc` ein bash-Kommando zur Arbeit mit der History-Liste. Es kann zur Bearbeitung und Wiederbenutzung von Kommandos verwendet werden.

```
fc -l [ANFANG] [ENDE]
fc -s [NUMMER]
```

Der Schalter `-f` in Kombination mit zwei Zahlen zeigt die Zeilen von `ANFANG` bis `ENDE` an. `ANFANG` und `ENDE` können auch Zeichenketten sein. Dann werden alle Zeichen vom ersten Auftauchen von `ANFANG` bis zum ersten Auftauchen von `ENDE` angezeigt.

Optionen	
`-l`	Zeigt ohne Parameter die letzten 16 Kommandos an.
`-s`	Führt ohne Parameter das letzte Kommando aus. Ansonsten das Kommando mit der angegebenen Nummer.
`-e`	Ermöglicht das Bearbeiten der History-Liste

Standardmäßig verwendet `fc` den Editor `vi` zum Bearbeiten. Durch setzen der Umgebungsvariablen `FCEDIT` kann auch ein anderer Editor dafür eingesetzt werden. So setzt

```
tapico@defiant:~> FCEDIT=/usr/bin/joe
tapico@defiant:~> export FCEDIT
```

den Editor `joe` als Standardeditor ein.

5.4 Umleitungen

5.4.1 Umleitung der Eingabe und Ausgabe

Standardmäßig kommen die Eingaben von der Tastatur und die Ausgabe des Programms und seiner Fehlermeldungen landen auf dem Bildschirm. Diese Ausgaben können aber in Dateien oder Geräte umgeleitet werden.

Eingabe Mit dem Zeichen < liest das Programm nicht von der Standardeingabe sondern holt seine Daten aus der angegeben Datei.
`cat < meinText`

Ausgabe Mit dem Zeichen > oder `1>` wird die Ausgabe des Programm in die angegebene Datei umgelenkt.
`cat * > alleDateien`
Dabei wird jedesmal eine neue Datei erstellt und der Inhalt der alten Datei gelöscht. Soll die Ausgabe aber an eine Datei angehängt werden, so verwendet man >>.
`echo $UID $USER >> logdatei`

Fehlerausgabe Mit `2>` wird die Ausgabe der Fehlermeldung umgeleitet. Oft stören diese, wenn sie nur melden, daß keine Zugriffsrechte auf ein Verzeichnis bestehen.
`ls -Rl 2> /dev/null`

Beide Ausgaben Um beide Ausgaben (Fehler und Programm) gemeinsam in eine Datei zu leiten, wird `&>` verwendet.
`cat *.log &> alllogs.txt`
oder
`cat *.log >& alllogs.txt`
Es geht aber auch
`cat *.log > alllogs.txt 2>&1`

5.4.2 Pipelines

Piplines ermöglichen es, die Ausgabe eines Kommandos direkt an die Eingabe eines anderen Kommandos zu senden. Realisiert wird dies durch ein zwischengeschaltetes Pipe-Zeichen `|`.
`ls -Al | less`

5.4.3 tee

Das Kommando `tee` sendet gleichzeitig die Ausgabe an eine Datei und an die Standardausgabe, die der Bildschirm oder eine weitere Pipeline sein kann.

`tee [OPTIONEN] DATEILISTE`

Optionen	
-a	Die Daten werden an eine vorhandene Datei angehängt

Beispiele
Diese Kommandosequenz sendet den Inhalt des Verzeichnisses in die Datei `myDir` und auf den Bildschirm.
`ls -al | tee myDir`

Hier wird der Inhalt der Datei `amAngelhaken` an die Datei `angeln.txt` angefügt und gleichzeitig mit `less` angezeigt.
`cat amAngelhaken | tee -a angeln.txt | less`

5.4.4 xargs

Der Befehl `xargs` ermöglicht die Weitergabe von Argumenten von einem Kommando zu einem anderen.

```
xargs KOMMANDO
```

Dabei ordnet `xargs` die übergebenen Daten als durch Leerzeichen getrennte Reihe von Worten an.

Beispiele
Dieser Befehl ist eine gutes Beispiel für die Verarbeitung der übergebenen Daten.

```
ls -l | xargs echo
```

Dieses Beispiel faßt die in der Datei `liste` aufgeführten Dateien zu einer Datei `myFiles` zusammen und gibt sie gleichzeit mit `less` auf dem Bildschirm aus.

```
cat liste | xargs cat | tee myFiles | less
```

Oder wollten Sie nicht schon mal wissen, was eigentlich die Programme im Verzeichnis `/bin` so machen. Der Befehl `whatis` (6.3.3) gibt eine Kurzbeschreibung aus.

```
ls /bin | xargs whatis | less
```

....

Notizen:

....

V	Arbeitsblatt Shell 2	5.1

Sollte eine Aufgabe zu einer Fehlermeldung führen, kann das von mir gewollt sein! Prüfen Sie aber dennoch, ob Sie keinen Tippfehler gemacht haben, und ob die Voraussetzungen wie in der Aufgabenstellung gegeben sind. Auch sollten keine Verzeichniswechsel ausgeführt werden, wenn dies nicht ausdrücklich in der Aufgabe verlangt wird! Notieren Sie die Ergebnisse auf einem seperaten Zettel.

Vergewissern Sie sich vor der Bearbeitung der Aufgaben, daß die Korn-Shell installiert wurde.

1 Loggen Sie sich als Walter auf der Konsole 1 ein.

2 Führen Sie das Kommando `rm -rf *` aus.

3 Ermitteln Sie den Namen der aktuellen Shell.

4 Starten Sie jetzt die Korn-Shell.

5 Wiederholen Sie den Befehl aus Aufgabe 3. Welches Ergebnis erhalten Sie diesmal?

6 Kehren Sie zur ursprünglichen Shell zurück.

7 Ändern Sie für Ihren Benutzer die Standardshell auf die Korn-Shell.

8 Loggen Sie sich aus.

9 Loggen Sie sich wieder ein.

10 Wiederholen Sie den Befehl aus Aufgabe 3. Welches Ergebnis erhalten Sie diesmal?

11 Machen Sie die Shell-Änderung wieder rückgängig.

12 Loggen Sie sich wieder aus und dann wieder ein.

13 Lassen Sie sich das Tagesdatum in der in Deutschland gebräuchlichen Form (Tag.Monat.Jahr) anzeigen.

14 Welche Tastenkombination unterbricht ein laufendes Programm?

15 Was für ein Wochentag war der 18. März 1970?

16 Was war das besondere am September 1752?

17 Speichern Sie den Kalender für das Jahr 2001 in der in Deutschland gebräuchlichen Form in der Datei `Kalender`.

18 Wie spät ist es auf dem Linux-Rechner?

19 Mit welchem Terminaltyp arbeiten Sie gerade?

20 Erstellen Sie eine persönliche `inputrc`-Datei. Belegen Sie die Tasten
`<F1>` mit dem Befehl für den aktuellen Jahreskalender
`<F2>` mit dem Befehl für die Uhrzeit und
`<F3>` mit dem Befehl für den Editor `vi`.

21 Testen Sie die neuen Tastenbelegungen.

22 Lassen Sie sich die Datei `/etc/DIR_COLORS` anzeigen.

23 Kopieren Sie die Datei `/etc/DIR_COLORS` als `.dir_colors` in Ihr Heimatverzeichnis.

24 Ändern Sie den `ls`-Farbcode für Verzeichnis auf rot und unterstrichen.

25 Testen Sie die Änderung der Farbcodes.

26 Lassen Sie sich die Umgebungsvariable für den aktuellen Prompt anzeigen.

V	Arbeitsblatt Shell 2	5.2

27 Ändern Sie den Prompt zu: “BENUTZERNAME `@` HOSTNAME SHELL SHELLVERSION+PATCHLEVEL am DATUM : VERZEICHNIS >”
Bsp.: `tapico@defiant bash 2.03.0 am Fri Dec 29: ~ >`

28 Loggen Sie sich aus und wieder ein. Wie sieht der Prompt nun aus?

29 Richten Sie nun die Promptänderung so ein, daß Sie bei jedem Einloggen gilt.

30 Lassen Sie sich alle Umgebungsvariablen anzeigen.

31 Welche Bedeutung hat die Variable `_`.

32 Geben Sie die Liste der Umgebungsvariablen zusammen mit dem Datum und dem Text “Umgebungsvariablen” in die Datei mit dem Namen `variablenJJ-MM-TT.txt` aus. Dabei steht `JJ-MM-TT` für das aktuelle Datum.

33 Speichern Sie die Dateinamen aus dem Verzeichnis `/usr/share/doc/packages/cron` in der Datei `dateiliste`.

34 Kopieren Sie alle Dateien aus `/usr/share/doc/packages/cron` und `/usr/share/doc/packages/yast2` in Ihr Heimatverzeichnis.

35 Fassen Sie den Inhalt der Dateien, die Sie aus `/usr/share/doc/packages/cron` kopiert haben und die sich nun in Ihrem Heimatverzeichnis befinden, zu einer Datei `alleAnleitungen.txt` zusammen.

36 Wechseln Sie zur Konsole 2.

37 Loggen Sie sich als **root** ein.

38 Legen Sie das Verzeichnis `/kurs/bin` an und wechseln Sie in das Verzeichnis.

39 Erstellen Sie eine Datei mit dem Namen `hurra` und dem Befehl
`echo -e "Hurra\n"`
als Inhalt.

40 Geben Sie den Befehl `chmod 755 hurra` ein. Die Datei wird damit ausführbar.

41 Wechseln Sie wieder zur Konsole 1.

42 Starten Sie das Programm `hurra`. Klappt das?

43 Starten Sie nun das Programm `hurra` mit Angabe des Pfades `/kurs/bin`. Klappt das?

44 Überprüfen Sie, ob das Verzeichnis `/kurs/bin` in der Pfadliste enthalten ist.

45 Fügen Sie das Verzeichnis `/kurs/bin` der Verzeichnisliste hinzu.

46 Starten Sie nun wieder das Programm `hurra.` Klappt das?

47 Legen Sie das Alias `adir` mit `ls -al | less` an.

48 Legen Sie das Alias `werbinich` an. Es soll den Namen ausgeben, mit dem Sie sich eingeloggt haben.

49 Testen Sie die Aliase.

50 Loggen Sie sich aus und wieder ein.

V	Arbeitsblatt Shell 2	5.3

51 Testen Sie wieder die Aliase.

52 Legen Sie in Ihrem Heimatverzeichnis eine Datei `.alias` ein und tragen Sie dort die beiden Aliase ein.

53 Loggen Sie sich aus und wieder ein.

54 Testen Sie wieder die Aliase.

55 Legen Sie ein Alias `myenv` in der Datei `.alias` ein. Das Alias soll für den Befehl aus Aufgabe 32 stehen.

56 Führen Sie die Datei `.alias` aus.

57 Testen Sie den neuen Alias.

58 Führen Sie wieder den Befehl `hurra` aus. Klappt das? Wenn nicht, warum nicht?

Notizen:

Notizen:

Kapitel 6

Hilfe und Dokumentation

Ein Mathematiker, ein Physiker und ein Computeruser werden getrennt von einander eingeschlossen. Jeder erhält zwei Glaskugeln. Nach einer Stunde schaut man was diese Leute damit machen. Der Mathematiker sitzt dort und berechnet das Volumen und die Oberfläche der Kugeln. Der Physiker hält die Kugeln gegen das Licht und berechnet Brechzahl und Absorptionskoeffizient. Als letztes schaut man beim Computerbenutzer herein und stellt fest, daß eine Kugel weg ist und das Fenster zerbrochen. Auf die Frage, was denn passiert sei, zuckt der User nur mit den Achseln und sagt: "Ich hab' nix gemacht!..."

Anonymous

Um mit dem Linux-System arbeiten zu können, muß man die Befehle des Systems kennen. Umfangreiche Hilfefunktionen helfen dabei.

6.1 Lokale Dokumentation

Es gibt eine Reihe von lokalen Dokumentationen in einem Linux-System. Tabelle 6.1 zeigt die Typen und ihre Position im Dateisystem.

6.1.1 Manual-Pages

Die Manual-Pages, auch oft Man-Pages genannt, sind eine extensive Online-Hilfe zu fast jedem Tool, Systemaufruf oder Kommando des Systems. Sie sind nicht gerade benutzerfreundlich, da sie mehr als Referenz für die Syntax gedacht sind und nicht als Online-Tutorial.

Für die Verwendung der Manual-Pages wird das Tool `man` (6.1.2) verwendet. Dieses Tool sucht den passenden Hilfetext heraus und stellte ihn durch einen Textbetrachter dar. In den meisten Fällen ist `less` der Pager.

Die Hilfetexte sind in mehrere Teile aufgeteilt, wie in Tabelle 6.2 zu sehen. Es muß aber nicht unbedingt

Dokumentation	Position	SuSE
Manual-Pages	`/usr/man`	`/usr/share/man`
info	`/usr/info`	`/usr/share/info`
HOWTOs	`/usr/doc/HOWTO`	`/usr/share/doc/howto`
Frequently Ask Questions (FAQ)	`/usr/doc/FAQ`	nicht mehr vorhanden
Programm Dokumentationen	`/usr/doc/PROGRAMMNAME`	`/usr/share/doc/packages/PAKETNAME`

Tabelle 6.1: Die lokalen Dokumentationen und ihre Position

Abschnitt	Beschreibung
NAME	Name und kurze Beschreibung des Kommandos
SYNOPSIS	Anwendung und Syntax des Kommandos
DESCRIPTION	Beschreibung des Kommandos und seiner Optionen
OPTIONS	Liste aller Optionen mit Kurzbeschreibung
SEE ALSO	Manual-Pages zu verwandten Inhalten
DIAGNOSTICS	Beschreibung von Fehlermeldungen
FILES	Liste der Dateien, die von dem Kommando verwendet werden
BUGS	Bekannte Probleme
HISTORY	Meilensteine der Programmentwicklung
AUTHOR	Autor des Programms und Mitarbeiter

Tabelle 6.2: Die Abschnitte in den Manual-Pages

Verzeichnis	Beschreibung
`man1`	Ausführbare Programm oder Shell-Kommandos
`man2`	Systemaufrufe (Kernelfunktionen)
`man3`	Bibliotheksaufrufe (Funktionen mit Systembibliotheken)
`man4`	Spezielle Dateien (meistens in /dev)
`man5`	Dateiformate und Konventionen
`man6`	Spiele
`man7`	Makropakete und Konventionen
`man8`	Systemadministrationskommandos
`man9`	Kernelroutinen (kein Standard)

Tabelle 6.3: Die neun “Kapitel” der Manual-Pages

jeder Abschnitt in dem Text vorkommen.

Die Dateien für die Manual-Pages können in verschiedenen Pfaden liegen. Generell ist dies der Pfad `/usr/man`. Bei der SuSE-Distribution liegen die Hilfetexte im Verzeichnis `/usr/share/man`. Die Umgebungsvariable `MANPATH` enthält die Pfade zu den Hilfetexten.

In diesem Verzeichnis existieren mehrere Unterverzeichnisse für die neun “Kapitel” der Manual-Pages. Eine Übersicht über die Kapitel liefert die Tabelle 6.3. Wenn nun nach einem Thema gesucht wird, wird die erste Seite angezeigt, die gefunden wird. Dabei werden die Kapitel in der Reihenfolge
1, 8, 2, 3, 4, 5, 6, 7, 9
durchsucht.

6.1.2 man

Ruft die Online-Dokumentation (Manual-Pages) zu einem Befehl auf.

```
man [OPTIONEN] [SEKTION] [THEMA]
```

Dabei wird die erste passende Seite zu dem Thema angezeigt.

Optionen

`-a`	Zeigt alle passenden Seiten hintereinander an
`-f`	Zeigt eine Kurzbeschreibung zu den passenden Seiten an.
`-k`	Stichwortsuche in der Hilfe nach den Kurzbeschreibungen
`-K`	Volltextsuche in der Hilfe (Achtung: Nicht in jedem man-Tool implantiert)

Die Konfigurationsdatei für `man` ist `/etc/man.config` bzw. `/etc/manpath.config`. Die Umgebungsvariable `PAGER` enthält den Namen des Textbetrachters, den `man` verwendet.

6.1.3 TexInfo

TexInfo[1], die GNU Hypertext Dokumentation, ist einfacher zu benutzen als die Manual-Pages und oft auch ausführlicher. Für die Darstellung der Seiten wird das Kommando `info` verwendet.

6.1.4 info

Ruft die Online-Dokumentation (TexInfo-Dateien) eines Befehls auf.

```
info [KOMMANDO]
```

Bei Linux befinden sich die TexInfo-Dateien in den Verzeichnissen `/usr/local/info`, `/usr/share/info` und `/usr/info`.

Steuerung	
`<q>`	beendet das Programm
`<STRG>+<h>`	ruft die Hilfe auf
`<n>`	zum nächsten Knoten (Menüpunkt)
`<p>`	zum vorherigen Knoten (Menüpunkt)
`<u>`	zum aufwärtigen Knoten (Menüpunkt)
`<l>`	zum letzten dargestellten Knoten (auch die Hilfe ist ein Knoten)
`<m>`	Menüpunkt mit Namen aufrufen
`<1> ... <9>`	Menüpunkt mit dieser Nummer folgen

6.1.5 help

Zeigt hilfreiche Informationen über eingebaute Befehle an.

```
help [KOMMANDO]
```

6.1.6 HOWTOs

Die Linux HOWTOs sind ausführliche Dokumente, die die Bearbeitung bestimmter Aufgaben in Linux beschreiben. Die HOWTO-Dokumente behandeln meist komplexere Aufgaben und sind daher länger. Für einfachere Aufgaben wie `LILO` oder Drucken gibt es kürzere Texte, die *mini-HOWTOs*.

Die HOWTO-Texte befinden sich meistens in komprimierter Form im Verzeichnis `/usr/doc/HOWTO`, `/usr/doc/howto` bzw. `/usr/share/doc/howto`. Bezugsquelle für die HOWTOs sind
`ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO`
`http://sunsite.unc.edu/LDP/HOWTO/`
`http://www.linuxdoc.org/HOWTO/`
bzw. einer Ihrer Spiegelserver.

6.1.7 FAQ

Die FAQs (Frequently Asked Questions) sind eine Sammlung von Dokumenten im Zusammenhang mit den HOWTOs. Sie sind in der Form Frage-Antwort aufgebaut, wobei die Fragen gesammelt werden, die die Benutzer am häufigsten stellen.

Die FAQs sind im HTML-, PostScript- und Text-Format vorhanden und befinden sich im Verzeichnis `/usr/doc/FAQ`. Die Online-Quelle für die neuesten FAQs ist
`http://www.linuxdoc.org/FAQ/`.

Seit SuSE 8.0 sind die FAQs kein Bestandteil mehr der Distribution. Das ist aber zu verschmerzen, da ihr Informationswert nicht sehr groß ist.

[1] Das Paket `texinfo` gehört nicht mehr zur Standardinstallation ab SuSE 8.0 und muß daher nachinstalliert werden.

6.1.8 Programmdokumentation

Dokumentationen über spezielle Programme finden Sie im Verzeichnis `/usr/doc` in einem Unterverzeichnis mit dem Namen des Programms. Bei der SuSE-Distribution finden Sie weitere Informationstexte zu den Programmen der Softwarepakete im Verzeichnis `/usr/share/doc/packages` und dessen Unterverzeichnissen. Die meisten Texte können mit `less` oder `more` gelesen werden. Andere Dokumentationen sind als LaTeX-DVI, PostScript, HMTL oder PDF zugänglich. Manche Programm legen aber auch bei Ihrer Installation eine Manual-Page an.

6.2 Internetquellen

Das Internet war und ist für Linux die Basis seiner Entwicklung. Der Quellcode für Linux ist übers Internet erhältlich. Desweiteren gibt es auch zahlreiche Dokumentationen an verschiedenen Orten. Die erste Quelle für Informationen zu Linux ist das **Linux Documentation Project** auf `http://metalab.unc.edu/LDP/`.

Auch gibt es durch dieses Projekt einige Bücher, die spezielle Themen abdecken. Diese können von `http://sunsite.unc.edu/LDP/` heruntergeladen werden. U. a. sind dies

- Goldt, Sven; Meer, Sven van der; Burkett, Scott; Welsh, Matt. *The Linux Programmers Guide*
- Greenfield, Larry. *The Linux User's Guide*
- Johnson, Michael K.. *The Linux Kernel Hacker's Guide*
- Kirch, Olaf. *The Network Administrator's Guide*
- Pomerantz, Ori. *The Linux Kernel Module Programming Guide*
- Rusling, David. *The Linux Kernel*
- Welsh, Matt. *Installation and Getting Started Guide*
- Welsh, Matt. *The Linux Documentation Project Manifesto*
- Wirzenius, Lars. *The Linux System Administrators Guide*

Ein weiteres Projekt für eine mehr benutzerfreundliche Dokumentation ist Linuxnewbie.org (`http://www.linuxnewbie.org/`).

Die Linuxgazette (`http://www.linuxgazette.com/`) ist ein monatlich erscheinendes Online-Magazin zum Thema Linux. Es ist Mitglied des Linux Documentation Projects. Es gibt auch eine deutsche Ausgabe (`http://www.linuxgazette.de/`) in der einige Artikel der englischen Ausgabe ins Deutsche übersetzt wurden.

Besonders für SuSE-Nutzer ist die SuSE-Seite (`http://www.suse.de`) mit ihrer Supportdatenbank (`http://sdb.suse.de/sdb/de/html/`)zu empfehlen, da dort die gängigsten Probleme und ihre Lösungen beschrieben werden.

Natürlich soll auch die Seite `http://www.fibel.org` nicht vergessen werden, wo Sie dieses Skript herunterladen können. `;-)`

6.2.1 Newsgroups

Eine weitere Möglichkeit um an Informationen über Linux heranzukommen sind die Newsgroups. Allerdings sollten Sie zuerst alle anderen Möglichkeiten ausgeschöpft haben, bevor Sie eine Frage in der Newsgroup stellen. D. h. konsultieren Sie zuerst Manual-Pages, TexInfo, HOWTOs, FAQs, Programm- und Paketdokumentationen und auch die FAQs, die zu den Newsgroups gehören.

Für das Lesen der Newsgroups benötigen Sie einen Newsreader wie er in Netscape oder Mozilla integriert ist oder ein separates Programm wie z. B. KNode (2.4.6). Die Suchmaschine Google (`http://www.google.de`) hat das Newsgrouparchiv von `http://www.deja.com` übernommen. Sie ist eine gute Alternative um die Newsgroups mit dem Browser durchsuchen und lesen zu können.

- **comp.os.linux.announce** ist eine moderierte Newsgroup, die sich mit den neuesten Informationen zu Software-Updates, neuen Ports, Benutzergruppentreffen und kommerziellen Produkten beschäftigt.
- **comp.os.linux.answers** enthält alle FAQs, HOWTOs und andere wichtige Dokumentationen. Die HOWTOs werden immer am Anfang des Monats gepostet.
- **comp.os.linux.advocacy** ist das Diskussionsforum für juristischen Fragen zu Linux.
- **comp.os.linux.alpha** legt seinen Themenschwerpunkt auf Fragen zu Linux für Alpha-Systeme.
- **comp.os.linux.development.apps** beschäftig sich mit der Entwicklung von Anwendungsprogrammen für Linux.
- **comp.os.linux.development.system** behandelt den Bereich der Entwicklung von Linux.
- **comp.os.linux.embedded** ist eine Newsgroup für Embedded-Geräte unter Linux.
- **comp.os.linux.hardware** enthält Fragen zur Zusammenarbeit von Linux und der Hardware.
- **comp.os.linux.m68k** legt seinen Themenschwerpunkt auf Fragen zu Linux für Apple-Prozessoren.
- **comp.os.linux.networking** kümmert sich um den Netzwerkbereich von Linux.
- **comp.os.linux.portable** deckt den Bereich Linux für Notebooks ab.
- **comp.os.linux.powerpc** legt seinen Themenschwerpunkt auf Fragen zu Linux für den PowerPC.
- **comp.os.linux.security** ist die richtige Newsgroup, wenn es um Sicherheitsfragen zu Linux geht.
- **comp.os.linux.setup** beschäftigt sich mit der Administration und Konfiguration von Linux.
- **comp.os.linux.x** ist der graphischen Oberfläche X-Window gewidmet.
- **comp.os.linux.misc** deckt die Themen ab, die nicht zu den anderen Newsgroups passen.

6.2.2 KNode

Unter KDE ist das Programm KNode zum Arbeiten mit den Newsgroups vorgesehen.

Wenn Sie KNode zum ersten Mal starten, erscheint gleich das Einstellungsfenster. Sie können es aber auch nachträglich unter *Einstellungen/KNode einrichten...* wieder aufrufen.

Als erstes tragen Sie unter Identität die persönlichen Angaben ein, wie Sie es auch für die eMail machen.

Unter *Zugang/News* tragen Sie die zu verwendenden Newsserver ein. Diese halten die Artikel der einzelnen Newsgroups vor und sorgen auch für den Abgleich der Server untereinander. Entweder haben Sie von Ihrem Provider einen Newsserver benannt bekommen oder Sie müssen sich einen freien Newsserver suchen. Unter `http://newssearch.pilum.net` finden Sie eine Suchmaschine für freie Newsserver. Geben Sie einfach den Namen der Newsgroup an und pilum.net ermittelt die passenden Newsserver dafür. Über den Button *Neu* bekommen Sie jetzt ein Fenster für den neuen Newsserver. Tragen Sie einen Namen für die Anzeige und den Namen des Newsservers ein. Unter dem Reiter *Identität* können Sie individuell für jeden Newsserver sich eine Identität zulegen.

Weitere Einträge brauchen Sie zuerst nicht. Im linken Rahmen von Knode ist nun der eingetragene Newsserver aufgetaucht. Sie müssen nun eine Newsgroup von dem Server abonnieren. Dazu klicken Sie mit der rechten Maustaste auf den Servernamen und wählen den Punkt *Newsgruppen abonnieren...* aus. Nun holt KNode vom Server die vorhandenen Gruppen. Das kann bei manchen Servern sehr lange dauern. Die Gruppen werden Ihrer Hierachie gemäß in Baumform angezeigt.

Wählen Sie nun die gewünschten Gruppen (z. B. alle Linuxgruppen) aus, indem Sie auf die schwarz umrandeten Kästchen klicken. Bestätigen Sie die Auswahl mit *OK*. Die Gruppen sind nun unter dem Newsserversymbol erschienen. Wenn Sie nun auf eine der Gruppen klicken, dann zeigt das linke Fenster die Zahl der vorhandenen Artikel an, während eine Liste der Artikel im rechten oberen Rahmen erscheint. Klicken Sie dort auf einen Eintrag, so erscheint der gewählte Artikel im rechten unteren Rahmen.

6.2.3 Mailinglisten

Neben den Newsgroups gibt es auch einige Mailinglisten zum Thema Linux.

comp.os.linux.announce

Um einen Spiegel dieses Forums zu bekommen, senden Sie eine eMail an
`Linux-Announce-Request@NEWS-DIGESTS.MIT.EDU`
mit dem Wort **subscribe** im Nachrichtenrumpf.

majordomo@vger.rutgers.edu

Hier werden einige Mailinglisten zum Thema Linux gehostet. Um eine Liste aller Mailinglisten zu bekommen, senden Sie einfach eine Nachricht an diese Adresse mit dem Wort **list** im Nachrichtenrumpf. Steht das Wort **help** im Nachrichtenrumpf, dann bekommen Sie eine eMail mit der Hilfe zu diesen Mailinglisten.

SuSE Mailinglisten

Auch SuSE bietet Mailinglisten an. Diese können Sie unter
`http://www.suse.de/de/private/support/mailinglists/index.html`
abonnieren. Stellen Sie sich aber auf einige hundert eMails am Tag ein, wenn Sie das tun.

6.3 Suchen nach Informationen

6.3.1 whereis

Das Kommando `whereis` lokalisiert die Programmdatei, Quellcodedatei und die Online-Hilfe (Manual-Pages) für ein Kommando.

```
whereis [OPTIONEN] KOMMANDO
```

Die Suche ist dabei auf eine limitierte Anzahl von bekannten Verzeichnissen beschränkt. Wenn die gesuchte Datei nicht gefunden wurde, wird nur der gesuchte Name zurückgegeben.

Optionen	
`-b`	Suche nach Programmdatei (binary)
`-s`	Suche nach Quellcodedatei (source file)
`-m`	Suche nach Online-Hilfe (manpages)
`-u`	Suche nach anderen Dateien (unusal entries)

Voreinstellung ist -bmsu zur Suche nach allen möglichen Dateien.

6.3.2 which

Das Kommando `which` zeigt an, welches Programm mit welchem Pfad bei Eingabe eines Kommandos gestartet wird.

```
which KOMMANDO
```

`which` durchsucht die Verzeichnisse, die in der Umgebungsvariablen `PATH` angegeben sind, und gibt den Pfad des ersten Fundorts des gesuchten Programms an.

6.3.3 whatis

Zeigt die Kurzbeschreibung der Online-Hilfe (Manual-Pages) an.

`whatis [OPTIONEN] THEMA`

Innerhalb jeder Manualseite ist eine Kurzbeschreibung vorhanden. `whatis` sucht Schlüsselwort in den Kurzbeschreibungen der Indexdatenbank. Falls es eine solche nicht im Manualpfad gibt, durchsucht es die `whatis`-Datenbank, die sich in `/usr/man/man1` befindet, nach dem Schlüsselwort.

`whatis` verhält sich wie `man -f`.

6.3.4 mandb

Erzeugt die von `whatis` (6.3.3) benutzte Indexdatenbank für die Manualpages.

`mandb [OPTIONEN] [PFAD]`

6.3.5 apropos

Der Befehl `apropos` ermöglich eine Stichwortsuche in der Kurzbeschreibung der Hilfe.

`apropos [OPTIONEN] STICHWORT`

Die Suche mit `apropos` ist identisch zu `man -k`.

6.3.6 type

Das Kommando `type` zeigt an, wie ein Begriff interpretiert wird, wenn er als Kommando benutzt wird.

`type [OPTIONEN] BEGRIFFE`

Optionen	
`-t`	Ausgabe eines einzelnen Wortes ('alias', 'keyword', 'function', 'builtin', 'file' oder ' ')
`-p`	Gibt die Datei aus, die ausgeführt wird. (Wird nur ausgegeben, wenn `-t` 'file' liefert.)
`-a`	Gibt die Datei aus, die ausgeführt wird. (Wird nur ausgegeben, wenn `-t` 'file', 'alias' oder 'function' liefert.)

Die Schalter -t, -p und -a können nicht zusammen verwendet werden.

6.4 Dokumentation und Support

Neben den reinen Kenntnissen über das Betriebssystem, die Software und die Hardware gehören für die Tätigkeit als Systemadministrator auch weiche Fähigkeiten (Soft-Skills) dazu. Zu diesen Soft-Skills gehört das Schreiben von Dokumentationen und der Benutzer-Support.

6.4.1 Dokumentationen schreiben

Eine der Aufgaben, denen sich die Leute am liebsten entziehen, ist die Dokumentation. Trotzdem ist gerade dieser Bereich der EDV besonders wichtig, weil hier Erfahrungen dauerhaft gespeichert werden können.

Systemübersicht Für jeden Rechner sollte eine Systemübersicht existieren. Diese sollte umfassen: Art des Festplattencontroller, Anzahl und Art der Festplatten, Partitionsübersicht, Hostname, IP-Adresse, Peripheriegeräte, BIOS-Einstellungen u. v. m.

Problemlösungen Wenn ein Problem aufgetreten ist, sollte man seine Lösung ausführlich dokumentieren. Wenn später der Fall wieder auftritt, braucht man nur nachzuschlagen.

Änderungen am System Jegliche Änderungen am System sollte auch festgehalten werden. Dies umfaßt die Änderungen an der Hardware, Konfigurationsdateien, Programmen u. s. w. Dies sollte ausreichen, um das System innerhalb kürzester Zeit wieder zu rekonstruieren.

Dokumentation für Benutzer Neben der Dokumentation für sich, den Administrator, sollten Sie auch in der Lage sein, Dokumentationen für die Benutzer zu schreiben. So können Sie z. B. erklären wie der Benutzer sich einloggt oder seine eMail abholen kann. Daneben sollten auch andere Administratoren in der Lage sein mit Hilfe der Dokumentation Ihr System zu warten[2].

Jetzt kommt es noch darauf an, welche Art der Dokumentation man wählen soll. Es gibt zwei Arten: die elektronische Dokumentation und die Dokumentation auf Papier.

Elektronische Dokumentation Die Vorteile der elektronischen Dokumentation liegen auf der Hand. Sie ist leicht auf dem neuesten Stand zu halten und leicht zu durchsuchen. Kommandoausgaben und Screenshots können leicht integriert werden. Außerdem ist sie schnell zu kopieren und kann vielen zugänglich gemacht werden. Der Nachteil ist allerdings, sie steht nur zur Verfügung wenn die Maschine läuft. Auch läßt sie sich schlecht herumtragen.

Dokumentation auf Papier Die alte Methode der Dokumentation auf Papier hat auch heute noch ihre Vorteile. Sie ist auch zugänglich wenn das System unten ist. Dank der modernen Textverarbeitung können auch Screenshots und Kommandoausgaben eingearbeitet werden. Vor allem kann man sie überall hinnehmen, wo man Sie braucht. Der Nachteil dieser Form der Dokumentation liegt in ihrer schlechten Wartbarkeit. Eine Papierdokumentation auf dem neuesten Stand zu halten ist aufwendig. Daneben kann diese Form der Dokumentation schnell zu Chaos führen. Das Entziffern der Notizen, gerade bei handschriftlichen Dokumentationen, ist auch ein nicht zu unterschätzender Nachteil.

6.4.2 Benutzer-Support

Der Benutzer-Support ist eine Aufgabe, die besondere Fähigkeiten verlangt. Neben dem fachlichen Wissen sind hier vor allem der Umgang mit Menschen und die Kommunikationsfähigkeit gefragt. Sie müssen in der Lage sein Informationen so an den Benutzer weiterzugeben, daß er sie verstehen und anwenden kann.

Obwohl der Benutzer-Support sehr aufwendig ist, sollten Sie dem Benutzer nie das Gefühl geben seine Anfrage wäre Zeitverschwendung. Diese Anfragen können auf viele Arten und Weisen erfolgen. Ihr Ziel sollte es aber immer sein den Benutzer produktiver werden zu lassen.

- **Zeit** Sie sollten alle Anfragen innerhalb eines kurzen Zeitrahmen beantworten. Wenn Sie keine Zeit haben, informieren Sie den Benutzer darüber und wann Sie wieder für Ihn Zeit haben.
- **Analyse des Problems** Bevor Sie ein Problem lösen können, müssen Sie es erstmal analysieren. Fragen Sie genau nach wann, wo und wie das Problem aufgetreten ist. Es könnte ja auch kein Fehler sondern eine Anfrage für eine neue Funktion sein.
- **Dokumentation** Wenn Sie das Problem erkannt haben, konsultieren Sie Ihre Dokumentation. Wenn Sie Glück haben, ist es dort mit der Lösung verzeichnet.
- **Telefonsupport** Hier kommt es darauf an die richtigen Fragen zu stellen. Aber Vorsicht! DAUs[3] können ihnen den letzten Nerv abverlangen.
- **eMail-Support** Bei schriftlichen Anfragen, wie z. B. per eMail, werden nicht immer alle wichtigen Einzelheiten aufgeführt. Fragen Sie also wieder nach!
- **Sprache** Beim Kontakt mit dem Endanwender vermeiden Sie bitte den EDV-Slang. Nicht jeder ist so ein Abkürzungslexikon wie Sie. Versuchen Sie dem Benutzer alles mit möglichst einfachen Worten zu erklären.

[2] Als Administrator unersetzlich zu sein hat nämlich nicht nur Vorteile.
[3] Dümmster Anzunehmender User

- **Allgemeines Wissen** Solange Dokumentationen keine sicherheitsrelevanten Bereiche betreffen, sollten Sie sie den Benutzern zur Verfügung stellen. Je mehr der Benutzer weiß, desto weniger wird er Sie fragen müssen.
- **Ende** Das Problem ist erst dann gelöst, wenn der Benutzer meint, es ist gelöst. Sollte sich die Problemlösung verzögern, dann informieren Sie den Benutzer darüber und wann er mit den Informationen rechnen kann. Sie sollten dem Benutzer das Gefühl geben ein Teil auf dem Weg zur Lösung zu sein.

Notizen:

Notizen:

VI	Arbeitsblatt Dokumentation und Hilfe	6.1

Sollte eine Aufgabe zu einer Fehlermeldung führen, kann das von mir gewollt sein! Prüfen Sie aber dennoch, ob Sie keinen Tippfehler gemacht haben, und ob die Voraussetzungen wie in der Aufgabenstellung gegeben sind. Auch sollten keine Verzeichniswechsel ausgeführt werden, wenn dies nicht ausdrücklich in der Aufgabe verlangt wird! Notieren Sie die Ergebnisse auf einem seperaten Zettel.

Teil 1

1 Loggen Sie sich als Walter auf der Konsole 1 ein.

2 Führen Sie den Befehl `rm -rf *` aus.

3 Lassen Sie sich die Online-Hilfe zum Befehl `man` anzeigen und informieren Sie sich über den Befehl.

4 Was sagt TexInfo über `man` aus?

5 Was ist Ihnen aufgefallen?

6 Suchen Sie nach dem Stichwort "man" in der Hilfe (Manualpages).

7 Stellen Sie fest was `man` ist?

8 Rufen Sie die Manual-Page des Befehls `which` auf.

9 Lassen Sie sich den Ort der Binärdatei zu `which` zeigen.

10 Um welchen Typ handelt es sich bei `which`?

11 Nutzen Sie die oberen Informationen, um die wahre Funktion von `which` zu ermitteln.

12 Sorgen Sie dafür, daß `which` seine ursprüngliche Aufgabe dauerhaft wieder erledigen kann. (Nutzen Sie dafür das Ergebnis aus Aufgabe 9.)

13 Was ist `dir` ?

14 Zu welchem Typ gehört `dir` ?

15 Was bewirkt der Parameter `-P` beim Befehl `cp` ?

16 Unter welche Gruppe fällt `cp` nach `info`?

17 Welche Befehle fallen noch in die Gruppe und was tun sie?

18 Was ist `passwd` und wo befindet es sich?

19 Zu welchem Zweck dient der Befehl `rman`?

....

Notizen:

....

VI	Arbeitsblatt Dokumentation und Hilfe	6.2

Teil 2

Für das Archivieren und platzsparende Packen von Dateien werden Packprogramme verwendet. Verwenden Sie die Hilfen des Linux-Betriebssystems um die folgenden Fragen zu beantworten.

1 Loggen Sie sich als **walter** ein oder wechseln Sie das Heimatverzeichnis von **walter**!

2 Führen Sie den Befehl `rm -rf *` aus.

3 Beschreiben Sie mit ein paar Sätzen jeweils die Funktion der folgenden Programme. (Anwendung, Anwendungsgebiet, Verhalten.)

1. `tar`
2. `gzip`
3. `compress`
4. `zcat`
5. `zip`

4 Erläutern Sie die Schalter `-d`, `-l`, `-r`, `-S`, `-t` und `-v` von `gzip`.

5 Kopieren Sie aus `/usr/share/man/man1` die Dateien `zip.1.gz`, `zipgrep.1.gz` und `zipinfo.1.gz` in Ihr Heimatverzeichnis.

6 Fassen Sie den Inhalt der Dateien in der neuen Datei `zip.all` zusammen.

7 Lassen Sie sich den Inhalt der Datei `zip.1.gz` mit `less` anzeigen.

8 Entpacken Sie die gepackten Dateien.

9 Packen Sie alle '`.1`'-Dateien zum Archiv `zip.zip` zusammen.

10 Packen Sie die Dateien noch einmal jeweils mit den Parametern `-1` und `-9` und vergleichen Sie die entstandenen Zip-Archive `zip1.zip` und `zip9.zip` miteinander.

11 Löschen Sie alle Dateien mit der Endung `.1`.

12 Welche Funktion hat das Programm `unzip`?

13 Lassen Sie sich Informationen über die Dateien im Zip-Archiv `zip.zip` anzeigen.

14 Entpacken Sie aus dem Archiv `zip.zip` die Datei `zipgrep.1`.

15 Entpacken Sie aus dem Archiv `zip1.zip` alle Dateien außer `zipgrep.1`.

16 Am 02.01.2001 wurde zum letzten Mal ihr Heimatverzeichnis gesichert. Wie sichern Sie nun alle neuen und geänderten Dateien in das Zip-Archiv `bup.zip`?

17 Wandeln Sie die Manpage für `zip` (`zip.1`) in eine HTML-Datei namens `zip.html` um.

18 Lassen Sie sich die HTML-Datei im Browser anzeigen.

19 Löschen Sie alle Zip-Archive im Heimatverzeichnis.

Kapitel 7

Textfilter

7.1 Ausgabe von ganzen Dateien

Um ganze Dateien auszugeben haben wir schon die Pager `less` (4.5.10) und `more` (4.5.9) sowie den Verbindungsbefehl `cat` (4.5.2) kennengelernt. Aber es gibt noch weitere Befehle, die sich mit der Ausgabe der ganzen Datei beschäftigen.

7.1.1 tac

Das Kommando `tac` dreht die Reihenfolge der Datensätze (Voreinstellung: Zeilen) in Dateien um. Diese werden dann zusammengefügt und ausgegeben.

```
tac [OPTIONEN] [DATEILISTE]
```

`tac` dreht die Reihenfolge der Datensätze in einer Datei um. Werden keine speziellen Datensatztrenner (Seperatoren) angegeben, wird die Zeile als Datensatz interpretiert.

Optionen

`-r`	Seperator wird als regulärer Ausdruck interpretiert
`-s SEP`	Neuen Separator eingeben (Voreinstellung: NeueZeile)

Beispiel
Gibt die Datei `t1` in umgekehrter Reihenfolge in die Datei `t2` aus.
`tac t1 > t2`

7.1.2 nl

Mit diesem Kommando werden der Ausgabe einer Textdatei Zeilennummern hinzugefügt.

```
nl [OPTIONEN] [DATEILISTE]
```

`nl` fügt für jede Zeile einer Textdatei eine Zeilennummer hinzu. Die Nummerierung beginnt bei jeder neuen Seite wieder neu.

Eine Seite gliedert sich in drei Abschnitte: header, body und footer (Kopf, Rumpf und Fuß). Die Abschnitte werden durch Zeilen eingeleitet die nur folgende Zeichenfolgen enthalten:

`\:\:\:`	für den Kopf
`\:\:`	für den Rumpf
`\:`	für den Fuß

Optionen

`-b STYLE`	Wählt die Nummeriungsart für den Rumpf aus. Wird eine Zeile nicht nummeriert, so wird der Zeilenzähler nicht erhöht. Das Trennzeichen wird aber eingefügt. Als Stilarten stehen zu Verfügung `-a` alle Zeilen nummerieren `-n` keine Zeilen nummerieren (header, footer) `-t` alle Zeilen mit Inhalt nummerieren (body) `-pREGEXP` nur Zeilen, die den Ausdruck REGEXP enthalten, nummerieren
`-h`	Wie `-b`, aber für den Kopf.
`-f`	Wie `-b`, aber für den Fuß.
`-d CD`	Setzt die Abschnittsmarkierung neu (Voreinstellung \:)
`-i N`	Schrittweite der Nummerierung. Für jede Zeile wird N hinzuaddiert. (Voreinstellung: 1)
`-l N`	Zählt N aufeinanderfolgende leere Zeilen als eine (Voreinstellung: 1)
`-n FORMAT`	Setzt das Format für die Zeilenummeriung fest `ln` links ausgerichtet ohne führende Nullen `rn` rechts ausgerichtet ohne führende Nullen `rz` rechts ausgerichtet mit führenden Nullen
`-p`	Startet die Nummeriung am Anfang einer logischen Seite nicht neu
`-s STRING`	STRING ist die Trennzeichenkette zwischen Nummerierung und Zeile (Voreinstellung: TAB-Zeichen)
`-v N`	Startet die Nummeriung mit N (Voreinstellung: 1)
`-w N`	Anzahl der Zeichen für die Nummerierung (Voreinstellung: 6)

7.1.3 od

Das Kommando `od` gibt den Inhalt einer Datei in Oktal-, Dezimal-, Hexadezimal- oder ASCII-Darstellung auf der Standardausgabe aus.

```
od [OPTIONEN] [DATEILISTE]
```

Dieser Befehl wird meistens zur Ausgabe von Binärdateien eingesetzt.

Optionen

`-b`	Byteweise Ausgabe in oktaler Form
`-c`	Byteweise Ausgabe in ASCII-Format
`-d`	Dezimale Ausgabe von je zwei Byte
`-x`	Hexadezimale Ausgabe von je zwei Byte

7.2 Textformatierung

Die Befehle `fmt`, `pr` und `fold` formatieren den Dateitext um ihn z. B. für Ausgaben zu optimieren.

7.2.1 fmt

Das Kommando `fmt` erzeugt durch Trennen und Zusammenfügen Ausgabezeilen mit vorgegebener Zeilenlänge. Dabei liest `fmt` aus Dateien oder von der Standardeingabe und gibt das Ergebnis auf der Standardausgabe aus.

```
fmt [OPTIONEN] [DATEILISTE]
```

Optionen

`-s`	Erlaubt nur das Zerlegen von Zeilen und nicht das Zusammenfügen von zu kurzen Zeilen
`-u`	Reduziert die Anzahl der Leerzeichen zwischen Wörtern auf ein und zwischen Sätzen auf zwei Leerzeichen
`-WIDTH`	Gibt Zeilen mit einer Breite von `WIDTH` Zeichen aus. (Voreinstellung: 75)
`-w WIDTH`	
`-p PREFIX`	Es werden nur Zeilen verarbeitet, die mit der Zeichenfolge PREFIX beginnen. Führende Leerzeichen werden ignoriert. Führende Leerzeichen und PREFIX werden entfernt, die Zeilen werden umgebrochen und dann werden vor jeder Zeile die Leerzeichen und der PREFIX wieder angefügt. Einsatzgebiet: Umbrechen von Kommentaren in Programmen, ohne daß der Programmcode verändert wird.

Beispiel
Diese Kommandosequenz entfernt aus einer HTML-Datei alle überzähligen Leerzeichen.
`fmt -u index.html > index.neu.html`

Hier werden die Kommentare in einem Perl-Skript auf eine Zeilenlänge von 70 Zeichen umgebrochen.
`fmt -p \# -w 70 sort.pl > sort.new.pl`

7.2.2 pr

Das Kommando `pr` formatiert eine Textdatei entsprechend festgelegter Optionen.

`pr [OPTIONEN] [DATEILISTE]`

Wenn keine Optionen ausgewählt werden, werden die Dateien seitenweise formatiert. Jede Seite besteht aus 66 Textzeilen: einem fünfzeiligen Kopf, einem Textbereich und einem fünfzeiligen Fuß. Die Kopfzeile enthält Seitenzahl, Dateiname, Datum und Uhrzeit. Alternativ kann der Text auch in Spalten ausgegeben werden.

Optionen

`+ANFANG[:ENDE]`	Ausgabe von Seite ANFANG bis Seite ENDE
`-SP`	Gibt den Text in SP Spalten aus (Voreinstellung: 1)
`-c`	Ausgabe von nichtdruckbaren Zeichen (show-control-**c**har)
`-d`	Fügt Leerzeilen ein (**d**ouble-space)
`-f`	Verwendet Zeilenvorschub anstatt NeueZeile
`-F`	Kopf- und Fußbereich werden auf drei Zeilen reduziert (**f**orm-feed)
`-h HEADER`	Ersetzt den Dateinamen im Kopf durch die Zeichenkette HEADER
`-l LENGTH`	Setzt die Seitenlänge (Voreinstellung: 66 Zeilen) (page-**l**ength)
`-m`	Fügt Dateien spaltenweise zusammen (**m**erge)
`-n[SEP[ZAHL]]`	Nummeriert die Zeilen durch `SEP` Trennungszeichen `ZAHL` Anzahl der Ziffern
`-N ANFANG`	Beginnt Nummeriung der Zeilen mit ANFANG (nur im Zusammenhang mit `-n` verwendet)
`-s SEP`	Trennzeichen für Spalten
`-S SEP`	Trennzeichenkette für Spalten
`-t`	Führt keine Seitenformatierung durch
`-w WIDTH`	Einstellung Textbreite auf WIDTH Zeichen

Beispiele
Gibt die Datei zweispaltig von Seite 4 bis Seite 10 aus.
`pr -2 +4:10 linux.tex`

Gibt die Datei mit Zeilennummerierung (Trennzeichen ':', 4 Ziffern) auf dem Drucker aus.
`pr -n:4 hallo.pl > lpr`

7.2.3 fold

Das Kommando **fold** gibt den Inhalt einer Datei aus, wobei die Zeilen auf eine definierte Länge umgebrochen werden.

`fold [OPTIONEN] [DATEILISTE]`

Im Normalfall ermittelt **fold** die Bildschirmspalten und bricht die Zeilen auf diese Länge um. Ein Tabulatorzeichen kann mehrere Spalten umfassen.

Optionen

`-b`	Zählung nach Bytes und nicht nach Spalten (**b**ytes)
`-s`	Umbruch erfolgt wenn möglich nur an Leerzeichen (**s**paces)
`-w WIDTH`	Stellt Textbreite auf WIDTH Spalten/Bytes ein (**w**idth)

Beispiel
Die Zeilen der Datei werden auf maximal 70 Spalten umgebrochen. Der Umbruch erfolgt nur an Leerzeichen und die Ausgabe erfolgt über `less` auf der Standardausgabe.
`fold -s -w 70 kn20000904.txt | less`

7.3 Teilen von Texten

Diese Befehlsgruppen zeigt nur Teile des Textes (`head` und `tail`) oder zerlegt ihn in mehrere Teile (`split`).

7.3.1 head

Das Kommando `head` gibt den Anfang einer Datei (Voreinstellung 10 Zeilen) auf der Standardausgabe auf.

`head [OPTIONEN] [DATEILISTE]`

Werden mehrere Dateien angegeben, so fügt `head` den Dateinamen in der Form
`==> DATEINAME <==`
als Kopf vor dem Ausgabetext ein. Der Befehl `tail` (siehe 7.3.2) arbeitet ähnlich.

Optionen

`-c B`	Gibt anstatt der ersten 10 Zeilen, die ersten B Zeichen aus.
`-n N`	Gibt anstatt der ersten 10 Zeilen, die ersten N Zeilen aus.
`-N`	
`-q`	Unterdrückt die Ausgabe der Dateinamen als Kopfzeile
`-v`	Schreibt immer den Dateinamen als Kopfzeile vor der Ausgabe

Beispiel
Diese Kommandosequenz gibt die ersten 5 Zeilen der Datei mit dem Dateinamen als Kopf aus.
`head -n 5 -v links.html`

7.3.2 tail

Das Kommando `tail` gibt das Ende einer Datei (Voreinstellung 10 Zeilen) auf der Standardausgabe auf.

`tail [OPTIONEN] [DATEILISTE]`

Werden mehrere Dateien angegeben, so fügt `tail` den Dateinamen in der Form
`==> DATEINAME <==`
als Kopf vor dem Ausgabetext ein. Der Befehl `head` (siehe 7.3.1) arbeitet ähnlich.

Optionen

`-c B`	Gibt anstatt der letzten 10 Zeilen, die letzten B Zeichen aus.
`-n N`	Gibt anstatt der letzten 10 Zeilen, die letzten N Zeilen aus.
`-N`	
`-q`	Unterdrückt die Ausgabe der Dateinamen als Kopfzeile
`-v`	Schreibt immer den Dateinamen als Kopfzeile vor der Ausgabe
`-f`	Mit dieser Option überwacht `tail` kontinuierlich das Ende einer oder mehrerer Dateien. Wird an die Datei etwas angehängt, so werden die Änderungen ausgegeben. Die Überwachung wird mit STRG+C abgebrochen.

Beispiel
Diese Kommandosequenz gibt die letzten fünf Zeilen der Dateien ohne den Dateinamen als Kopf aus.
`tail -n 5 -q seite1.html seite2.html seite3.html`

Mit der folgenden Sequenz werden mehrere Dateien auf Veränderung überwacht.
`tail -fn 15 /etc/passwd /etc/group /tmp/mylogfile`

7.3.3 split

Das Kommando **split** zerlegt Dateien in mehrere kleinere Dateien.

`split [OPTIONEN] [DATEI [PREFIX]]`

Wird keine Datei angegeben bzw. "-" als Dateiname, dann wird von der Standardeingabe gelesen.

`split` zerlegt eine große Datei in kleinere Dateien vordefinierter Größe (Voreinstellung: 1000 Zeilen). Die dabei entstehenden Dateien beginnen mit einem Prefix (Voreinstellung: x) und werden zweistellig mit Buchstaben durchnummeriert. (`xaa`, `xab`, `xac`, ...).

Optionen

`-LINES`	Größe der Ergebnisdateien in Zeilen
`-l LINES`	
`-b BYTES`	Größe der Ergebnisdateien in Bytes `b` angehängt bedeutet BYTES x 512 `k` angehängt bedeutet BYTES x 1024 `m` angehängt bedeutet BYTES x 1048576

7.4 Textstatistik

Die folgenden Befehle analysieren den Text (`wc`) oder ermitteln eine Checksumme (`sum` und `chsum`) für die Fehlerkontrolle.

7.4.1 wc

Das Kommando `wc` zählt die Zeichen, die durch Leerzeichen getrennten Worte und die Zeilen einer Eingabe.

`wc [OPTIONEN] [DATEILISTE]`

`wc` gibt pro Datei die Anzahl der Zeilen, die Anzahl der Worte, die Anzahl der Zeichen und den Namen der Datei. Werden mehrere Dateien angegeben, so wird am Schluß eine Zusammenfassung aller Dateien angezeigt.

Optionen

`-c`	Anzahl der Zeichen ausgeben
`-w`	Anzahl der Worte ausgeben
`-l`	Anzahl der Zeilen ausgeben

Beispiel
Gibt die Anzahl der Worte und Zeilen in den drei Dateien an.

```
wc -lw artikel1.txt artikel2.txt artikel3.txt
```

Bestimmt die Anzahl der Dateien im Verzeichnis /etc. `ls /etc | wc -l`

7.4.2 sum

Das Kommando `sum` ermittelt eine 16-bit-Checksumme und die Größe (in 1 k-Blöcken) einer Datei.

```
sum [OPTIONEN] [DATEILISTE]
```

7.4.3 cksum

Das Kommando `cksum` ermittelt eine CRC-Checksumme und die Größe (in Bytes) einer Datei.

```
cksum [OPTIONEN] [DATEILISTE]
```

Diese Funktion benutzt einen stabileren Algorithmus (CRC: cyclic redundancy check) als `sum` (siehe 7.4.2).

7.5 Sortieren

Daten werden meistens erst dann richtig lesbar, wenn Sie sortiert werden. Die Tools `sort`, `comm` und `uniq` kümmern sich um diesen Bereich.

7.5.1 sort

Der Befehl `sort` durchsucht, verbindet oder vergleicht alle Zeilen einer Datei.

```
sort [OPTIONEN] [DATEILISTE]
```

`sort` kennt drei Arten der Funktion. Sortieren, Verbinden und Prüfen auf Sortierung.

Optionen	
`-c`	Prüft, ob die Datei sortiert ist (**c**heck sort)
`-m`	Verbindet mehrere Dateien (die nicht sortiert werden, sondern als Gruppe behandelt werden) miteinander
`-b`	Führende Leerzeichen eines Schlüsselfeldes werden ignoriert
`-d`	Sortieren als Telefonverzeichnis: Nur Buchstaben, Ziffern und Leerzeichen werden berücksichtigt
`-f`	Unterscheidet nicht zwischen Groß- und Kleinbuchstaben
`-g`	Sortiert nach Zahlen anstatt lexikalisch
`-n`	Die Optionen unterscheiden sich in der Konvertierung der Zahlen zum Vergleich
`-o DATEI`	Ausgabe in die Datei `DATEI` umlenken
`-r`	Umkehrung der Sortierung
`-t SEP`	Trennzeichen für die Felder definieren
`-u`	Unterdrückung von gleichen Zeilen
`+POS1[-POS2]`	Als Schlüsselfelder werden die Felder ab POS1 bis POS2 (oder bis zum Zeilenende) verwendet.

Beispiel
Die Zählung der Spalten beginnt bei 0! Daher sortiert dieser Befehl die Datei `mai.log` nach der sechsten Spalte. Trennzeichen ist das Leerzeichen.

```
sort +5 -t " " mai.log > hits.log
```

Dieser Kommandosequenz sortiert eine Datei nach Telefonbuchmethode und unter Vernachlässigung der Groß- und Kleinschreibung.
`sort -fd telefon.buch | less`

7.5.2 comm

Das Kommando `comm` vergleicht zwei sortierte Dateien miteinander.

`comm [OPTIONEN] DATEI1 DATEI2`

`comm` gibt drei Spalten aus. In den ersten beiden Spalten stehen die Zeilen aus DATEI1 und DATEI2, die nicht in der jeweiligen anderen Datei vorkommen. In der dritten Zeile werden die gleichen Zeilen ausgegeben.

7.5.3 uniq

Das Kommando `uniq` entfernt aus einer sortierten Datei die doppelten Zeilen.

`uniq [OPTIONEN] [INPUT [OUTPUT]]`

Optionen	
`-c`	Gibt an, wie oft die Zeile vorkommt
`-i`	Ignoriert Groß- und Kleinschreibung
`-d`	Gibt nur die doppelten Zeilen aus
`-u`	Gibt nur die nicht doppelten Zeilen aus

Beispiel
Dieser Befehl extrahiert die siebente Spalte aus der Datei `mai.log`, sortiert sie, entfernt die doppelten Zeilen und gibt an, wie oft die doppelten Zeilen vorkamen.
`cut -f 7 -d " " mai.log | sort | uniq -c | less`

7.6 Zeilenoperationen

Um einen Text Zeile für Zeile zu analysieren und zu bearbeiten sind die folgenden Befehle programmiert worden.

7.6.1 cut

Das Kommando `cut` gibt ausgewählte Felder (Spalten) aus jeder Zeile einer Datei auf der Standardausgabe aus.

`cut [OPTIONEN] [DATEILISTE]}`

Optionen	
`-b BYTES`	Zeige die angegebenen BYTES an.
`-f NR`	Zeige das Feld (die Spalte) Nummer `NR` (**f**ield)
`-d ZEICHEN`	`ZEICHEN` ist der Spaltentrenner (**d**elimiter)
`-s`	Zeilen ohne Trennzeichen (delimiter) werden nicht ausgegeben

Das Tabulatorzeichen ist als Trennzeichen voreingestellt. Das Trennzeichen kann ein Leerzeichen oder ein anderes Sonderzeichen sein. Dann sollte es in Anführungsstriche gesetzt (oder allgemeiner maskiert) werden.

Beispiel
Diese Kommandosequenz gibt die Benutzernamen aus der Datei `/etc/passwd` aus.
`cut -f 1 -d : /etc/passwd`

Gibt die ersten 10 Zeichen jeder Zeile aus.
`ls -l | cut -b -10`

7.6.2 join

Das Kommando `join` gibt die Zeilen aus zwei Dateien aus, die identische Vergleichsfelder besitzen.

```
join [OPTIONEN] DATEI1 DATEI2
```

Optionen	
`-j FIELD`	Vergleichsfeld aus DATEI1 und DATEI2 angeben
`-j1 FIELD`	Vergleichsfeld aus DATEI1 angeben
`-j2 FIELD`	Vergleichsfeld aus DATEI2 angeben
`-t CHAR`	Das Zeichen CHAR ist Feldtrennzeichen
`-i`	Ignoriert Groß- und Kleinschreibung

Beispiel
Dieser Befehl vergleicht das 2. Feld in t1.txt mit dem 3. Feld in t2.txt und gibt bei Gleichheit die passenden Zeilen aus. `join -j1 2 -j2 3 t1.txt t2.txt`

7.6.3 paste

Das Kommando `paste` fügt Dateien spaltenweise zusammen. Jede Datei erhält eine eigene Spalte.

```
paste [OPTIONEN] [DATEILISTE]
```

`paste` ohne Angabe von Dateien funktioniert wie `cat` (siehe 4.5.2). Ein `-` in der Dateiliste führt zu einem Lesen von der Standardeingabe.

`paste` eignet sich zum Zusammenführen von Daten aus mehreren Dateien.

Optionen	
`-d ZEICHEN`	ZEICHEN ist das Spaltentrennzeichen
`-s`	Verknüpt die Zeilen einer Datei zu einer einzigen langen Zeile

Beispiel
Die Dateien werden zeilenweise zusammengefügt und die einzelnen Spalten mit einem `;` getrennt.
`paste -d ; sonnenwind.dat magnetfeld.dat teilchen.dat > new.dat`

Diese Kommandosequenz liest von der Standardeingabe und schreibt die Eingaben mit einem `+` getrennt in eine Zeile in die Datei `punkte.dat`.
`paste -d + -s > punkte.dat`

7.7 Suchen und Ersetzen

Um bestimmte Stellen aus Texten zu extrahieren oder zu änderen werden die Tools `grep`, `tr`, `expand`, `sed`, `diff` und `patch` verwendet.

7.7.1 grep

Der Befehl `grep` (*Global Regular Expression Print*) durchsucht eine Textdatei nach bestimmten Mustern und gibt die Zeilen, in denen das Muster vorkommt, auf der Standardausgabe aus.

```
grep [OPTIONS] MUSTER [DATEILISTE]
```

Optionen

`-G`	Interpretiert das `MUSTER` als regulären Ausdruck; Standarteinstellung. Nicht zusammen mit `-F` und `-E` verwenden
`-E`	Interpretiert das `MUSTER` als erweiterten regulären Ausdruck. Nicht zusammen mit `-F` und `-G` verwenden
`-F`	Interpretiert das `MUSTER` als einfache Zeichenkette. Nicht zusammen mit `-F` und `-E` verwenden
`-c`	Zeigt nur die Zeilennummern der gefundenen Zeilen an
`-n`	Zeigt zusätzlich zur Zeile auch die Zeilennummer an
`-v`	Zeigt die Zeilen an, die nicht dem `MUSTER` entsprechen
`-f DATEINAME`	Die Liste der zu bearbeitenden Dateien
`-h`	Unterdrückt die Ausgabe des Dateinamens bei Verwendung einer Dateiliste.
`-i`	Unterscheidet bei der Suche nicht nach Groß- und Kleinschreibung
`-w`	`MUSTER` wird als ganzes Wort und nicht als Teil des Wortes betrachtet
`-l`	Zeigt den Namen der Datei an, wenn die Zeile darin gefunden wurde
`-s`	Fehlermeldungen unterdrücken
`-r`	Durchsucht auch die Unterverzeichnisse

Beispiele
Durchsucht die Datei `kuno.txt` nach Zeilen mit dem Wort `Bruno`.

```
grep Bruno kuno.txt
```

Durchsucht alle Dateien im Verzeichnis nach der Zeichenkette 'midnight' ohne Berücksichtigung der Groß- und Kleinschreibung und gibt die Namen der Dateien aus, die die Zeichenkette enthalten.

```
grep -il midnight *
```

7.7.2 egrep

Das Kommando `egrep` entspricht dem Befehl `grep -E`. Es verwendet erweiterte reguläre Ausdrücke.

7.7.3 fgrep

Das Kommando `fgrep` entspricht dem Befehl `grep -F`. Es benutzt als Suchmuster nur reine Zeichenketten.

7.7.4 tr

Das Kommando `tr` sendet die Standardeingabe zur Standardausgabe, wobei es mehrere Operationen auf die Daten ausführt. Mögliche Operationen sind:

- Verändern von Zeichen und das Löschen von Zeichen aus dem Ergebnis, die wiederholt werden.
- Löschen von Zeichen, die wiederholt werden.
- Löschen von Zeichen.
- Löschen von Zeichen und das Löschen von Zeichen aus dem Ergebnis, die wiederholt werden.

```
tr [OPTIONEN] ZEICHENKETTE1 [ZEICHENKETTE2]
```

Optionen

`-d`	Löscht die angegebenen Zeichen
`-s`	Löscht doppelt vorkommende Zeichen

Beispiele
So wandelt `tr` mit dem Befehl

```
cat stundenplan.txt | tr m M > stundenplan.neu
```

jedes Vorkommen des Buchstabens 'm' in den Buchstaben 'M' um.

Das folgende Kommando löscht die Zeichen 'm' und 'y'.
`cat etwas.txt | tr -d my`

Doppelt vorkommende Zeichen werden mit dem Schalter `-s` gelöscht.
`tr -s le < harry.txt` Dabei wird dann aus der 'Allee' ganz schnell 'Ale'.

Um den einfachen ROT13 Verschlüsselungsalgorithmus zu verwenden, reicht die folgende Sequenz.
`cat harry.txt | tr '[A-M][N-Z][a-m][n-z]' '[N-Z][A-M][n-z][a-m]'`
oder
`cat harry.txt | tr 'A-Za-z' 'N-ZA-Mn-za-m'`
Dabei wird der erste Buchstabe mit dem 13. Buchstaben, der zweite mit dem 14. Buchstaben, u. s. w. getauscht. Wendet man ROT13 wieder auf verschlüsselten Text an, so erhält man den Originaltext.

7.7.5 expand

Das Kommando `expand` liest aus Dateien oder von der Standardeingabe, wandelt die Tabzeichen in Leerzeichen und gibt das Ergebnis auf der Standardausgabe aus. Wird nichts anderes angegeben, dann wird ein Tabulator durch acht Leerzeichen ersetzt.

```
expand [OPTIONEN] [DATEILISTE]
```

Optionen

`-TAB1[,TAB2[,...]]`
`-t`
`TAB1[,TAB2[,...]]`
	TABx ist die Anzahl der Leerzeichen, durch die das jeweilige Tabulatorzeichen ersetzt werden soll.

`-i`	Tabulatoren ohne vorhergehende Leerzeichen werden nicht konvertiert.

Wird nur für das erste Tabulatorzeichen ein Wert angegeben, so gilt dieser Wert für alle Tabulatoren. Werden zwei oder mehr Werte eingegeben, so gelten Sie für die jeweiligen Tabulatoren. Tabulatoren, für die kein Wert angegeben wurde, werden durch einfache Leerzeichen ersetzt.

Beispiel
Diese Kommandosequenz ersetzt den ersten Tabulator jeder Zeile der Datei `helloworld.java` durch 6 Leerzeichen, den zweiten durch 8 Leerzeichen und alle folgenden Tabulatoren durch ein Leerzeichen. Das Ergebnis wird in die Datei `helloworld.txt` geschrieben. `expand -6,8 helloworld.java > helloworld.txt`

7.7.6 sed

`sed` ist die Abkürzung für **S**treaming **Ed**itor. Eigentlich ist das Kommando kein Editor sondern ein Textfilter, der bestimmte Zeichenkombinationen sucht und ersetzt. Die Verarbeitung eines Streams oder Datenstroms bedeutet, daß von der Standardeingabe (`stdin`) Daten angenommen werden und auf die Standardausgabe (`stdout`) wieder ausgegeben werden. Als Datenquelle kann neben der Standardeingabe auch eine Datei fungieren.

```
sed [OPTIONEN] [KOMMANDO] [DATEILISTE]
sed [OPTIONEN] [-f SCRIPTFILE] [DATEILISTE]
```

Es werden grundsätzlich zwei Möglichkeiten genutzt um `sed` aufzurufen. Im ersten Fall wird das `KOMMANDO` auf der Befehlszeile eingegeben und auf die angegebenen Dateien angewendet. Im zweiten Fall stehen die Anweisungen in einer externen Skriptdatei. Diese wird abgearbeitet und die darin enthaltenen Kommandos auf den Inhalt der Dateien angewendetet. Sollte keine Datei angegeben sein, so liest `sed` von der Standardeingabe. Dabei wird jede gelesene Zeile mit den `sed`-Kommandos bearbeitet und in einen Puffer geschrieben. Dessen Inhalt wird dann zum Schluß auf der Standardausgabe ausgegeben.

Optionen

`-V`	Versionsnummer
`-h`	Hife
`-e SCRIPT`	Zusätzliche Skriptanweisung zur Bearbeitung
`-f SCRIPTFILE`	Datei, die die Skriptbefehle enthält
`-n`	Ausgabe erfolgt nur bei Benutzung des 'p' Kommandos

Jeder Befehl kann durch einen Bereich, für den er gültig ist, eingeschränkt werden. Die Bereichsangabe erfolgt im Format
`VON`
oder
`VON, BIS`.

Dabei können die Werte für `VON` und `BIS`

- die Zeilenummern,
- das Zeichen $ für das Ende der Datei oder
- ein Suchmuster, begrenzt durch das Zeichen `/`, z. B. `/unix/`, sein.

Ersetzen Die folgenden Sequenzen weisen `sed` an, eine Zeichenkette durch eine andere zu ersetzen.

```
sed s/SUCHEN/ERSETZEN/
sed s/SUCHEN/ERSETZEN/g
sed s/SUCHEN/ERSETZEN/p
sed -n s/SUCHEN/ERSETZEN/gp
sed s3/SUCHEN/ERSETZEN/
```

Die Zeichenkette zwischen dem ersten und dem zweiten Schrägstrich gibt das Suchmuster an, während die Zeichenkette zwischen dem zweiten und dem dritten Schrägstrich den Ersetzungstext enthält.

Um das Suchmuster effektiver zu gestalten, können reguläre Ausdrücke[1] verwendet werden. Diese bestehen normalerweise aus zwei Komponenten: Die Angabe, nach welchem Zeichen gesucht wird, und die Angabe, wie oft das Zeichen auftreten darf.

Die hinten angestellten Optionen verändern die Arbeitsweise von `sed`. Dabei bewirkt `g`, daß alle Zeichenketten in der Zeile durch die neue Zeichenkette ersetzt werden (Sonst nur die erste Zeichenkette). Das `p` führt dazu, daß die Zeilen ausgegeben werden, in denen eine Ersetzung erfolgte. Sind beide Optionen gesetzt, so wird bei einer mehrfachen Ersetzung in einer Zeile die Zeile auch mehrfach angezeigt. Verhindert wird dies durch den Schalter `-n`. Das `s3` sorgt dafür, daß das dritte Vorkommen von `SUCHEN` in der Zeile ersetzt wird.

Suchen Die Sequenz
`/SUCHEN/p`
im Zusammenhang mit dem Schalter `-n` bewirkt, daß nur die Zeilen, die das Suchmuster enthalten, ausgegeben werden.

Zeilen löschen Wenn der Schalter `-n` nicht gesetzt ist, werden alle Zeilen nach der Bearbeitung ausgegeben. Um zu verhindern, daß bestimmten Zeilen ausgegeben werden, wird die Sequenz
`/SUCHEN/d`
verwendet.

Dateien einfügen Durch die Sequenz
`/SUCHEN/r DATEINAME`
werden alle Zeilen, die das Suchmuster enthalten, durch den Inhalt der Datei DATEINAME ersetzt.

[1]Weitere Informationen bekommen Sie unter `man 7 regex`.

Auswahl in Datei schreiben Durch die Sequenz
`/SUCHEN/w DATEINAME`
werden alle Zeilen, die das Suchmuster enthalten, in die Datei DATEINAME geschrieben.

Beispiele
Ersetzt jedes Wort "UNIX" durch das Wort "Linux" in der Datei `einleitung.tex`.

```
sed 's/UNIX/Linux/g' einleitung.tex
```

Ersetzt jedes Wort "paragraph" durch das Wort "subsubsection" in der Datei `linux.tex` und schreibt das Ergebnis in die Datei `linuxneu.tex`.

```
sed 's/paragraph/subsubsection/g' linux.tex > linuxneu.tex
```

Einen interessanten Effekt hat das Zeichen '`&`' in der Ersetzungszeichenkette. Die gesuchte Zeichenkette wird nicht ersetzt, sondern die Ersetzungszeichenkette wird hinten angefügt.

```
sed 's/</&lt;/g' index.html
```

Damit die schließende spitze Klammer ersetzt wird, muß das kaufmännische Und auskommentiert werden.

```
sed 's/</\&lt;/g' index.html
```

Ersetzt alle Zeichenfolgen "man" durch "frau" in den Zeilen 1 bis 3 in der Datei `einleitung.tex`. Dabei werden nur die Zeilen ausgegeben, in denen die Änderung erfolgte.

```
sed -n '1,3s/man/frau/p' einleitung.tex
```

Gibt alle Zeilen aus, die entweder "man" oder "frau" enthalten.

```
sed -n -e '/man/p' -e '/frau/p' einleitung.tex
```

Das folgende Kommando löscht alle Zeilen, die die Zahl '0' enthalten.

```
sed '/0/d' einleitung.tex
```

Alle Zeilen, die die Zeichenkette "include" enthalten, werden durch den Inhalt der Datei `include.txt` ersetzt.

```
sed '/include/r include.txt' einleitung.txt
```

Alle Zeilen, die die Zeichenkette "section" enthalten, werden in die Datei `inhalt.txt` geschrieben.

```
sed '/section/w inhalt.txt' einleitung.tex
```

7.7.7 diff

Das Kommando `diff` vergleicht zwei Textdateien miteinander. Als Ergebnis wird eine Liste aller Zeilen ausgegeben, die voneinander abweichen. Dabei erkennt das Programm auch eingefügte Zeilen und arbeitet danach reibungslos weiter. Im Gegensatz zu `comm` (siehe 7.5.2) kann `diff` auch auf unsortierte Dateien angewendet werden. Der Befehl wird hauptsächlich dazu verwendet die Abweichungen zwischen zwei Versionen eines Programmlistings rasch zu dokumentieren.

```
diff [OPTIONEN] DATEI1 DATEI2
```

Optionen

`-b`	Betrachtet mehrfache Leerzeichen und Leerzeilen als einfache Leerzeichen bzw. Leerzeilen
`-c`	Ausgabe als 'Context Diffs'
`-e`	Ausgabe als Skript für den Befehl `ed`
`-r`	Vergleicht den Inhalt zweier Verzeichnisbäume
`-u`	Ausgabe als 'Unified Diffs'
`-w`	Ignoriert Leerzeichen und Leerzeilen ganz

Die Ausgabe von `diff` besteht aus sogenannten 'hunks', was man frei als 'Stücke' übersetzen kann. Ein Hunk besteht aus bis zu vier Teilen: einer Informationszeile, dem alten Text, einer Trennzeile und dem neuen Text. Beim Einfügen und Löschen von Zeilen fehlt die alte oder neue Textinformation und die dann überflüssige Trennzeile fällt auch weg.

In der Informationszeile steht, an welcher Stelle in der jeweiligen Datei die Änderung stattgefunden hat und welche Operationen nötig sind um die erste Eingabedatei `DATEI1` in die zweite Eingabedatei `DATEI2` zu verwandeln. Dabei stehen die Buchstaben 'a', 'c' und 'd' für 'add' (Zeilen hinzufügen), 'change' (Zeilen verändern) und 'delete' (Zeilen löschen). Die Zahlen bzw. Zahlenbereiche links und rechts von den Buchstaben geben die betroffenen Zeilen an.

Alle Hunks zusammengenommen bilden den Patch, der die erste Datei in die zweite Datei verwandelt. Es geht auch umgekehrt, da Patches dieser Art symmetrisch sind und rückwärts angewendet werden können (‘reverse patching´);

Beispiel
Um zwei Perlscripte miteinander zu vergleichen, ohne mehrfache Leerzeichen zu berücksichtigen, gibt man das folgende Kommando ein.

```
diff -b div.pl.old div.pl
```

Soll die Ausgabe für den Editor `ed` verarbeitbar sein, dann muß das Kommando lauten:

```
diff -be div.pl.old div.pl
```

Die Anwendung von `diff` fürs Patchen von Programmen beschreibt der Abschnitt 7.7.9.

7.7.8 patch

Da das Patchen von Programmen mit `ed` einige Probleme aufweist hat Larry Wall das Programm `patch` entwickelt, das später vom GNU-Projekt weiterentwickelt wurde. Es ist in der Lage, alle gängigen `diff`-Formate zu verstehen.

```
patch [OPTIONEN] [ORIGINALDATEI [PATCHDATEI]]
```

Optionen

`-c`	Interpretiere den Patch als Context Diff
`-e`	Interpretiere den Patch als ed-Skript
`-u`	Interpretiere den Patch als Unified Diff

7.7.9 Praxisbeispiel: Patchen von Programmen

Das Patchen von Programmen hat viele Vorteile. So muß z. B. nicht das ganze Programm heruntergeladen oder per eMail versandt werden. Außerdem enthält der Patch nur die Änderungen im Programm. So muß ein Mitautor nicht erst mühselig im Programm die Änderungen suchen, sondern sie werden ihm komprimiert übergeben.

Als Beispiel schauen wird uns zwei Versionen eines Perl-Skripts an.

Listing 7.1 *div.pl*

```
#!/usr/bin/perl

# Dieses Programm teilt zwei Zahlen durcheinander

# Eingabe
print "Zähler: ";
$z = <STDIN>;
chomp($z);
print "Nenner: ";
$n = <STDIN>;
chomp($n);

# Verarbeitung
$erg = int($z / $n);
$rest = "Rest ". ($z % $n);

```

```
17: # Ausgabe
18: print "$z / $n = $erg $rest\n";
19:
20: exit(0);
```

Listing 7.2 *div2.pl*

```
 1: #!/usr/bin/perl
 2:
 3: # Dieses Programm teilt zwei Zahlen durcheinander
 4: # Version 2
 5:
 6: # Eingabe
 7: print "Zähler: ";
 8: $z = <STDIN>;
 9: chomp($z);
10: print "Nenner: ";
11: $n = <STDIN>;
12: chomp($n);
13:
14: # Verarbeitung
15: # Verarbeitung nur, wenn der Nenner ungleich Null ist
16: if ($n != 0) {
17:   $erg = int($z / $n);
18:   $rest = "Rest ". ($z % $n);
19: } else {
20:   $erg = "unendlich";
21:   $rest = "";
22: }
23:
24: # Ausgabe
25: print "$z / $n = $erg $rest\n";
26:
27: exit(0);
```

`diff` liefert uns nun die Unterschiede der beiden Skripte.

```
tapico@defiant:~/perl > diff div.pl div2.pl
3a4
> # Version 2
14,15c15,22
< $erg = int($z / $n);
< $rest = "Rest ". ($z % $n);
---
> # Verarbeitung nur, wenn der Nenner ungleich Null ist
> if ($n != 0) {
>   $erg = int($z / $n);
>   $rest = "Rest ". ($z % $n);
> } else {
>   $erg = "unendlich";
>   $rest = "";
> }
```

Manuell können mit diesen Informationen nun die Änderungen an der Datei `div.pl` durchgeführt werden.

- Füge hinter der dritten Zeile als vierte Zeile den Kommentar ein.
- Lösche die Zeilen 14 und 15 und füge als Zeilen 15 bis 22 die angegebenen Zeilen ein.

Da dies bei größeren Projekten doch problematisch ist, kann auch ein Skript für den Editor `ed` erstellt werden.

```
tapico@defiant:~/perl > diff -e div.pl div2.pl
14,15c
# Verarbeitung nur, wenn der Nenner ungleich Null ist
if ($n != 0) {
  $erg = int($z / $n);
  $rest = "Rest ". ($z % $n);
} else {
  $erg = "unendlich";
  $rest = "";
}
.
3a
# Version 2
.
```

Wenn Sie die Ausgabe als Datei gespeichert haben, kann nun jemand anders mit dem Editor `ed` die alte Datei (`div.pl`) mit dem Patch (hier die Datei `div.patch`) bearbeiten und daraus die neue Datei (`div2.pl`) erstellen.

```
barclay@enterprise:~/projekt > (cat div.patch; echo w div2.pl) | ed - div.pl
```

Dies setzt aber voraus, daß an der alten Datei nichts geändert worden ist. Daher gibt es zwei andere Formate, die mehr Sicherheit beim Patchen versprechen.

Context Diff

```
tapico@defiant:~/perl > diff -c div.pl div2.pl
*** div.pl Sun Sep 23 14:36:48 2001
--- div2.pl Sun Sep 23 14:36:12 2001
***************
*** 1,6 ****
--- 1,7 ----
  #!/usr/bin/perl

  # Dieses Programm teilt zwei Zahlen durcheinander
+ # Version 2

  # Eingabe
  print "Zähler: ";
***************
*** 11,18 ****
  chomp($n);

  # Verarbeitung
! $erg = int($z / $n);
! $rest = "Rest ". ($z % $n);

  # Ausgabe
  print "$z / $n = $erg $rest\n";
--- 12,25 ----
  chomp($n);

  # Verarbeitung
! # Verarbeitung nur, wenn der Nenner ungleich Null ist
! if ($n != 0) {
!   $erg = int($z / $n);
!   $rest = "Rest ". ($z % $n);
! } else {
!   $erg = "unendlich";
!   $rest = "";
! }

  # Ausgabe
  print "$z / $n = $erg $rest\n";
```

Unified Diff

```
tapico@defiant:~/perl > diff -u div.pl div2.pl
--- div.pl Sun Sep 23 14:36:48 2001
+++ div2.pl Sun Sep 23 14:36:12 2001
@@ -1,6 +1,7 @@
 #!/usr/bin/perl

 # Dieses Programm teilt zwei Zahlen durcheinander
+# Version 2

 # Eingabe
 print "Zähler: ";
@@ -11,8 +12,14 @@
 chomp($n);
```

```
 # Verarbeitung
-$erg = int($z / $n);
-$rest = "Rest ". ($z % $n);
+# Verarbeitung nur, wenn der Nenner ungleich Null ist
+if ($n != 0) {
+  $erg = int($z / $n);
+  $rest = "Rest ". ($z % $n);
+} else {
+  $erg = "unendlich";
+  $rest = "";
+}

 # Ausgabe
 print "$z / $n = $erg $rest\n";
```

Diese beiden Formate, wie auch die anderen, können mit dem Befehl `patch` die Veränderung durchführen.

```
barclay@enterprise:~/projekt > patch < div.c.patch
```

Die im Patch angegebene Datei (`div.pl`) ist nun verändert worden.

Notizen:

Notizen:

VII	Arbeitsblatt Textverarbeitung und Textfilter	7.1

Sollte eine Aufgabe zu einer Fehlermeldung führen, kann das von mir gewollt sein! Prüfen Sie aber dennoch, ob Sie keinen Tippfehler gemacht haben, und ob die Voraussetzungen wie in der Aufgabenstellung gegeben sind. Auch sollten keine Verzeichniswechsel ausgeführt werden, wenn dies nicht ausdrücklich in der Aufgabe verlangt wird! Notieren Sie die Ergebnisse auf einem seperaten Zettel.

1 Loggen Sie sich als Walter ein!

2 Führen Sie den Befehl `rm -rf *` aus.

3 Kopieren Sie aus dem Verzeichnis `/usr/share/man/man1` die Datei `less.1.gz` ins Heimatverzeichnis.

4 Erzeugen Sie aus der Datei `less.1.gz` die HTML-Datei `less.html`.

5 Lassen Sie sich den Inhalt der Datei `less.html` seitenweise anzeigen!

6 Erzeugen Sie mit dem `vi` eine Textdatei, die mindestens 3 Zeilenvorschübe beinhaltet (4 Zeilen lang)! Name der Datei: `funny` !

7 Kopieren Sie die Datei `/usr/share/man/man1/more.1.gz` in Ihr Heimatverzeichnis und machen Sie wie in Aufgabe 4 eine HTML-Datei daraus.

8 Lassen Sie sich Inhalt von `more.html` und `funny` in Ihrem Heimatverzeichnis nacheinander seitenweise auf dem Bildschirm anzeigen, dazu ist nur ein Befehl einzugeben!

9 Erstellen Sie mit einem einzigen Befehl aus den Dateien `more.html` und `funny` eine neue Datei namens `hummer`.

10 Lassen Sie sich den Inhalt von `hummer` anzeigen.

11 Löschen Sie die Dateien `funny` und `hummer` in Ihrem Heimatverzeichnis!

12 Wie lauten die letzten (10) Zeilen der Datei `less.html`.

13 Lassen Sie sich den Inhalt der Datei `less.html` in oktaler Form anzeigen!

14 Kopieren Sie alle Dateien, die mit 'mc' beginnen, aus dem Verzeichnis `/usr/share/man/man1/` in Ihr Heimatverzeichnis.

15 Wandeln Sie wie in Aufgabe 4 die kopierten Dateien in HTML-Dateien.

16 Lassen Sie sich die ersten 5 Zeilen der Datei `mcopy.html` anzeigen.

17 Wieviele Wörter und Zeilen enthält die Datei `more.html`?

18 Welche Datei im Heimatverzeichnis enthält die Zeichenkette 'Midnight'?

19 Schauen Sie sich die Datei `/etc/passwd` an.

20 Wieviele Zeilen enthält sie?

21 Lassen Sie sich die ersten 20 Zeilen der Datei `/etc/passwd` numeriert anzeigen.

22 Geben Sie die erste Spalte der Datei `/etc/passwd` aus.

23 Geben Sie sortiert die erste Spalte der Datei `/etc/passwd` aus.

24 Geben Sie sortiert die erste Spalte der Datei `/etc/passwd` aus und speichern Sie das Ergebnis in der Datei `~/passwd.sort`.

VII	Arbeitsblatt Textverarbeitung und Textfilter	7.2

25 Schreiben Sie den Inhalt der Datei `/etc/passwd` in umgekehrter Zeilenfolge in die Datei `~/passwd.reverse`.

26 Vergleichen Sie die CRC-Checksummen von `/etc/passwd`, `~/passwd.sort` und `~/passwd.reverse`.

27 Lassen Sie sich die Logdatei für Warnmeldungen des Systems `/var/log/warn` anzeigen. Welche Bedeutung haben wohl die einzelnen Spalten?

28 Lassen Sie sich alle Zeilen von heute ausgeben.

29 Lassen Sie sich alle Zeilen vom letzten Arbeitstag ausgeben.

30 Wieviele Warnmeldungen gab es?

31 Geben Sie die beiden ersten Spalten aus.

32 Geben Sie die beiden ersten Spalten aus, entfernen Sie alle doppelten Zeilen.

33 Lassen Sie sich anzeigen, wieviele Warnmeldungen es pro Tag gab.

34 Sortieren Sie diese Liste absteigend nach der Häufigkeit.

35 Verfassen Sie mit dem Editor `vi` eine Botschaft an Ihren Nachbarn und speichern Sie diese in der Datei `gutenachricht.txt`.

36 Verschlüsseln Sie den Inhalt der Datei mit dem ROT13-Verfahren und speichern Sie das Ergebnis in der Datei `gutenachricht.krypto`.

37 Senden Sie Ihrem Nachbarn die Datei zu. (eMail, FTP, Turnschuhnetz etc.)

38 Entschlüsseln Sie die Nachricht Ihres Nachbarn.

39 Was bedeutet der Satz “Qvr Nhstnora fvaq sregvt”?

Notizen:

Kapitel 8

Benutzerverwaltung

8.1 Benutzer

Nach der Installation von Linux existiert nur ein Benutzerkonto. Dieses Benutzerkonto ist *root*, das Konto für den Superuser. Daher muß der Administrator (Superuser) für die Arbeit weitere Konten anlegen.

8.1.1 Der Superuser root

Das Konto *root* wird nicht umsonst als Superuser bezeichnet. Der Benutzer unter diesem Konto darf im System alles. Er hat vollen Zugriff auf alle Verzeichnisse, Dateien und Geräte im System. Damit kann er auch alles löschen. Diese Aktion kann aber zu erheblichen Schäden am System führen, somit sollte man als Superuser sehr vorsichtig handeln.

Der Superuser *root* besitzt die UID 0 (*User IDentification*) und wird wie alle anderen Benutzer in der Datei `/etc/passwd` definiert. Da das System die Benutzer nicht nach ihrem Namen identifiziert, sondern nach ihrer UID, kann der Superuser auch einen anderen Lognamen bekommen. Da Linux wie das Christentum nur einen Gott kennt, nämlich den Benutzer mit der UID 0, ist es ratsam für diesen Benutzer weitere Namen anzulegen. Wenn Sie nämlich das Paßwort für `root` vergessen, wünsche ich Ihnen viel Spaß `;-)`.

Um die Sicherheit ihres Systems zu erhöhen, sollten Sie mehrere Benutzer anlegen, die spezielle Aufgaben erledigen können. Diese Benutzer erhalten spezielle eingeschränkte Rechte für die verschiedenen administrativen Aufgaben. Wie das geht, wird in Abschnitt 8.3.6 beschrieben.

Für die alltäglichen Aufgaben sollten Sie sich nicht als *root* einloggen, da dies ein nicht zu unterschätzendes Risiko darstellt. Im normalen Betrieb sollten Sie als einfacher Benutzer arbeiten. Nur für administrative Aufgaben loggen Sie sich als *root* ein und erledigen die Aufgaben, um danach als einfacher Benutzer weiterzuarbeiten. Um diesen zeitaufwendigen Prozeß des Aus- und Einloggens abzukürzen, wurde der Befehl `su` entwickelt.

8.1.2 su

Das Kommando `su` erlaubt einen Wechsel der Benutzeridentität bzw. UID während einer Sitzung.

```
su [OPTIONEN] [BENUTZER]
```

Die einfache Eingabe von `su` ohne einen Benutzernamen bewirkt einen Wechsel zur Superuseridentität nach der Eingabe des Superuserkennworts.

Der Wechsel der Identität wird durch das Starten einer neuen Shell realisiert. Im Gegensatz zum Einloggen, bei dem auch eine Shell gestartet wird, bleibt beim Wechsel der Identität mit `su` die Umgebung des alten Benutzers erhalten. Nur die Umgebungsvariablen `HOME` und `SHELL` werden auf die entsprechenden Werte des neuen Benutzers geändert.

Spalte	Bezeichnung	Inhalt
1	Benutzername	Einzigartig um den Benutzer bei Einloggen zu identifizieren
2	Kennwort	Das verschlüsselte Kennwort (heute in `/etc/shadow`)
3	UID	Eindeutige Nummer zur Identifizierung des Benutzers
4	GID	Eindeutige Nummer zur Identifizierung der Gruppe
5	Kommentar	Voller Name des Benutzers und andere Angaben
6	Heimatverzeichnis	Arbeitsverzeichnis nach dem Einloggen
7	Login-Kommando	Das Kommando wird nach dem Einloggen ausgeführt; meist eine Shell

Tabelle 8.1: Die Felder der Datei `/etc/passwd`.

Jeder Benutzer kann seine Identität wechseln, solange er das Kennwort des neuen Benutzers kennt. Nur der Superuser kann seine Identität wechseln ohne ein Kennwort eingeben zu müssen.

Mit dem Befehl `exit` oder drücken der Tastenkombination `<STRG>+<D>` kehren Sie zur alten Identität zurück.

Optionen

`-, -l`	Die Konfigurationsdateien des Benutzers werden ausgeführt
`-c KOMMANDO`	Führt das angegebene Kommando unter dem neuen Benutzer aus

Beispiele

`su`
mit anschließender Kennworteingabe läßt den Benutzer nun als Superuser arbeiten.

`su tapico`
mit anschließender Kennworteingabe läßt den Benutzer nun unter der UID des Benutzers *tapico* arbeiten. Der Superuser benötigt keine Kennworteingabe um die Identität zu wechseln.

`su - tapico`
führt im Gegensatz zum obigen Beispiel dazu, daß die Konfigurationsdateien von *tapico* mit ausgeführt werden. Daher ist z. B. das Heimatverzeichnis von *tapico* jetzt das aktuelle Arbeitsverzeichnis.

8.2 Das Benutzerkonto

Als erstes nach der Installation des Systems, selbst wenn Sie der einzige Benutzer sind, sollten Sie sich einen eigenen einfachen Benutzer zulegen. Dazu müssen Sie für jeden anderen Benutzer ein eigenes Konto einrichten. Dies erhöht die Sicherheit im System, da so jeder Benutzer individuelle Privilegien erhält. Außerdem ist eine Verfolgung der Benutzeraktivitäten möglich.

8.2.1 /etc/passwd

Alle Informationen über das Benutzerkonto werden in der Datei `/etc/passwd` gespeichert. Es enthält u. a. den Loginnamen, die UID, das verschlüsselte Kennwort[1] und den vollen Namen. Ein Übersicht über den Inhalt gibt Tabelle 8.1.

Die Datei `/etc/passwd` gehört *root* mit dem Recht `rw-`. Alle anderen eingeschlossen die Gruppe haben nur das Lese-Recht.

Die Datei könnte z. B. folgende Einträge enthalten:

```
root:x:0:0:Herr und Meister:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:daemon:/sbin:/bin/bash
lp:x:4:7:lp daemon:/var/spool/lpd:/bin/bash
news:x:9:13:News system:/etc/news:/bin/bash
```

[1] Falls das Shadow-Paßwort-System installiert ist, was inzwischen bei den meisten Distributionen Standard ist, werden die Kennwörter nicht in `/etc/passwd` sondern in der getrennten Datei `/etc/shadow` gespeichert.

```
uucp:x:10:14:Unix-to-Unix CoPy system:/etc/uucp:/bin/bash
at:x:25:25::/var/spool/atjobs:/bin/bash
wwwrun:x:30:65534:Daemon user for apache:/tmp:/bin/bash
squid:x:31:65534:WWW proxy squid:/var/squid:/bin/bash
ftp:x:40:2:ftp account:/usr/local/ftp:/bin/bash
firewall:x:41:31:firewall account:/tmp:/bin/false
named:x:44:44:Nameserver Daemon:/var/named:/bin/bash
tapico:x:501:100:Thomas Prizzi:/home/tapico:/bin/bash
luke:x:502:100:Lukas Himmelsgeher:/home/luke:/bin/bash
lori:x:503:100:Lori Kalmar:/home/lori:/bin/bash
grayson:*:504:100:Grayson Death Carlyle:/home/grayson:/bin/bash
nobody:x:65534:65534:nobody:/tmp:/bin/bash
```

Die Felder werden durch Doppelpunkte (`:`) voneinander getrennt. Daher kann auch ein Feld, wie bei dem Benutzer *at* zu sehen, leer bleiben. Dieses Format wird bei fast allen UNIX-Systemen verwendet.

Feld: Benutzername

Das erste Feld ist der Benutzername. Er muß einmalig sein, da er für die Identifikation beim Einlogvorgang verwendet wird. Der Benutzername kann bis zu 8 Zeichen lang sein. Er kann aus Buchstaben, Ziffern, dem Unterstrich, dem Minuszeichen und anderen speziellen Zeichen bestehen. Bei Verwendung der nichtalphanumerischen Zeichen kann es zu Problemen mit Programmen kommen, die den Benutzernamen verwenden. Auch hier wird zwischen Groß- und Kleinschreibung unterschieden. So handelt es sich bei *tapico* und *Tapico* um zwei verschiedene Benutzer. Natürlich sollte die Anlage von zwei Benutzern, die sich nur in Groß- und Kleinschreibung unterscheiden, vermieden werden.

Wenn Sie viele Benutzer in ihrem System einrichten wollen, ist es wichtig sich eine Strategie zur Namensvergabe zu überlegen. In den meisten Fällen wird es eine Kombination aus Vorname und Nachname sein. So kann der Loginname der Benutzerin Angela Mahnke z. B. lauten: *mahnke*, *amahnke*, *aMahnke*, *A_Mahnke* oder *angelam*. Beim Benutzer Ole Vanhoefer wären es z. B.: *vanhoefe*, *ovanhoef*, *oVanhoef*, *o_Vanhoe* oder *olevan*.

Feld: Kennwort

Das nächste Feld im Datensatz ist das durchs Betriebssystem verschlüsselte Kennwort. Sollte der Eintrag beschädigt werden, so kann sich der Benutzer nicht mehr einloggen und der Administrator muß für den Benutzer ein neues Kennwort setzen.

Um einen Benutzer am Einloggen zu hindern, kann der Superuser hier ein Asterisk (`*`) einsetzen. Um einen Benutzer nur vorübergehend auszusperren, weil er im Urlaub ist oder den Superuser geärgert hat, ist es üblich an den Anfang des verschlüsselten Kennworts ein Asterisk zu setzen. Dabei ist darauf zu achten, daß die restlichen Zeichen des verschlüsselten Kennworts nicht verändert werden.

Ein leeres Feld an dieser Stelle bewirkt, daß Sie sich unter dem Benutzernamen ohne Eingabe eines Kennworts einloggen können. Solche Benutzerkonten sollten aber stark in ihren Rechten eingeschränkt werden, um nicht hier eine gewaltige Sicherheitslücke zu öffnen. Die Kennwortlänge beträgt bei den heutigen Distributionen zwischen fünf und acht Zeichen. Dies kann aber individuell konfiguriert werden.

Feld: UID

Das dritte Feld eines Eintrags in der `/etc/passwd` enthält die UID (*User IDentification*). Dies ist eine Zahl, mit der das System den Benutzer identifiziert. So wird beim Starten eines Prozesses durch einen Benutzer dessen UID dem Prozess zugeordnet.

Der Zahlenbereich kann den Typ eines Kontos schon einer bestimmten Gruppe zuordnen. UIDs unter 100 sind Systemkonten zugeordnet. Die UID 0 ist dem Superuser *root* vorbehalten. Normale Benutzer bekommen eine UID größer als 100 zugewiesen. Dabei ist die Grenze von Distribution zu Distribution unterschiedlich. Bei S.u.S.E. liegt die Grenze bei 500. In der Datei `/etc/login.defs` (Abschnitt 8.2.6) kann der Wertebereich der UID mit den Parametern `UID_MIN` und `UID_MAX` genau festgelegt werden.

Benutzerkonto	Einsatzgebiet
root	Superuser des Systems
daemon	Besitzt und setzt die Berechtigungen auf die Systemprozesse
bin	Besitzer der ausführbaren Dateien
sys	Besitzer der ausführbaren Dateien
adm	Besitzer der Log-Dateien
uucp	Benutzt für den UUCP-Zugriff

Tabelle 8.2: Besonder Benutzerkonten

Feld: GID

Die GID (*Group IDentification*) steht in Feld Nummer vier. Sie wird wie die UID vom System zur Verwaltung benutzt. Die Gruppen dienen zur besseren Organisation der Benutzer und deren Privilegien. Die hier angegebene Gruppe wird als Login-Gruppe des Benutzers bezeichnet. Die Gruppen werden in der Datei `/etc/group` (Abschnitt 8.4.1) definiert.

Wie auch bei der UID gibt die GID durch ihren Zahlenbereich schon Aufschluß über die Verwendung. Der Bereich 0 bis 49 ist den Systemgruppen vorbehalten. Die Nummern ab 50 bekommen die Benutzergruppen. Auch dies ist abhängig von der Distribution und wird ebenfalls in der Datei `/etc/login.defs` (Abschnitt 8.2.6) festgelegt. Als Standardgruppe für Benutzer werden meistens die Gruppe *group* oder *users* angelegt. Diesen Gruppen werden die Benutzer beim Erstellen standardmäßig zugeordnet.

Feld: Kommentar

Das Kommentarfeld kann leer gelassen werden. Normalerweise enthält es den vollen Namen des Benutzers. Es kann aber auch Abteilung, Telefonnummer, eMail oder sonstige Informationen enthalten. Der Befehl `finger` benutzt z. B. die hier enthaltenen Informationen. Auch nutzen einige eMail-Programme diesen Eintrag für den Absender.

Feld: Heimatverzeichnis

Dieser Eintrag bestimmt das Heimatverzeichnis des Benutzers. Hier findet er sich nach dem Einloggen wieder. Sollte der Eintrag ein nichtexistierendes Verzeichnis beinhalten, so schlägt der Einlogvorgang fehl.

Feld: Login-Kommando

In dem letzten Feld steht der Befehl, der nach dem Einloggen zuerst ausgeführt werden soll. Grundsätzlich handelt es sich hier um den Aufruf der zu benutzenden Shell. Dadurch kann der Benutzer in der Nutzung des System eingeschränkt werden. Ein leerer Eintrag führt dazu, daß die als Standard vorgegebene Shell benutzt wird.

Der Eintrag kann über den Befehl `chsh` (Abschnitt 4.1.1) geändert werden.

8.2.2 Besondere Benutzerkonten

Wenn Sie in die Datei `/etc/passwd` schauen, sehen Sie im Bereich der niedrigen UIDs die Systemkonten. Diese Konten sind virtuelle Benutzer unter dessen Identifikation die Systemprozesse laufen. Dies dient vor allen Dingen zur Sicherheit des Systems. Früher liefen die meisten Prozesse unter dem Konto *root*. Dies führte aber bei Sicherheitslücken dazu, daß der Angreifer unter der Identität von *root* arbeiten konnte. Wenn ihm heute ein Durchbruch gelingt, so kann er nur als einer der Systembenutzer arbeiten, die natürlich nur mit eingeschränkten Rechten versehen sind. Ein Liste solcher Systembenutzer zeigt Tabelle 8.2.

Neben diesen in der Tabelle aufgeführten Systembenutzer gibt es solche wie *mail*, *ftp*, *gopher*, *news*, *wwwroot* u.s.w. Denken Sie daran. Ändern Sie nie die Einträge für diese Besitzer, denn das kann verheerende Folgen für ihr System haben.

Vielleicht ist Ihnen auch aufgefallen, daß die Accounts im Kennwortfeld ein Asterisk (*) stehen haben. Es ist also nicht möglich sich interaktiv als Systembenutzer einloggen.

8.2.3 Benutzer am Einloggen hindern

Manchmal braucht man ein Benutzerkonto für eine bestimmte Aufgabe. Trotzdem soll sich keiner interaktiv mit diesem Konto auf dem Rechner einloggen können. Dies sind zum einen Systembenutzerkonten, zum anderen sind dies Konto, für die FTP- oder POP3-Zugriff möglich ist, aber eben kein direkter Shell-Login.

/bin/false

Wenn Sie sich die Datei `/etc/passwd` genauer anschauen, werden Sie bei vielen Systemkonten den Befehl `/bin/false` als Login-Shell eingetragen finden. Eigentlich ist `false` keine Shell, sondern ein Befehl der nichts tut und dann sich auch noch mit einem Statuscode beendet, der einen Fehler signalisiert. Das Ergebnis ist simpel. Der Benutzer loggt sich ein und sieht sofort wieder den Login-Prompt.

Bei Verwendung des Befehls `su` können Sie die Fehlermeldung besser sehen.

```
ole@enterprise:~> su martin
Password:
This account is currently not available.
ole@enterprise:~>
```

/sbin/nologin

Die Shell `/sbin/nologin` hat nur eine Aufgabe. Einem sich einloggenden Benutzer freundlich zu sagen, daß er sich nicht einloggen darf und ihn dann eiskalt aus dem System rauszuwerfen. Die übliche Nachricht wird durch den Inhalt der Datei `/etc/nologin.txt` ersetzt.

Beim normalen Login-Prompt auf der Konsole bekommen Sie nichts davon mit, da der neue Login-Prompt vorher den Bildschirm löscht. Versuchen Sie sich dagegen mit dem Befehl `su` einzuloggen, können Sie die Fehlermeldung gut sehen.

```
ole@enterprise:~> su martin
Password:
Hallo Fremder,
Du darfst Dich hier nicht einloggen.

Gruß Root
ole@enterprise:~>
```

Ohne die Datei `/etc/nologin.txt` ist die Fehlermeldung lange nicht so schön.

```
ole@enterprise:~> su martin
Password:
This account is currently not available.
ole@enterprise:~>
```

8.2.4 passwd

Um ein Kennwort zu ändern wird der Befehl `passwd` verwendet. Der Benutzer kann nur sein eigenes Kennwort ändern, während der allmächtige Superuser alle Kennwörter ändern kann.

```
passwd [BENUTZER]
```

Wenn ein Benutzer sein Kennwort ändern will, dann braucht er natürlich keinen Benutzernamen anzugeben. Er muß aber mit seinem alten Kennwort den Vorgang bestätigen und danach zweimal das neue Kennwort, was nicht auf dem Bildschirm angezeigt wird, eingeben. Dies soll eine Falscheingabe des neuen Kennworts durch Vertippen verhindern.

Die Richtlinien für die Kennwörter werden in der Datei `/etc/login.defs` (Abschnitt 8.2.6) festgelegt. Das kann die Mindestlänge des Kennworts, das Auslaufdatum des Kennworts und die Warnung davor sein. Außerdem finden Sie hier den Eintrag für das Verzeichnis der Mailboxen.

Bei der Kennwortvergabe ist darauf zu achten, daß gebräuchliche Worte aus Lexika (Brutal Force Attack) und Worte aus dem Umfeld der Person (scharf nachdenken) relativ leicht geknackt werden können. Die Kombination aus zwei Worten, Groß- und Kleinschreibung sowie Buchstaben und Zahlen, die mindestens eine Länge von sechs Zeichen aufweisen sollten, sind ein idealer Schutz.

8.2.5 chpasswd

Das Tool `chpasswd` erlaubt eine skriptähnliche Änderung von mehreren Kennwörtern gleichzeitig.

```
chpasswd [OPTIONEN]
```

Dabei liest chpasswd eine Datei von der Standardeingabe. Die Datei enthält pro Zeile den Benutzernamen und das Kennwort getrennt durch einen Doppelpunkt (`tapico:zorro67`). Das Kennwort steht dabei im Klartext. Ist das Kennwort bereits verschlüsselt, so muß der Schalter `-e` mitgegeben werden.

Beispiele

`cat newpass.txt | chpasswd`

Diese Kommandosequenz ändert die Kennwörter wie in der Datei `newpass.txt` beschrieben. Die Kennwörter stehen im Klartext in der Datei.

8.2.6 /etc/login.defs

Die Datei `/etc/login.defs` enthält die Einstellungen für den Login-Vorgang. Alle Einträge sind optional. Fehlen diese, so gelten sie als nicht gesetzt. Wie in den meisten Konfigurationsskripten üblich werden Kommentarzeilen (beginnen mit einem '`#`') und leere Zeilen ignoriert.

Hier ein Beispiel für die `/etc/login.defs`. Wie in guten Installationen üblich ergeben sich die Bedeutungen der Einträge aus den vorangestellten Kommentaren.

```
# Delay in seconds before being allowed another attempt after a login failure
#
FAIL_DELAY 3

# Enable logging and display of /var/log/faillog login failure info.
#
FAILLOG_ENAB yes

# Enable display of unknown usernames when login failures are recorded.
#
LOG_UNKFAIL_ENAB no

# Enable logging and display of /var/log/lastlog login time info.
#
LASTLOG_ENAB yes

# Enable additional checks upon password changes.
#
OBSCURE_CHECKS_ENAB yes

# If defined, ":" delimited list of "message of the day" files to
# be displayed upon login.
```

```
#
MOTD_FILE /etc/motd
#MOTD_FILE /etc/motd:/usr/lib/news/news-motd

# If defined, file which maps tty line to TERM environment parameter.
# Each line of the file is in a format something like "vt100  tty01".
#
TTYTYPE_FILE /etc/ttytype

# If defined, login failures will be logged here in a utmp format.
# last, when invoked as lastb, will read /var/log/btmp, so...
#
#FTMP_FILE /var/log/btmp

# If defined, file which inhibits all the usual chatter during the login
# sequence.  If a full pathname, then hushed mode will be enabled if the
# user's name or shell are found in the file.  If not a full pathname, then
# hushed mode will be enabled if the file exists in the user's home directory.
#
#HUSHLOGIN_FILE .hushlogin
HUSHLOGIN_FILE /etc/hushlogins

# The default PATH settings.
#
ENV_PATH                /usr/local/bin:/usr/bin:/bin

# The default PATH settings for root:
#
ENV_ROOTPATH            /sbin:/bin:/usr/sbin:/usr/bin

# Terminal permissions
#
# TTYGROUP Login tty will be assigned this group ownership.
# TTYPERM Login tty will be set to this permission.
#
# If you have a "write" program which is "setgid" to a special group
# which owns the terminals, define TTYGROUP to the group number and
# TTYPERM to 0620.  Otherwise leave TTYGROUP commented out and assign
# TTYPERM to either 622 or 600.
#
TTYGROUP tty
TTYPERM 0620

# Password aging controls:
#
# PASS_MAX_DAYS Maximum number of days a password may be used.
# PASS_MIN_DAYS Minimum number of days allowed between password changes.
# PASS_MIN_LEN Minimum acceptable password length.
# PASS_WARN_AGE Number of days warning given before a password expires.
#
PASS_MAX_DAYS 99999
PASS_MIN_DAYS 0
PASS_MIN_LEN 5
PASS_WARN_AGE 7

# If compiled with cracklib support, where are the dictionaries
#
CRACKLIB_DICTPATH /usr/lib/cracklib_dict

# Min/max values for automatic uid selection in useradd
#
UID_MIN   100
```

```
UID_MAX 60000

# Min/max values for automatic gid selection in groupadd
#
GID_MIN   100
GID_MAX 60000

# Max number of login retries if password is bad
#
LOGIN_RETRIES 3

# Max time in seconds for login
#
LOGIN_TIMEOUT 60

# Maximum number of attempts to change password if rejected (too easy)
#
PASS_CHANGE_TRIES 3

# Warn about weak passwords (but still allow them) if you are root.
#
PASS_ALWAYS_WARN yes

# Number of significant characters in the password for crypt().
# Default is 8, don't change unless your crypt() is better.
# Ignored if the "md5" option is given to the pam_pwcheck module.
#
PASS_MAX_LEN 8

# Require password before chfn/chsh can make any changes.
#
CHFN_AUTH yes

# Which fields may be changed by regular users using chfn - use
# any combination of letters "frwh" (full name, room number, work
# phone, home phone).  If not defined, no changes are allowed.
# For backward compatibility, "yes" = "rwh" and "no" = "frwh".
#
CHFN_RESTRICT rwh

# Should login be allowed if we can't cd to the home directory?
# Default is yes.
#
DEFAULT_HOME yes
```

8.3 Einrichten eines Benutzers

Damit ein Benutzer sich einloggen kann, benötigt er mindestens einen Benutzernamen und ein Heimatverzeichnis. Alles andere, sogar das Kennwort, werden aus anderen Gründen eingerichtet.

Es gibt mehrere Möglichkeiten einen Benutzer anzulegen. Sie werden hier die direkte Methode und den Befehl `useradd` kennenlernen. Viele Distributionen bieten weitere, meist komfortablere Tools an.

8.3.1 Bearbeiten der Datei `/etc/passwd`

Der ursprünglichste Weg (Hardcore-Administration) einen Benutzer einzurichten ist der direkte Eingriff in die Datei `/etc/passwd`. Bevor Sie aber dies tun, sollten Sie sich auf jeden Fall eine Sicherungskopie der Datei anlegen, denn ohne eine funktionstüchtige `passwd`-Datei können Sie sich selbst als *root* nicht mehr einloggen.

Um jetzt einen neuen Benutzer anzulegen, müssen Sie nur eine neue Zeile mit den entsprechenden Informa-

tionen hinzufügen. Denken Sie bitte daran, daß der Editor die Datei in reinem ASCII-Code (z. B. `vi` oder `joe`) abspeichert. Sie tragen ins erste Feld den Lognamen des neuen Benutzers ein. Das zweite Feld mit dem Kennwort lassen Sie leer, da es hier in verschlüsselter Form stehen muß. Für das dritte Feld müssen Sie sich erst eine Übersicht über die vorhandenen UIDs verschaffen, damit Sie dann die nächste freie UID hier eintragen können. Ein solcher Eintrag könnte dann lauten:
`tapico::523:100:Thomas Prizzi:/home/tapico:/bin/bash`

Nach dem Eintrag und dem Abspeichern der Datei müssen Sie den Befehl `passwd` verwenden um das Kennwort des Benutzers festzulegen.

Jetzt sollte das Heimatverzeichnis des Benutzers angelegt und die Konfigurationsdateien kopiert werden. Auf diesen Vorgang gehen wir später genauer ein. Denken Sie daran den Benutzer als neuen Besitzer und seine Gruppe als Gruppe für das Heimatverzeichnis einzutragen. Dazu können Sie das `chown`-Kommando (Abschnitt 9.2.1) benutzen.

8.3.2 useradd

Ein komfortablerer und weniger fehleranfälliger Weg einen Benutzer einzurichten ist der Befehl `useradd`.

`useradd [OPTIONEN] [BENUTZER]`

Optionen	
`-D`	Zeigt die Standardeinstellung des Systems
`-u UID`	Erlaubt eine UID vorzugeben
`-g GID`	Erlaubt eine GID vorzugeben
`-o`	Erlaubt die mehrfache Verwendung von UIDs
`-m`	Das Heimatverzeichnis wird automatisch erstellt
`-d VERZEICHNIS`	Angegebenes Verzeichnis als Heimatverzeichnis verwenden
`-k VERZEICHNIS`	Angegebenes Verzeichnis als Skeleton-Verzeichnis verwenden
`-c KOMMENTAR`	Inhalt des Kommentarfeldes
`-s SHELL`	Name der LOGIN-Shell
`-e MM/DD/YY`	Verfallsdatum des Kontos
`-f TAGE`	Zeitraum zwischen Auslaufen des Paßworts und schließen des Kontos

Im Normalfall benutzt `useradd` die nächste freie UID und weist dem neuen Benutzer die aktuelle Shell als Login-Kommando des Kontos zu. Ein Heimatverzeichns wird zwar in der `/etc/passwd` eingetragen, aber nicht erstellt. Erst der Schalter `-m` bewirkt das Anlegen des Heimatverzeichnis. Dies ist normalerweise `/home/BENUTZERNAME`. Durch den Schalter `-d` kann aber auch ein anderes Verzeichnis angegeben werden. Wenn das Heimatverzeichnis angelegt wird, werden automatisch die Dateien aus dem Skeleton-Verzeichnis (meistens `/etc/skel`) in das neue Heimatverzeichnis kopiert. Der Befehl `useradd` sorgt auch dafür, daß für alle angelegten Verzeichnisse und Dateien die richtigen Besitzer, Gruppen und Rechte eingetragen werden.

Mit dem Schalter `-e MM/DD/YY` kann ein Verfallsdatum für das Konto angegeben werden. Nach dem Auslaufen des Kennworts kann ein Konto auch gesperrt werden. Der Schalter `-f TAGE` gibt dabei an, wann dies erfolgen soll. Eine Null steht dabei für sofort, während `-1` dafür steht, daß das Konto nicht gesperrt wird, wenn das Kennwort abgelaufen ist.

Beispiele
Die Verwendung des Schalters `-D` ohne Angabe eines Benutzernamens zeigt die Standardeinstellungen des Systems.

```
root@defiant:/home > useradd -D
GROUP=100
HOME=/home
INACTIVE=0
EXPIRE=10000
SHELL=/bin/bash
SKEL=/etc/skel
```

`useradd -m luke`
legt den Benutzer *luke* zusammen mit seinem Heimatverzeichnis an.

8.3.3 usermod

Nachdem Sie einen Benutzer angelegt haben, können Sie mit dem Befehl `usermod` alle Felder in der `/etc/passwd` ändern.

```
usermod [OPTIONEN] BENUTZER
```

Der Benutzername kann nicht geändert werden, wenn der Benutzer gerade eingeloggt ist. Wird der Benutzername geändert, dann sollten Sie auch das Heimatverzeichnis entsprechend umbenennen. Dies ist allerdings nicht notwendig, denn solange das Heimatverzeichnis existiert, kann sich der Benutzer einloggen.

Optionen

`-l BENUTZERNAME`	Neuer Benutzername
`-u UID`	Neue UID
`-g GID`	Neue GID
`-c "KOMMENTAR"`	Neuer Kommentareintrag
`-d VERZEICHNIS`	Neues Heimatverzeichnis
`-s SHELL`	Neue Login-Shell
`-o`	Erlaubt mehrfache Nutzung von UIDs
`-m`	Neues Heimatverzeichnis wird angelegt und alte Dateien rüberkopiert

Wird die UID des Benutzers geändert, dann erhalten auch alle Dateien des Benutzers in seinem Heimatverzeichnis die neue UID zugeordnet. Dateien außerhalb des Heimatverzeichnisses werden nicht geändert. Auch die `crontab`- und `at`-Jobs müssen manuell geändert werden.

8.3.4 Entfernen eines Benutzers

Wenn ein Benutzer aus dem System entfernt werden soll, muß nicht nur sein Eintrag in der `/etc/passwd` gelöscht werden, sondern auch seine Hinterlassenschaften entweder gelöscht oder anderen Benutzern übertragen werden. Bei einer manuellen Entfernung sollten Sie folgenden Schritte ausführen:

1. Entfernen Sie die Zeile des Benutzers aus der `/etc/passwd`.

2. Löschen Sie das Heimatverzeichnis des Benutzers.

3. Suchen Sie nach Dateien des Benutzers, die sich außerhalb seines Heimatverzeichnis befinden und löschen Sie diese.

4. Löschen Sie die Mail-Dateien des Benutzers und jedes Mail-Alias

5. Entfernen Sie jeden Job (`crontab` oder `at`), die der Benutzer eingerichtet hat.

Ein leichterer Weg einen Benutzer zu löschen ist der Befehl `userdel`.

8.3.5 userdel

Der Befehl `userdel` dient dazu einen Benutzer zu entfernen.

```
userdel [OPTIONEN] BENUTZER
```

Optionen

`-r`	Löscht auch das Heimatverzeichnis des Benutzers

Die Ausführung des Befehls ohne Optionen löscht den Benutzer aus der Datei `/etc/passwd` und aus anderen Systemdateien, enfernt aber nicht sein Heimatverzeichnis. Dies hingegen bewirkt der Schalter `-r`. Dateien des Benutzers, die sich außerhalb seines Heimatverzeichnis befinden, werden nicht gelöscht.

8.3.6 Einrichten eines zweiten Superusers

Wie schon am Anfang des Kapitels erwähnt, ist nur der Benutzer mit der UID 0 Gott im System und darf alles. Wenn Sie nun das Paßwort von dem Benutzer vergessen oder unabsichtlich verstellen, was manchen meiner Teilnehmer schon mal passiert ist, ist das System nicht mehr administrierbar. Für diesen Fall sollte man sich einen zweiten (oder auch dritten) Superuser einrichten. Wie bekommt aber dieser nun die Rechte des Superusers. Das Einstellen in die Gruppe *root* bringt da nicht viel.

Die Lösung liegt auf der Hand. Dem zweiten Superuser wird auch die UID 0 zugewiesen. D. h. es gibt zwei Benutzer mit der gleichen UID. Eigentlich gibt es nur einen Benutzer, der sich aber unter zwei verschiedenen Namen einloggen kann. Eigentlich dürfen keine zwei Benutzer die gleiche UID verwenden. Wenn Sie mit `useradd` den zweiten Superuser einrichten wollen, dann wird dies mit einer Fehlermeldung quittiert. Erst die Verwendung des Schalters `-o` ermöglicht eine doppelte Vergabe von UIDs.

```
root@defiant:~> useradd -u 0 -g 0 -m -c "2. Superuser" admin
useradd: uid 0 is not unique
root@defiant:~> useradd -u 0 -g 0 -o -m -c "2. Superuser" admin
root@defiant:~> passwd admin
New password:
Re-enter new password:
Password changed
```

In Ihrer Firma sollten zumindest drei Superuserkonten vorhanden sein. `root` oder das umbenannte Konto für Sie, eins für Ihren Stellvertreter und eins, dessen Daten in einem versiegelten Umschlag im Tresor aufbewahrt werden. Denn wenn Ihrem Stellvertreter etwas zustößt, während Sie gerade in Ihrem Urlaub das australische Outback erforschen, ist das Firmennetz ohne zugänglichem Superuserkonto nur noch schwer administrierbar.

8.4 Gruppen

Gruppen werden dazu verwendet um die Benutzer und ihre Rechte zu organisieren. Gruppen werden erstellt um bestimmten Benutzern Rechte auf ausgewählte Verzeichnisse oder Geräte zu erteilen.

Wenn ein Benutzer auf eine Datei zugreifen will, dann prüft das Betriebssystem, ob er der Besitzer der Datei ist. Ist er es nicht, dann prüft es, ob er der Datei zugewiesenen Gruppe angehört. Trifft das zu, dann erhält er den hier definierten Zugang zur Datei. Gehört der Benutzer auch nicht zur Gruppe, dann werden ihm nur die Privilegien eingeräumt, die auch alle anderen bekommen.

Die Benutzer können Mitglieder vieler Gruppen sein. Es ist aber nur immer eine Gruppe mit dem Benutzer assoziiert. Diese Gruppe ist in der Datei `/etc/passwd` festgelegt.

Um eine Gruppe einzurichten kann entweder die Datei `/etc/group` (Abschnitt 8.4.1) um einen Eintrag erweitert oder der Befehl `groupadd` (Abschnitt 8.4.7) verwendet werden. Änderungen der Gruppeneigenschaften werden entweder in der Konfigurationsdatei selber ausgeführt oder über den Befehl `groupmod` (Abschnitt 8.4.9). Das Entfernen der Gruppe erfolgt entweder wieder über die Konfigurationsdatei oder über den Befehl `groupdel` (Abschnitt 8.4.11). Bei der Verwendung des Shadow-Paßwort-Systems (Abschnitt 8.5) sollten möglichst nur diese Tools verwendet werden.

8.4.1 /etc/group

Die Datei `/etc/group` ist die Konfigurationsdatei für die Gruppe und ihre Mitglieder. Sie ist ähnlich wie die Datei `/etc/passwd` aufgebaut.

```
root:*:0:root
bin:*:1:root,bin,daemon
daemon:*:2:
sys:*:3:
tty:*:5:
wwwadmin:*:8:
```

Systemgruppe	Einsatzgebiet
root	Ist den meisten Systemdateien zugeordnet.
daemon	Ist der Mail, den Druckern und anderer Systemsoftware zugeordnet.
kmem	Verwaltet den direkten Zugriff zum Kernelspeicher.
sys	Ist einigen Systemdateien, Swap-Dateien und Speicherabbildungen zugeordnet.
nobody	Ist der Software zugeordnet, die keine speziellen Rechte braucht.
tty	Ist den Gerätedateien zugeordnet, die die Terminals kontrollieren.

Tabelle 8.3: Systemgruppen

```
mail:*:12:
news:*:13:news
uucp:*:14:uucp,fax,root,ole
shadow:*:15:root
dialout:*:16:root,ole,tapico,grayson
at:*:25:at
firewall:*:31:
public:*:32:
named:*:44:named
users:*:100:
greydeath:*:101:root,grayson
nogroup:*:65534:root
```

Die durch Doppelpunkte abgetrennten Felder sind von links nach rechts Gruppenname, Gruppenkennwort, GID und die Mitgliederliste.

Feld: Gruppenname

In dem ersten Feld steht der Gruppenname. Er muß einmalig sein und darf bis zu acht Buchstaben lang sein. Für die Wahl der Zeichen gelten die gleichen Regeln wie bei den Benutzernamen.

Feld: Gruppenkennwort

Das zweite Feld ist für das Gruppenkennwort vorgesehen. Dieses Kennwort ist für den Wechsel in eine Gruppe gedacht. Normalerweise ist dieses Feld leer oder enthält nur einen Asterisk (*).

Feld: GID

Die Group Identification wird vom System benutzt und muß einmalig sein. Die Identifikation der Gruppe vom System her erfolgt nur über diese Nummer.

Feld: Mitgliederliste

Im letzten Feld werden die Mitglieder (Benutzer) der Gruppe eingetragen. Dabei werden die einzelnen Einträge durch Semikola getrennt.

8.4.2 Besondere Gruppen

Vom System werden einige Systemgruppen wie z. B. *bin*, *mail*, *sys* oder *adm* angelegt. Sie sollten auf keinem Fall einen Benutzer zu den Gruppen hinzufügen, da er sonst weitreichenden Rechte bekommen würde. Diese Gruppen sollten nur Systemkonten (Abschnitt 8.2.2) als Mitglieder enthalten. Eine Übersicht über die wichtigsten Systemgruppen liefert Tabelle 8.3.

8.4.3 id

Um herauszufinden welche GID einem Benutzer zugeordnet ist, wird der Befehl `id` verwendet.

```
id [OPTIONEN] [BENUTZERNAME]
```

Optionen	
`-g`	Gibt nur die Haupt-GID aus
`-G`	Zeigt nur die GIDs aller Gruppen
`-n`	Gibt die Namen aus (für -u, -g und -G)
`-r`	Gibt die reale ID aus anstatt der effektiven ID (für -u, -g und -G)
`-u`	Gibt nur die UID aus

Der Befehl zeigt die UID und die mit dem Benutzer assoziierte GID an. Zusätzlich werden auch die Gruppen angezeigt, in denen der Benutzer Mitglied ist.

Beispiel

```
tapico@defiant:~ > id
uid=501(tapico) gid=100(users) Gruppen=100(users),101(greydeath)
tapico@defiant:~ > groups
users greydeath defiant
```

8.4.4 groups

Der Befehl `groups` entspricht dem Befehl `id -Gn` und zeigt die Namen der Gruppen an, in denen der Benutzer Mitglied ist.

```
groups [BENUTZER]
```

8.4.5 newgrp

Um die assoziierte Gruppe zu wechseln, wird der Befehl `newgrp` benutzt.

```
newgrp [NEUEGRUPPE]
```

Die Angabe des Befehls ohne einen Gruppennamen, läßt die GID zur in der `/etc/passwd` eingetragenen Gruppe zurückwechseln.

Beispiel

```
tapico@defiant:~ > id
uid=501(tapico) gid=100(users) Gruppen=100(users),101(greydeath),102(defiant)
tapico@defiant:~ > newgrp defiant
tapico@defiant:~ > id
uid=501(tapico) gid=102(defiant) Gruppen=100(users),101(greydeath),102(defiant)
```

8.4.6 Anlegen von Gruppen

Sie können Gruppen auf zwei Arten anlegen. Entweder Sie editiern die Datei `/etc/group` direkt oder Sie benutzen ein Tool wie `groupadd`. Im Zusammenhang mit dem Shadow-Paßwort-System ist die Benutzung des Tools vorzuziehen, da sonst auch manuelle Eintragungen in der Datei `/etc/gshadow` nötig sind.

8.4.7 groupadd

Der Befehl `groupadd` legt eine neue Gruppe an.

```
groupadd [OPTIONEN] GRUPPENNAME
```

Die Gruppe wird mit einem leeren Gruppenkennwort eingerichtet und enthält noch keine Mitglieder. Die können mit dem Befehl `gpasswd` hinzugefügt werden.

Optionen	
`-g GID`	GID für die neue Gruppe vorgeben
`-o`	Mehrfache Nutzung der GID zulassen
`-r`	Systemgruppen einrichten (nicht S.u.S.E)

8.4.8 gpasswd

Das Tool `gpasswd` wird zur Mitglieder- und Kennwortverwaltung der Gruppen benutzt.

```
gpasswd [OPTIONEN] GRUPPE
```

Optionen	
`-R`	Verhindert die Benutzung des Befehls `newgrp` für diese Gruppe
`-a BENUTZER`	Fügt neues Mitglied hinzu
`-d BENUTZER`	Entfernt eine Mitglied aus der Gruppe
`-A BENUTZER`	Legt das Mitglied als Gruppenadministrator fest
`-r`	Entfernt das Kennwort der Gruppe

Beispiel

```
root@defiant:/ > groupadd defiant
root@defiant:/ > gpasswd -a tapico defiant
Adding user tapico to group defiant
root@defiant:/ > id tapico
uid=501(tapico) gid=100(users) Gruppen=100(users),101(greydeath),102(defiant)
root@defiant:/ > grep defiant /etc/group
defiant:x:102:tapico
```

8.4.9 groupmod

Um die Eigenschaften einer Gruppe zu ändern wird der Befehl `groupmod` verwendet.

```
groupmod [OPTIONEN] GRUPPE
```

Optionen	
`-n NAME`	Neuer Gruppenname
`-g GID`	Neue GID
`-o`	Mehrfache Nutzung einer GID zulassen

Bei der Änderung der GID sollte die `/etc/passwd` überprüft werden, ob dort nicht die alte GID einem Benutzer zugeordnet wurde. Diese muß natürlich dann auf die neue GID geändert werden.

8.4.10 Löschen einer Gruppe

Um eine Gruppe zu löschen, entfernen Sie einfach die entsprechende Zeile aus der `/etc/group`. Davor sollten Sie noch folgenden Schritte durchführen.

1. Prüfen Sie die Datei `/etc/passwd` und vergewissern Sie sich, daß kein Benutzer die GID der gelöschten Datei als Login-Gruppe verwendet. Sollte dies der Fall sein, so ändern Sie die GID auf eine existierende Gruppe. Tun Sie das nicht, dann kann der Benutzer sich nicht mehr einloggen.
2. Prüfen Sie im Dateisystem nach, ob Verzeichnisse oder Dateien mit der gelöschten GID assoziiert sind. Dies könnte den Zugriff auf die Dateien erschweren oder sogar gänzlich unmöglich machen.
3. Prüfen Sie, ob nicht die Gruppe eine Systemgruppe ist. Das Entfernen einer solchen Gruppe könnte zu erheblichen Schäden am System führen.

8.4.11 groupdel

Einfacher geht das Löschen einer Gruppe mit dem Befehl `groupdel`.

```
groupdel GRUPPE
```

Sollten Sie diesen Befehl auf eine Gruppe anwenden, die einem Benutzer als Login-Gruppe zugewiesen ist, wird der Befehl nicht ausgeführt.

Auch wie beim manuellen Löschen müssen Sie sich selber davon überzeugen, daß keine Verzeichnisse oder Dateien noch mit der gelöschten GID assoziiert sind.

8.5 Shadow-Paßwort-System

Die Speicherung des Kennwort, wenn auch in einer verschlüsselten Form, in einer von allen lesbaren Datei ist ein Sicherheitsrisiko. Der Einsatz von Entschlüsselungsprogrammen ermöglicht es relativ leicht an die Kennwörter zu gelangen. Deswegen ist der Einsatz des Shadow-Paßwort-Systems bei den heutigen Linux-Distributionen üblich. In der `/etc/passwd` steht für das Kennwort nur noch ein ‘x’. Die Kennwörter stehen in der nur für *root* zugänglichen `/etc/shadow`.

8.5.1 /etc/shadow

Die `/etc/shadow` enthält nicht nur den Namen und das Kennwort des Benutzers, sondern noch weitergehende Informationen zum Konto.

```
root@defiant:/home/ole > grep tapico /etc/shadow
tapico:9Fd4SU/BbEP52:11317:0:10000::::
```

Die Felder von links nach rechts sind:

1. Der Benutzername
2. Das verschlüsselte Kennwort.
3. Die Anzahl von Tagen zwischen dem 01.01.1970 und der letzten Kennwortänderung.
4. Die Zeit in Tagen, die zwischen zwei Kennwortänderungen liegen muß.
5. Die Zeit in Tagen, wie lange ein Kennwort gültig ist.
6. Die Zeit in Tagen, wie lange der Benutzer vor dem Auslaufen des Kennworts gewarnt wird.
7. Die Zeit in Tagen bis das Konto nach dem Auslaufen des Kennworts gesperrt wird.
8. Auslaufen des Kontos in Tagen seit dem 01.01.1970.

Im Gegensatz zur `/etc/passwd` darf das Kennwortfeld nicht leer sein. Um einen Benutzer daran zu hindern sein Kennwort zu ändern, brauchen Sie nur das minimale Kennwortalter größer zu machen als das maximale Kennwortalter.

8.5.2 /etc/gshadow

Genau wie das Kennwort kann auch das Gruppenkennwort durch das Shadow-Paßwort-System besser geschützt werden. Hier ist die Datei `/etc/gshadow` dafür verantwortlich. Ist das System eingerichtet, wird das Kennwortfeld in der `/etc/group` durch ein 'x' ersetzt und das Kennwort in der `/etc/gshadow` eingetragen. Auch auf die `/etc/gshadow` sollte nur *root* Zugriffsrechte besitzen.

Sie sollten auf keinen Fall die Dateien `/etc/shadow` oder `/etc/gshadow` selbst editieren. Benutzen Sie lieber dafür die vorgesehen Tools wie `useradd`, `usermod` oder `gpasswd`.

8.5.3 pwconv

Um das Shadow-Paßwort-System zu nutzen, müssen Sie die Kennwörter aus der `passwd`-Datei in eine neue Datei umgewandelt werden. Der Befehl `pwconv` erzeugt aus Ihrer `/etc/passwd` eine `/etc/shadow`-Datei und ersetzt die Kennwörter durch 'x'.

```
pwconv
```

Um diesen Befehl erfolgreich ausführen zu können, darf kein Paßwortfeld in der `/etc/passwd` *leer sein.*

8.5.4 pwunconv

Um das Shadow-Paßwort-System zu deaktivieren, wird der Befehl `pwunconv` verwendet.

```
pwunconv
```

Der Befehl bringt die Datei `/etc/passwd` auf den neuesten Stand und löscht die `/etc/shadow`.

8.5.5 grpconv

Was `pwconv` für die `/etc/passwd` ist `grpconv` für `/etc/group`.

```
grpconv
```

8.5.6 grpunconv

Wie bei den Benutzern kann auch das verbesserte System zu Kennwortspeicherung auch bei den Gruppen rückgängig gemacht werden.

```
grpunconv
```

Wenn Sie die Dateien `/etc/passwd` und `/etc/group` per Hand editieren müssen, ist es oft notwendig, daß Sie die Shadow-Dateien updaten. Dazu können Sie die Befehle `pwconv` und `grpconv` verwenden. In der Regel ist es am besten erst mit `pwunconv` und `grpunconv` das Shadow-Paßwort-System abzuschalten, die Änderungen vorzunehmen und dann mit `pwconv` und `grpconv` wieder zu aktivieren.

8.6 Startdateien des Benutzers

Das Aussehen und die Funktion der Shell ist durch die Umgebungsvariablen festgelegt. Diese Parameter sind in verschiedenen Konfigurationsdateien im Verzeichnis `/etc` und im Heimatverzeichnis des Benutzers festgelegt bzw. werden bei jedem Start und Einloggen definiert.

8.6.1 /etc/profile

Die Standardeinträge für die Umgebungsvariablen für alle Benutzer, die sich interaktiv einloggen, befinden sich in der `/etc/profile`. Hier das Beispiel für eine solche Datei.

```
# /etc/profile

PROFILEREAD=true

umask 022

# adjust some limits (see bash(1))
ulimit -Sc 0 # don't create core files
ulimit -d unlimited
fi
ulimit -m unlimited

# make path more comfortable
#
MACHINE=`test -x /bin/uname && /bin/uname --machine`
PATH=/usr/local/bin:/usr/bin:/usr/X11R6/bin:/bin
test "$UID" = 0 || PATH="$PATH:."
export PATH

# set some environment variables
#
if test -n "$TEXINPUTS" ; then
  TEXINPUTS=":$TEXINPUTS:~/.TeX:/usr/doc/.TeX"
else
  TEXINPUTS=":~/.TeX:/usr/doc/.TeX"
fi
export TEXINPUTS
PRINTER='lp'
export PRINTER

LESSCHARSET=latin1
export LESSCHARSET
LESS="-M -S -I"
export LESS
LESSKEY=/etc/lesskey.bin
export LESSKEY
LESSOPEN="|lesspipe.sh %s"
export LESSOPEN

MANPATH=/usr/local/man:/usr/share/man:/usr/man:/usr/X11R6/man
export MANPATH

# set INFOPATH to tell xemacs where he can find the info files
#
INFODIR=/usr/local/info:/usr/share/info:/usr/info
INFOPATH=$INFODIR
export INFODIR INFOPATH

# do not save dupes in the bash history file
HISTCONTROL=ignoredups
export HISTCONTROL
```

```
alias dir='ls -l'
alias la='ls -la'
alias l='ls -alF'
alias ls-l='ls -l'
alias o='less'
alias ..='cd ..'
alias ...='cd ../..'
alias rd=rmdir
alias md='mkdir -p'
alias beep='echo -en "\x07"'
alias unmount='echo "Error: Try the command: umount"'

function startx { /usr/X11R6/bin/startx $* 2>&1 | tee ~/.X.err ; }

PAGER=less
export PAGER

if [ "$TERM" = "" -o "$TERM" = "unknown" ]; then
 TERM=linux
fi

# set prompt
#
set -p
if test "$UID" = 0 ; then
    PS1="\h:\w # "
else
    PS1="\u@\h:\w > "
fi
PS2='> '
ignoreeof=0
export PS1 PS2 ignoreeof
```

Um systemweite Änderungen in den Umgebungen der Benutzer durchzuführen, sollten Sie diese Datei editieren. S.u.S.E.-Linux erstellt die `/etc/profile` selber. Für die Änderungen des Administrators ist die `/etc/profile.local` vorgesehen, die aus der `profile` aufgerufen wird.

Sie können die `bash` auch aufrufen, ohne daß die `/etc/profile` abgearbeitet wird. Dies bewirkt der Schalter `--noprofile`.

Wenn der Benutzer sein eigenes Umfeld gestalten will, kann er das in der Datei `.profile` machen, die in seinem Heimatverzeichnis liegt. Wenn ein Benutzer sich einloggt, dann liest das Betriebssystem erst die `/etc/profile` aus. Dann sucht es nach den Dateien `.bash_profile`, `.bash_login` und `.profile` im Heimatverzeichnis des Benutzers.

Beim Ausloggen aus einer interaktiven Shell wird, wenn vorhanden, die Datei `.bash_logout` im Heimatverzeichnis ausgeführt.

8.6.2 /etc/bashrc

Wenn die `bash` eine interactive Shell aufruft, die nicht die Login-Shell ist, wie es z. B. bei der Eingabe des Befehls `bash` vorkommt, dann wird die Datei `/etc/bashrc` abgearbeitet anstatt die `/etc/profile`.

Durch die Option `-norc` kann die Abarbeitung der Konfigurationsdatei verhindert werden. Der Schalter `-rcfile DATEINAME` ermöglicht sogar die Angabe einer alternativen Konfigurationsdatei.

Kommt es zum Aufruf einer nicht-interactiven Shell, z. B. wenn ein Shell-Skript ausgeführt wird, dann schaut das Betriebssystem erst einmal in der Umgebungsvariablen `BASH_ENV` nach, ob dort der Name einer Konfigurationsdatei steht. Ansonsten wird dann auch die `/etc/bashrc` ausgeführt.

Auch diese Konfigurationsdatei gibt es als individuelle Version für den Benutzer als `~/.bashrc`

8.6.3 sh

Der Befehl `sh` ist im Prinzip nur ein anderer Namen für den Befehl `bash`. Allerdings wertet er beim Aufruf die Umgebungsvariable `ENV` aus. Dort steht der Name der Konfigurationsdatei, die ausgeführt werden soll. Andere Konfigurationsdateien werden im Gegensatz zu `bash` nicht ausgeführt.

8.7 Das "Who-is-Who" der Benutzer

8.7.1 who

Wer ist denn zur Zeit eingeloggt? Diese Frage stellt sich dem Administrator immer dann, wenn er den Rechner herunterfahren will. Aber auch da kann Linux weiterhelfen. Der Befehl `who` zeigt alle eingeloggten Benutzer mit ihren Terminals an.

```
who [OPTIONEN]
```

Optionen	
`-H`	Zeigt eine Überschriftenzeile
`-i`, `-u`	Zeigt an, wie lange ein Benutzer nicht mehr gearbeitet hat
`-m`	Zeigt den aktuellen Benutzer an
`-q`	Zeigt die eingeloggten Benutzer und ihre Anzahl an

Für die Ausgabe des Befehls wertet er die Dateien `/var/run/utmp` und `/var/log/wtmp` aus.

Der Befehl kann aber auch auf eines der wichtigsten Probleme im Leben eines Administrators eine Antwort geben: "Wer bin ich?". Einfach `who am I` eingeben und die Antwort ist da. Wer es liebevoller mag, dem sei die Version `who mom likes` an Herz gelegt. Im Endeffekt sind aber diese Erweiterungen nur andere Namen für `who -m`.

Beispiel

```
tapico@defiant:~ > who
tapico   tty1     Feb 10 13:26
tapico   tty3     Feb 10 15:28
tapico   tty4     Feb 10 15:26
grayson  pts/0    Feb 10 14:05
tapico@defiant:~ > who -Hi
BENU     LEIT     LOGIN-ZEIT   RUHIG VON
tapico   tty1     Feb 10 13:26 05:29
tapico   tty3     Feb 10 15:28 00:15
tapico   tty4     Feb 10 15:26 00:30
grayson  pts/0    Feb 10 14:05   .
root     pts/1    Feb 10 18:49 00:05
```

8.7.2 whoami

Der Befehl `whoami` sagt zwar auch, wer ich gerade bin. Aber er macht das wesentlich kürzer.

```
whoami
```

8.7.3 logname

Der Befehl `logname` erzählt dem Benutzer unter welchem Namen er sich ursprünglich mal eingeloggt hat.

```
logname
```

Frisch nach dem Einloggen zeigen die Befehle `whoami` und `logname` keine Unterschiede. Erst wenn ein Identitätswechsel vollzogen wurde, werden die Unterschiede der beiden Befehle deutlich.

```
ovidio@defiant:~ > whoami
ovidio
ovidio@defiant:~ > logname
ovidio
ovidio@defiant:~ > su tapico
Password:
tapico@defiant:/home/ovidio > whoami
tapico
tapico@defiant:/home/ovidio > logname
ovidio
```

8.7.4 w

Der Befehl `w` zeigt wie `who` die eingeloggten Benutzer an. Er liefert allerdings weitergehende Informationen wie Einlogzeitpunkt, CPU-Nutzung und was der Benutzer gerade ausführt.

```
w [OPTIIONEN]
```

Beispiel

```
tapico@defiant:~ > w
  4:23pm  up  2:00,  6 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU  WHAT
tapico   tty1     -                2:24pm  1:58m 44.56s  0.04s  sh /usr/X11R6/bin/startx
root     tty2     -                4:22pm 22.00s  0.12s  0.06s  -bash
lori     tty3     -                4:23pm 14.00s  0.08s  0.05s  -bash
tapico   tty5     -                4:21pm  1:33   0.08s  0.04s  -bash
tapico   pts/0    -                4:21pm  0.00s  0.08s  0.03s  w
tapico   pts/1    -                3:37pm  2:05   2.32s  0.10s  bash
```

8.7.5 finger

Auch der Befehl `finger` zeigt die eingeloggten Benutzer an. Dabei kann er eine Übersicht über die Benutzer liefern oder ausführliche Informationen über einzelne Benutzer.

```
finger [OPTIONEN] [BENUTZER]
```

Der Befehl `finger` wertet für seine Ausgabe die Datei `/etc/passwd` sowie die Dateien `.plan` und `.project` im Heimatverzeichnis des jeweiligen Benutzers aus. Diese beiden Dateien geben Auskunft über zukünftige Pläne und Projekte.

Beispiele

```
lori@defiant:~ > finger
Login     Name                 Tty      Idle  Login Time   Where
lori      Lori Kalmar          3           5     Sun 16:23
root      Herr und Meister     2           5     Sun 16:22
tapico    Thomas Prizzi        1        2:04     Sun 14:24
tapico    Thomas Prizzi        pts/0       -     Sun 16:21
tapico    Thomas Prizzi        pts/1       7     Sun 15:37
tapico    Thomas Prizzi        5           6     Sun 16:21
lori@defiant:~ > finger tapico
```

```
Login: tapico                         Name: Thomas Prizzi
Directory: /home/tapico               Shell: /bin/bash
On since Sun Feb 11 16:21 (MET) on tty5, idle 0:12
No Mail.
Plan:
Bin vom 15. bis 17. Februar nicht in der Firma.
```

8.7.6 last

Der Befehl `last` zeigt die Benutzer an, die sich zuletzt eingeloggt haben. Er durchsucht dafür die Datei `/var/log/wtmp`.

```
last [OPTIONEN] [BENUTZER]
```

Daneben zeigt er auch an, seit wann die Datei `/var/log/wtmp` Informationen enthält.

```
tapico@defiant:~ > last -10
lori     tty3                        Sun Feb 11 16:23   still logged in
root     tty2                        Sun Feb 11 16:22   still logged in
tapico   tty5                        Sun Feb 11 16:21   still logged in
lori     pts/0                       Sun Feb 11 16:21   still logged in
tapico   pts/1                       Sun Feb 11 15:37   still logged in
tapico   pts/0                       Sun Feb 11 14:47 - 15:54  (01:07)
root     tty1                        Sun Feb 11 14:24   still logged in
reboot   system boot  2.2.14         Sun Feb 11 14:23          (04:32)
lori     pts/2                       Sat Feb 10 19:09 - 01:05  (05:55)
tapico   pts/1                       Sat Feb 10 18:49 - 01:05  (06:15)

wtmp begins Mon Feb  5 20:18:53 2001
```

8.7.7 lastlog

Mit dem Befehl `lastlog` ist es möglich den Zeitpunkt des letzten Einloggens eines Benutzer zu bestimmen.

```
lastlog [OPTIONEN]
```

Optionen	
`-t N`	beschränkt die Ausgabe auf die letzten `N` Tage. (*time*)
`-u NAME`	erlaubt die direkte Anzeige der Daten eines Benutzers. (*user*)

Für die Auswertung greift der Befehl auf die Logdatei `/var/log/lastlog` zurück.

```
tapico@defiant:~ > lastlog -u tapico
Username         Port     From             Latest
tapico           tty5                      Son Feb 11 16:21:59 +0100 2001
tapico@defiant:~ > lastlog -t 1
Username         Port     From             Latest
root             tty2                      Son Feb 11 16:22:16 +0100 2001
tapico           tty5                      Son Feb 11 16:21:59 +0100 2001
lori             tty3                      Son Feb 11 16:23:18 +0100 2001
```

Notizen:

VIII	Arbeitsblatt Benutzerverwaltung	8.1

1 Loggen Sie sich als *root* im Terminal 1 ein.

2 Richten Sie den Benutzer *grayson* mit den Default-Bedingungen ein.

3 Welche Einträge sind für den Benutzer in `/etc/passwd` und `/etc/shadow` gemacht worden?

4 Ist ein Home-Verzeichnis für *grayson* angelegt worden?

5 Wechseln Sie zu Terminal 2 und loggen Sie sich als *grayson* ein.

6 Wechseln Sie zu Terminal 1 und ändern das Paßwort von *grayson* auf 'werner'.

7 Welches Paßwort steht in `/etc/shadow` für *grayson*? Vergleichen Sie den Eintrag mit dem Eintrag ihres Nachbarn.

8 Wechseln Sie zu Terminal 2 und loggen Sie sich als *grayson* ein.

9 Welches ist das aktuelle Verzeichnis?

10 Legen Sie den Benutzer *lori* mit folgendem Befehl an!
`useradd -c "Lori Kalmar" -m -p locust lori`
Was bewirkt der Befehl?

11 Wechseln Sie zu Terminal 3 und versuchen sich als *lori* einzuloggen. Was passiert?

12 Wechseln Sie wieder zu Terminal 1 und lassen sie sich die `/etc/shadow` anzeigen. Warum konnten Sie sich nicht als *lori* einloggen?

13 Ändern Sie das Paßwort von *lori* so, daß Sie sich einloggen können? Testen Sie das.

14 Editieren Sie die Datei `/etc/shadow`. Kopieren Sie das Paßwort von *lori* in das Paßwortfeld von *grayson*. Versuchen Sie sich nun wieder an Terminal 2 mit dem Paßwort von *lori* als *grayson* einzuloggen.

15 Editieren Sie die Datei `/etc/shadow` für den Benutzer *grayson*. Fügen Sie vor dem verschlüsselten Paßwort ein '`*`' (Asterisk, Stern) ein. Versuchen Sie sich wieder am Terminal 2 als *grayson* einzuloggen. (Eventuell vorher ausloggen.)

16 Machen Sie die Änderung in `/etc/shadow` rückgängig und versuchen Sie sich wieder am Terminal 2 als *grayson* einzuloggen.

17 Wie müssen Sie vorgehen um ein Konto fürs Einloggen zu deaktieren? Dabei darf das Paßwort nicht unbrauchbar gemacht werden, da es für einen FTP-Zugang gebraucht wird. Beschreiben Sie das Verfahren und testen Sie es am Benutzer *grayson*.

18 Wechseln Sie zu Terminal 1 und legen Sie die Gruppe *greydeath* an.

19 Lassen Sie sich alle Gruppen anzeigen in denen *root* Mitglied ist. Benutzen Sie dazu die Befehle `groups` und `groups root`. Erklären Sie den Unterschied.

20 Fügen Sie *root* zur Gruppe *greydeath* hinzu.

21 Lassen Sie sich alle Gruppen anzeigen in denen *root* Mitglied ist. Was ist das Problem? (Testen Sie die unterschiedliche Arbeitsweise von `groups` und `groups root`.)

22 Lassen Sie sich die Datei `/etc/group` anzeigen. Ist *root* eingetragen bei der Gruppe *greydeath*?

23 Loggen Sie sich einmal aus und wieder als *root* ein. Überprüfen Sie wiederum die Gruppenzugehörigkeit.

VIII	Arbeitsblatt Benutzerverwaltung	8.2

24 Legen Sie die Datei `/trellwan` (Inhalt: ein schöner Spruch) an. Ändern Sie die Gruppenzugehörigkeit auf *greydeath* und die Rechte mit dem Befehl `chmod 640 /trellwan`. Damit besitzt der Besitzer der Datei Schreib- und Leserecht und die Gruppe *greydeath* das Leserecht.

25 Können *grayson* und *lori* auf `/trellwan` zugreifen?

26 Fügen Sie als *root grayson* zur Gruppe *greydeath* hinzu. Lassen Sie sich als *root* und als *grayson* die Gruppenzugehörigkeit von *grayson* anzeigen.

27 Lassen Sie sich nun als *grayson* den Inhalt von `/trellwan` anzeigen.

28 Loggen Sie sich als Benutzer *lori* aus. Löschen Sie den Benutzer *lori* mit seinem Heimatverzeichnis.

29 Löschen Sie die Gruppe *greydeath*.

30 Versuchen Sie nun den Benutzer *grayson* zu löschen. Was passiert?

31 Loggen Sie als *grayson* aus und löschen den Benutzer. Was ist mit seinem Heimatverzeichnis passiert?

32 Welche Informationen bekommen Sie über die Datei `/trellwan`?

33 Löschen Sie die Datei `/trellwan`.

34 Sie sollen auf der Applikationsebene einen Fehler für einen Benutzer beheben. Leider kennen Sie das Passwort des Benutzers nicht, wollen es aber auch nicht permanent ändern. Beschreiben Sie ihre Vorgehensweise.

Notizen:

Kapitel 9

Rechte im Linux-Dateisystem

9.1 Die Rechte

Basierend auf der Art, wie auf Dateien zugegriffen werden kann, unterscheidet Linux drei Rechte. Sie werden abkürzend mit `r`, `w` oder `x` bezeichnet. Die einzelnen Rechte sind an die jeweilige Datei gebunden und werden in der Inode der Datei gespeichert. Dabei ist besonders wichtig in welcher Stellung Sie sich zu der Datei befinden. Sind Sie der Besitzer, gehören Sie zur Gruppe, die Zugriff auf die Datei hat, oder gehören Sie zum Rest der Welt.

Wie vom Befehl `ls -l` bekannt, besteht die Rechtetabelle aus 9 Angaben, z. B. `rwxr--xr--`. Die ersten drei angegebenen Rechte gelten dabei für den Dateibesitzer, die folgenden drei für die Besitzergruppe und die letzten für den Rest der Welt.

9.1.1 Rechte auf Dateien

Die Wirkung der Rechte ist bei Dateien und Verzeichnissen unterschiedlich. Bei Dateien arbeiten Sie wie folgt:

r (read = lesen) Der Benutzer kann den Inhalt der Datei einsehen, d. h. unter anderem, er kann sie am Bildschirm anzeigen lassen, sie drucken oder kopieren.

w (write = schreiben) Der Benutzer kann die Datei verändert unter dem bisherigen Namen speichern.

x (execute = ausführen) Die Datei kann als Programm gestartet werden. Dies setzt natürlich voraus, daß die Datei ein Programm ist.

Angenommen also, Sie haben an der Datei `uschistagebuch` kein r-Recht, wollen Sie aber trotzdem auf dem Bildschirm sehen. Die Eingabe von
`cat uschistagebuch`
führt dann zu einer Fehlermeldung der Art:
`uschistagebuch: permission denied`

Viele Anwender, die aus einer Einplatzumgebung zu Linux stoßen, überschätzen allerdings die Bedeutung der Rechte, die man an einer (gewöhnlichen) Datei hat. Dies mag daran liegen, das MS-DOS und viele seiner Verwandten den Zugriff auf Dateien durch Attribute zu regeln versuchen. Die vielzitierte "schreibgeschützte Datei" gibt es unter Linux nicht.

Der Dateischutz wird ausschließlich über die Rechte gewährleistet. Hierbei gilt eine glasklare Logik, die sich im wesentlichen aus dem Dateisystem ergibt: Wird in einer Befehlszeile ein Pfad angegeben, auf den der Befehl zugreifen soll, werden die einzelnen Bestandteile des Pfades daraufhin überprüft, ob hinreichende Rechte bestehen. Ist dies nicht der Fall, wird der Zugriff verweigert.

9.1.2 Rechte auf Verzeichnisse

Verzeichnisse sind für Linux nur eine spezielle Art von Dateien. Dies macht sich hier daran bemerkbar, daß die Rechte `r` und `w` für Verzeichnisse genauso gehandhabt werden, wie für Dateien. Daraus ergeben sich weitreichende Konsequenzen für die darin enthaltenen Dateien!

Das r-Recht an einem Verzeichnis zu besitzen heißt, man kann den Inhalt des Verzeichnisses lesen, mit anderen Worten, man kann den Befehl `ls` für dieses Verzeichnis einsetzen.

Was aber ist nun der Inhalt eines Verzeichnisses? Ein Verzeichnis ist nichts anderes als eine kleine Tabelle. Jede Zeile dieser Tabelle hat zwei Einträge: Den Dateinamen und die zugehörige Inode-Nummer. Wie Sie bereits wissen, ist der Dateiname für Linux nicht das entscheidende Kriterium für die Bearbeitung einer Datei. Linux orientiert sich an der Inode.

Was bedeutet in diesem Zusammenhang das r-Recht für ein Verzeichnis?

Ganz einfach: Mit dem r-Recht am Verzeichnis können darin Dateien gefunden werden (der Befehl `ls` liefert die Namen der im Verzeichnis enthaltenen Dateien), zusätzlich kann man die Inode-Nummern dieser Dateien abfragen (`ls -i`).

Da in der Inode selbst die Verwaltungsinformationen (darunter die Rechtetabelle für die Datei) gespeichert sind, ist ein Zugriff auf eine Datei also erst dann möglich, wenn man ihre Inode kennt! Das heißt also, wenn Ihnen das r-Recht an einem Verzeichnis fehlt, haben Sie keinen Zugang zu den darin enthaltenen Dateien! Insbesondere sind auch Unterverzeichnisse dem Zugriff des Anwenders entzogen!

Das für das Verständnis der Linux-Rechte wichtigste Recht ist wohl das w-Recht für Verzeichnisse. Dieses Recht beinhaltet die Möglichkeit ein Verzeichnis verändert abzuspeichern. Anders ausgedrückt: Ein Benutzer, der an einem Verzeichnis das w-Recht hat, kann dieses verändern.

Nun ist wie gesagt ein Verzeichnis eine Liste von Dateinamen (plus zugehöriger Inode-Nummer). Besitzen Sie das w-Recht an einem Verzeichnis, können Sie einen Verzeichniseintrag löschen, das Verzeichnis wird verändert gespeichert, der Dateiname und die Inode-Nummer existieren nicht mehr. Einen Verzeichniseintrag löschen heißt also den Zugang (Link) zu einer Datei löschen! Dies geschieht allerdings nicht mit einem Editor sondern mit Hilfe des Befehls `rm`.

Verzeichnisse sind insofern "schreibgeschützt", als Linux aus Konsistenzgründen schreibende Zugriffe auf Verzeichnisse nur mit dafür geeigneten Befehlen erlaubt. Das Bearbeiten mit einem Editor würde viele Benutzer überfordern, denn es muß Rücksicht auf die Struktur des Verzeichnisinhalts genommen werden. Wird diese aber zerstört, sind die Daten im Verzeichnis nicht mehr verfügbar!

Nehmen wir nun einen extremen Fall an. Eine Datei, an der Sie keinerlei Rechte besitzen, befinde sich in einem Verzeichnis, in dem Sie das w-Recht haben. Diese Datei können Sie, auch ohne Rechte an ihr selbst, löschen, denn Sie können das Verzeichnis manipulieren! Sie entfernen mit dem Befehl `rm` den Eintrag der Datei im Verzeichnis und somit löschen Sie den Zugang zu den Daten.

Gleiches gilt für das Umbenennen (ändert den Eintrag in einem Verzeichnis), Verschieben (löscht den Eintrag in einem Verzeichnis) oder Erstellen (erzeugt einen neuen Eintrag in einem Verzeichnis) einer Datei. Auch hierzu ist das w-Recht am Verzeichnis von entscheidender Bedeutung.

Wie Sie sehen hängen sehr viele Dateioperationen von Verzeichnisrechten ab. Vereinfachend gesagt, Verwaltungsoperationen mit Dateien, wie z. B. Löschen, Verschieben oder Umbenennen sind Eingriffe in die Verzeichnisstruktur und werden deshalb über Rechte an Verzeichnissen geregelt. Die inhaltliche Manipulation von Dateien ist dagegen nur möglich, wenn Sie Rechte direkt an der Datei haben. Es kann sein, daß Sie eine Datei löschen, umbenennen oder verschieben können, ohne ihren Inhalt auch nur sehen zu können. Zur Manipulation der Dateiinhalte benötigt Sie zumindest das r-Recht an ihr, alle anderen genannten Operationen erfordern das w-Recht am Verzeichnis.

Alles in allem kann man feststellen, daß Verzeichnisse bezüglich der Rechte `r` und `w` wie gewöhnliche Dateien behandelt werden. Da man aber mit einem Verzeichnis eine Liste von Dateinamen bearbeitet, bedeutet das Ändern von Verzeichnissen sehr häufig ein Löschen, Umbenennen oder Verschieben von Dateien.

Bis jetzt haben sich Verzeichnisse wie Dateien verhalten. Beim x-Recht gibt es aber einen signifikanten Unterschied. `x` steht für das Recht ein Programm ausführen zu können. In diesem Sinne kann also ein Verzeichnis nicht ausführbar sein.

Das x-Recht an einem Verzeichnis bedeutet nun, daß Sie

- mit cd in das Verzeichnis wechseln dürfen und
- Zugang zu den Inodes aller darin enthaltenen Dateien erhalten.

Rechte an einem Verzeichnis betreffen die darin enthaltenen Dateien. Wollen Sie sich den Inhalt der Datei `/home/tapico/.profile` mit
`cat /home/tapico/.profile`
ansehen, so setzt dies folgende Rechte voraus:

- Das r-Recht an der Datei selbst.
- Das r-Recht am Verzeichnis `/home/tapico` .
- Das x-Recht an diesem Verzeichnis.

`cat` muß zunächst den Inhalt von `/home/tapico` lesen, um zu prüfen, ob die Datei überhaupt existiert. Dies erlaubt das r-Recht am Verzeichnis. Weiter benötigt `cat` die Inode-Nummer der Datei, um über diese die Inode selbst finden, Zugang dazu erhält `cat` über das x-Recht an `/home/tapico` In der Inode sind nun aber alle relevanten Informationen, die `cat` braucht, um den Dateiinhalt endlich lesen zu können. Hier ist die Rechtetabelle, die über das r-Recht die Erlaubnis zum Lesen gewährt und schließlich sind hier die Informationen abgelegt, mit deren Hilfe die Daten auf der Platte gefunden werden können.

In der Praxis erhält man mit dem r-Recht an einem Verzeichnis stets auch das x-Recht. Natürlich kann man das x-Recht wieder nehmen (sofern man dazu die Erlaubnis hat), dann schafft man aber eine skurile Situation: Das Leserecht am Verzeichnis erlaubt es dann zum Beispiel `ls`, Dateien zu finden, weitere Informationen werden aber verweigert. Man erhält in jedem Fall eine Fehlermeldung wie die folgende
/bin/ls: /home/tapico/.profile: Permission denied
`ls` meldet, daß man auf die Inode von `.profile` im Verzeichnis `/home/tapico` nicht zugreifen darf: `Permission denied`.

An den meisten Verzeichnissen im Dateisystem hat der gewöhnliche Benutzer die Rechte `r` und `x`, am Heimatverzeichnis aber alle Rechte.

Neben den Abkürzungen `r`, `w` und `x` gibt es auch noch eine oktale[1] Notation der Rechte. Dies ist eine dreistellige Zahl, bei der die erste Ziffer für den Besitzer gilt, die zweite für die Gruppe, die dritte für alle anderen.

Ziffer	Bedeutung
4	r-Recht
2	w-Recht
1	x-Recht
0	kein Recht

Die Rechtekombinationen ergeben sich durch Addition! So bedeutet 754 dasselbe wie `rwxr-xr--`!

Anstelle der Kennzeichnung `x` treten hin und wieder ein `s` oder ein `t` auf.

9.1.3 Spezielle Rechte: SUID und SGID

SUID (*Set User ID*) und SGID (*Set Group ID*) werden durch ein `s` anstelle eines `x` bei den Eingentümer- und bei den Gruppenrechten ausgedrückt. Das s-Recht an einer ausführbaren Datei bedeutet, daß der Benutzer, der es startet, während des Programmlaufes die UID des Dateibesitzers bzw. die GID der Besitzergruppe erhält. Das hängt davon ab, ob das `s` beim Besitzer oder der Besitzergruppe steht. Im Falle des Programms `/bin/passwd` wird man z. B. während des Programmlaufs zum Superuser!

```
tapico@defiant:~ > ls -l /etc/passwd /etc/shadow
-rw-r--r--   1 root     root         3810 Dec 26 19:56 /etc/passwd
-rw-r-----   1 root     shadow       2422 Dec 26 19:56 /etc/shadow
tapico@defiant:~ > ls -l /usr/bin/passwd
-rwsr-xr-x   1 root     shadow      27920 Mar 11  2000 /usr/bin/passwd
```

[1] Auf der Basis der Zahl 8!

Wie Sie sehen, hat nur der Superuser Schreibrechte auf die Dateien `/etc/passwd` und `/etc/shadow`. Das s-Recht auf den Befehl `/usr/bin/passwd` bewirkt, daß der Befehl immer unter der UID des Superuser ausgeführt wird. So kann auch ein normaler Benutzer sein Kennwort ändern.

Wird das SGID-Recht auf ein Verzeichnis gegeben, so hat es eine andere Funktion. In diesem Fall erhalten alle Dateien, die in diesem Verzeichnis erstellt werden, automatisch die gleiche Gruppe, die auch dem Verzeichnis zugeordnet ist.

Beispiel

Nehmen wir mal an, der User *tapico*, dem die Hauptgruppe *users* zugeteilt wurde, ist auch Mitglied der Gruppe *web*. Für das Verzeichnis `/home/wwwrun` mit der zugeordneten Gruppe *web* wurde das SGID-Recht gesetzt. Legt *tapico* in einem Verzeichnis z. B. seinem Heimatverzeichnis eine Datei an, so wird der Datei seine Hauptgruppe zugeordnet. Legt er aber in dem Verzeichnis `/home/wwwrun` eine Datei an, so bekommt die Datei die Gruppe des Verzeichnis zugewiesen.

```
tapico@defiant:~> touch screen
tapico@defiant:~> ls -l screen
-rw-r--r--    1 tapico   users           0 Nov 18 09:07 screen
tapico@defiant:~> cd /home/wwwrun
tapico@defiant:/home/wwwrun> ls -ld .
drwxrwsr-x    2 wwwrun   web            35 Nov 18 08:58 .
tapico@defiant:/home/wwwrun> touch mahal.html
tapico@defiant:/home/wwwrun> ls -l
insgesamt 0
-rw-r--r--    1 tapico   web             0 Nov 18 09:03 mahal.html
```

9.1.4 Skript: Formatieren von Disketten

Wie schon im Abschnitt 10.3.1 erwähnt, kann nur der Benutzer *root* Dateisystem einrichten. Sollen nun normale Benutzer Disketten formatieren können, so kann dies durch ein Skript erfolgen, was durch das Setzen des SUID-Rechtes unter der Identität von *root* läuft.

Beispiel für das Skript:

```
root@defiant:/usr/bin> ls -l fd0format
-rwsr-xr-x    1 root     users         124 Okt 19 22:33 fd0format
root@defiant:/usr/bin> cat fd0format
#Formatiert eine Diskette Low-Level und erzeugt das Dateisystem ext2
#SUID für das Skript setzen, da /sbin nur für root zugänglich

/usr/bin/fdformat /dev/fd0
/sbin/mkfs -t ext2 /dev/fd0
```

9.1.5 Spezielle Rechte: Sticky Bit

Das t-Recht, auch *Sticky Bit* oder *Save Text Bit* genannt, wirkt sich auf Verzeichnisse und Dateien unterschiedlich aus.

Das Sticky Bit bei ausführbaren Dateien veranlaßt Linux, den Programmcode des gestarteten Programms nicht wieder aus dem Arbeitsspeicher zu entfernen. Dies hat den Vorteil, daß bei einem weiteren Start des Programms dieses nicht erst in den RAM geladen werden muß. Dieses Merkmal ist besonders nützlich, wenn ein Programm häufig eingesetzt wird; denn der Programmcode muß nur einmal im Speicher sein, er kann aber von mehreren Benutzern gleichzeitig genutzt werden.

Bei Verzeichnissen hingegen ermöglicht das Sticky Bit das Anlegen von gemeinsam genutzten Verzeichnissen. Normalerweise muß ein solches Verzeichnis die Rechte `drwxrwxrxw` besitzen. Jeder kann nun in dieses Verzeichnis schreiben, allerdings auch jede Datei darin löschen. Durch das Setzen des Sticky Bits wird es nur dem Eigentümer einer Datei erlaubt, diese zu löschen.

Das t-Recht läßt sich nur vom Superuser vergeben.

Oktal werden die neuen Rechte durch eine weitere vorangestellte Ziffer ausgedrückt. So werden z. B. die Rechte `rwxrwxrwt` durch 1777 ausgedrückt.

Ziffer	Bedeutung
4	SUID
2	GUID
1	Sticky Bit

9.2 Verwalten und Setzen von Rechten

Das Verwalten und Setzen von Rechten übernehmen Programme wie `chown`, `chmod`, `chgrp` und `umask`.

9.2.1 chown

Mit dem Befehl `chown` ist es möglich der Datei einen neuen Besitzer und eine neue Gruppe zu geben.

`chown [OPTIONEN] [BENUTZER][:GRUPPE] DATEILISTE`

Allerdings ist es nur dem Superuser gestattet, den Eigentümer einer Datei zu ändern. Die Änderung der Gruppe ist für den Besitzer und für den Superuser erlaubt. Der Besitzer darf allerdings seiner Datei nur eine Gruppe zuordnen, in der er selber Mitglied ist.

Optionen

`-c`	Informationen über alle geänderten Dateien werden angezeigt
`-v`	Informationen über alle Aktionen werden ausgegeben
`-f`	Fehlermeldungen werden nicht ausgegeben
`-R`	Änderungen werden rekursiv den Verzeichnisbaum herunter durchgeführt

Beispiele

`chown tapico pampelmuse`
ändert den Besitzer der Datei `pampelmuse` auf **tapico**.

`chown -R tapico:users /home/uschi`
ändert den Besitzer auf **tapico** und die Gruppe auf **users** für das Verzeichnis `/home/uschi` und seinem Inhalt.

`chown :xtra xtra*`
ändert die Gruppe für alle Dateien, die mit "xtra" beginnen auf **xtra**.

9.2.2 chgrp

Ändert für eine Datei die zugehörige Gruppe.

`chgrp [OPTIONEN] GRUPPE DATEILISTE`

Die Änderung der Gruppe ist für den Besitzer und für den Superuser erlaubt. Der Besitzer darf allerdings seiner Datei nur eine Gruppe zuordnen, in der er selber Mitglied ist.

Optionen

`-c`	Informationen über alle geänderten Dateien werden angezeigt
`-v`	Informationen über alle Aktionen werden ausgegeben
`-f`	Fehlermeldungen werden nicht ausgegeben
`-R`	Änderungen werden rekursiv den Verzeichnisbaum herunter durchgeführt

Beispiel

`chgrp users pampelmuse`
ändert die Gruppe der Datei `pampelmuse` auf **users**.

9.2.3 chmod

Der Befehl `chmod` erlaubt das Ändern der Rechte für eine Datei.

```
chmod [OPTIONEN] RECHTE DATEILISTE
```

Die Rechte können dabei in der Form *WerWieWas* oder *Maske* angegeben werden.

Wer ist dann `u` (user/Besitzer), `g` (group/Gruppe), `o` (others/Sonstige), eine Kombination daraus oder `a` (all user/alle Benutzer).

Was ist eines der Rechte `r`, `w`, `x`, `s`, `t` oder eine Kombination daraus. Wobei sich `x`, `s` und `t` sich gegenseitig ausschließen.

Wie ist `=` (Rechte werden exakt so gesetzt wie in *Was* angegeben), `+` (Rechte aus *Was* werden zusätzlich vergeben) oder `-` (Rechte aus *was* werden entzogen)

WerWieWas kann mehrfach, dann durch Komma getrennt angegeben werden. In der ganzen Liste der *WerWieWas* darf kein Leerzeichen sein, da alles nach diesem Leerzeichen als Dateiname interpretiert wird!

Maske ist eine dreistellige (Superuser auch vierstellige) oktale Zahl wie in Abschnitt 9.1.2 und 9.1.5 beschrieben.

Optionen

`-c`	Informationen über alle geänderten Dateien werden angezeigt
`-v`	Informationen über alle Aktionen werden ausgegeben
`-f`	Fehlermeldungen werden nicht ausgegeben
`-R`	Änderungen werden rekursiv den Verzeichnisbaum herunter durchgeführt

Beispiele

`chmod u+x memo*.txt`
fügt für den Besitzer das x-Recht hinzu.

`chmod 755 makeitso`
setzt die Rechte auf `rwxr-xr-x`.

`chmod a-rwx,u=rwx meinsein`
entzieht allen alle Rechte und gibt dem Besitzer alle Rechte.

9.2.4 umask

Für die Erstellung neuer Dateien gibt es für die Rechte eine Maske, die bestimmt welche Rechte vergeben werden. Für diese Maske ist der Befehl `umask` (*User's creation MASK*) zuständig.

```
umask [MASKE]
```

Die Eingabe von `umask` ohne Parameter gibt die aktuell eingestellte Maske wieder. Die Maske selber ist eine dreistellige oktale Zahl. Die Bedeutung der Zahlen ist identisch mit derer für die Rechte aus Abschnitt 9.1.2. Allerdings gibt `umask` nicht an, welche Rechte gegeben werden, sondern welche Rechte entzogen werden.

Allerdings verhält sich die Rechtevergabe doch anders als erwartet. Manche Systeme liefern auch eine vierte Ziffer (an erster Stelle stehend) zurück. Diese beschreibt das besondere Verhalten der Rechtevergabe für neue Dateien, wie aus der Tabelle 9.1 zu ersehen ist. Dabei verhält sich die `bash` wie durch den Wert 0 gegeben. Die für die Benutzergruppen angegebenen Rechte werden von dieser Grundeinstellung abgezogen.

Beim Kopieren von Dateien hat `umask` auch seine Finger im Spiel. Als Vorgabe gelten die Rechte, die die Quelldatei besitzt. Diese werden dann mit der Einstellung von `umask` gefiltert.

Der durch `umask` eingestellte neue Wert gilt nur für die Dauer einer Sitzung. Eine permanente Änderung erreichen Sie, wenn Sie den Befehl `umask` in die Konfigurationsskripte `.profile` oder `.bashrc` eintragen.

Beispiel

`umask 022`
setzt die Defaultmaske auf `rwxr-xr-x`.

Wert	Bedeutung
0	rw für Dateien, rwx für Verzeichnisse
1	rw für Dateien und Verzeichnisse
2	r für Dateien, rx für Verzeichnisse
3	r für Dateien und Verzeichnisse
4	w für Dateien, wx für Verzeichnisse
5	w für Dateien und Verzeichnisse
6	x für Dateien und Verzeichnisse
7	Keine Rechte für Dateien und Verzeichnisse

Tabelle 9.1: Besondere Berechtigungen für `umask`

9.3 Mit anderen Rechten arbeiten

Manchmal ist es notwendig, daß ein normaler Benutzer ein Kommando ausführen soll, daß ansonsten nur `root` zugänglich ist. Bei normalen Skripten ist dies weniger ein Problem, als bei den Systemprogrammen. Diese besitzen meistens intern noch weitere Prüfroutinen auf den Benutzer. Leider sind diese meistens so restriktiv, daß man sogar mit dem SUID- und SGID-Recht nicht weiterkommt.

Im Prinzip können Sie Ihre Identität mit dem Befehl `su` wechseln. Sie können auch einen Befehl direkt mit `su` starten.

```
ole@enterprise:~> su -c "mount -t auto /dev/hda1 /mnt"
Password:
ole@enterprise:~>
```

Allerdings setzt diese Operation die Kenntnis des Paßworts desjenigen voraus, dessen Identität man annehmen will. Wenn Sie allerdings dies kennen, dann können Sie alles unter dieser Identität machen.

Um dieses Problem zu erledigen wurde der Befehl `sudo` entwickelt.

9.3.1 sudo

Der Befehl `sudo` erlaubt die Ausführung eines Programms unter fremder Identität. Dabei wird in der Konfigurationsdatei `/etc/sudoers` festgelegt, welcher Benutzer auf welchem Host was tun darf.

```
sudo [OPTIONEN] Kommando
sudo -v
```

Dabei wird die reale und die effektive UID und GID so gesetzt, daß sie mit den Daten des angegebenen Benutzers übereinstimmen. Die Standardeinstellung ist, daß der Benutzer sich vor der Ausführung mit seinem eigenen Paßwort identifizieren muß. Wenn sich ein Benutzer authentifiziert hat, dann wird ein Zeitstempel aktualisiert. Der Benutzer kann nun für 5 Minuten `sudo` benutzen, ohne sein Paßwort neu eingeben zu müssen. Die Zeitstempel werden im Verzeichnis `/var/run/sudo` gespeichert.

Optionen

`-V`	Informationen über Version und Einstellungen (nur *root* (*version*)
`-l`	Zeigt die möglichen Befehle eines Benutzers an (*list commands*)
`-L`	Zeigt die möglichen Standardoptionen der Datei `/etc/sudoers` an (*list options*)
`-h`	Zeigt Hilfe zu `sudo` an (*help*)
`-v`	Aktualisiert den Zeitstempel, damit es zu keinem Timeout kommt (*validate*)
`-k`	Setzt den Zeitstempel auf eine ungültige Zeit, beim nächsten `sudo`-Befehl wird das Paßwort wieder abgefragt (*kill*)
`-K`	Löscht den Zeitstempel völlig, beim nächsten `sudo`-Befehl wird das Paßwort wieder abgefragt (*sure kill*)
`-b`	Startet das Kommando als Hintergrundprozeß (*background*)
`-p PROMPT`	Erlaubt die Definition eines eigenen Paßwortprompts (%u für Benutzername und %h für Rechnername stehen als Variablen zur Verfügung) (*prompt*)
`-u BENUTZER`	Das Kommando wird unter dem angegebenen Benutzer ausgeführt, Standard ist *root*; um eine UID anstatt einem Benutzernamen zu benutzen, wird ein Schweinegatter davorgestellt (#512) (*user*)
`-s`	Benutzt die in der Variablen SHELL angegebene Shell (*shell*)
`-H`	Setzt die ansonsten unveränderte Variable `HOME` auf den Wert des angegebenen Benutzers (*home*)
`-P`	Benutzt die Gruppen des Benutzers anstatt der Gruppen des angegebenen Benutzers (*preserve*)
`-S`	Das Password wird von der Standardeingabe gelesen(*stdin*)

Beispiele

Was kann der aktuelle Benutzer ausführen?

```
ole@enterprise:~> sudo -l
User ole may run the following commands on this host:
    (root) /usr/sbin/useradd
    (root) /usr/sbin/userdel
    (root) /usr/bin/passwd
```

Danach sollte der Benutzer einen Benutzer anlegen, sein Passwort ändern und ihn auch wieder löschen können.

```
ole@enterprise:~> /usr/sbin/useradd theodor
useradd: unable to lock password file
ole@enterprise:~> sudo /usr/sbin/useradd theodor
Password:
ole@enterprise:~> sudo passwd theodor
Changing password for theodor.
New password:
Re-enter new password:
Password changed
ole@enterprise:~> sudo /usr/sbin/userdel theodor
ole@enterprise:~>
```

9.3.2 /etc/sudoers

In der Datei `/etc/sudoers` wird das Verhalten von `sudo` definiert. Eine Vielzahl von Parametern steuert das Programm. Hier wird nur auf die einfachen Einträge eingegangen.

So könnten eine ganz einfach `/etc/sudoers`-Datei aussehen.

Listing 9.1 */etc/sudoers*

```
# sudoers file.
```

```
#
# This file MUST be edited with the 'visudo' command as root.
#
# See the sudoers man page for the details on how to write a sudoers file.
#
# Host alias specification
# User alias specification
# Cmnd alias specification
# Defaults specification
# User privilege specification
root    ALL=(ALL) ALL
ole     ALL=/usr/sbin/useradd, /usr/sbin/userdel, /usr/bin/passwd
```

Ein Eintrag besteht aus mehreren Teilen. Dem Benutzer bzw. Gruppennamen, von welchem Rechner aus das Kommando ausgeführt werden soll, welche Kommandos ausgeführt werden dürfen und als welcher Benutzer etwas ausgeführt werden soll.

Die Benutzer aus der Gruppe *gott* dürfen alles ausführen.

```
%gott ALL=(ALL) ALL
```

Jetzt sollen Sie dies auch tun können ohne sich noch einmal extra identifizieren zu müssen.

```
%gott ALL=(ALL) NOPASSWD: ALL
```

Die Mitglieder der Gruppe *users* sollen den Rechner herunterfahren können. Allerdings nur, wenn sie auch direkt auf dem Rechner eingeloggt sind.

```
%users localhost=/sbin/shutdown -h now
```

Der Benutzer *ole* soll neue Benutzer anlegen, ihr Paßwort ändern und die Benutzer auch wieder löschen können.

```
ole     ALL=/usr/sbin/useradd, /usr/sbin/userdel, /usr/bin/passwd
```

Weitere Informationen entnehmen Sie bitten den Manual-Pages (`man 5 sudoers`).

9.3.3 visudo

Natürlich kann man die Datei `/etc/sudoers` auch direkt mit einem Editor bearbeiten. Die Benutzung des Kommandos `visudo` bring dennoch Vorteile. Es ermöglicht die Bearbeitung der Datei auf einer sicheren Art und Weise. `visudo` sperrt die Datei `/etc/sudoers` gegen gleichzeitigen Zugriff. Außerdem kann es einfache Gültigkeits- und Syntaxtests durchführen.

```
visudo [OPTIONEN]
```

Das Kommando verwendet als Editor den Standard-Editor des Systems. Also in den meisten Fällen den `vi`. Weitere Informationen entnehmen Sie bitten den Manual-Pages (`man 8 visudo`).

IX	Arbeitsblatt Rechte	9.1

Zu Beginn der Aufgabe sollten Sie sich im Heimatverzeichnis befinden! Wechseln Sie das Verzeichnis nur, falls Sie dazu ausdrücklich aufgefordert werden. Einige Aufgaben lassen sich nicht ausführen! Versuchen Sie in diesen Fällen die Fehlermeldung zu interpretieren.

1 Loggen Sie sich als Benutzer *walter* auf der Konsole 2 ein und versuchen Sie die Datei `/boot/vmlinuz` zu löschen.

2 Kopieren Sie die Datei `shadow` aus dem Verzeichnis `/etc` in ihr Heimatverzeichnis!

3 Kopieren Sie die Datei `passwd` aus `/etc` ins aktuelle Verzeichnis!

4 Wem gehört die Datei `passwd` a) in `/etc`, b) im aktuellen Verzeichnis?

5 Lassen Sie sich aus der Datei `/etc/passwd` nur die Zeile anzeigen, die Ihren Benutzernamen enthält!

6 Lassen Sie sich nur die Benutzernamen der Datei `/etc/passwd` anzeigen!

7 Erstellen Sie mit dem `vi` die Datei `adressen`!

```
Thomas Prizzi Wall 40 24103 Kiel 0431/978767
Martin Lecker Eggerstedtstr. 1 24103 Kiel 0431/978234
Norbert Nase Nasa-Str. 23 55111 Nasenhausen 0321/918273
Zacharias Ackermann Feldweg 33 12345 Kleindorf 06789/1011
```

8 Lassen Sie sich die Adresse von Norbert Nase aus der Datei `adressen` ausgeben !

9 Wieviele Adressen stammen aus Kiel ?

10 Geben Sie die Adressen sortiert nach Nachnamen aus !

11 Geben Sie nur die Nachnamen aus ?

12 Wieviele Buchstaben hat der Nachname von Martin ?

13 Stellen Sie fest, welche Voreinstellung für das Erstellen (neuer) gewöhnlicher Dateien besteht, ändern Sie sie gegebenenfalls so, daß alle (user, group, other) das Recht zum Lesen erhalten und sonst keines!

14 Kopieren Sie die Datei `vdir` aus `/usr/bin` in Ihr Heimatverzeichnis!

15 Erstellen Sie mittels `cat` eine Datei namens `indertat` (Inhalt: Ein beliebiges Sprichwort)!

16 Prüfen Sie den Inhalt Ihres Heimatverzeichnis und damit die Rechte der Dateien `vdir` und `indertat`. Vergleichen Sie das Ergebnis mit den Voraussetzungen, die Sie in Aufgabe 13 geschaffen haben!

17 Verändern Sie die Voreinstellung zum Erstellen neuer gewöhnlicher Dateien so, daß Sie (Besitzer) diese Dateien Lesen und Verändern können, alle sonstigen Teilnehmer (Gruppe, Andere) nur Lesen können.

18 Kopieren Sie die Datei `mv` aus `/bin` in ihr Heimatverzeichnis.

19 Erstellen Sie mit `cat` die Datei `hilfe` (Inhalt: Eine Kurzbeschreibung von 3 Linuxbefehlen z. B.: cp kopiert Dateien, je Befehl eine Zeile!)

20 Prüfen Sie erneut den Inhalt Ihres Heimatverzeichnis und vergleichen Sie das Ergebnis mit Aufgabe 17 !

21 Verändern Sie die Rechtetabelle von `mv` so, daß kein Recht mehr enthalten ist: `----------`

IX	Arbeitsblatt Rechte	9.2

22 Versuchen Sie die Rechtetabelle von `mv` so zu ändern, daß Sie als Besitzer die Datei ausführen können, und so wenig wie möglich Rechte gesetzt sind !

23 Versuchen Sie die Datei `mv` und `vdir` im Heimatverzeichnis zu löschen! Geht das? Warum geht es bzw. warum nicht?

24 Verändern Sie die Rechte der Datei `.bash_history` so, daß nur Sie sie Lesen, Ändern und Ausführen können!

25 Verändern Sie die Rechte der Datei `adressen` so, daß die Gruppe und die Anderen Sie nicht Lesen oder Ändern können!

26 Prüfen Sie die Rechtetabelle aller Dateien Ihres Heimatverzeichnis!

27 Legen Sie ein Verzeichnis `rights` an!

28 Prüfen Sie die Rechtetabelle am neuen Verzeichnis!

29 Erstellen Sie mittels `cat` eine Textdatei namens `uhr`
Inhalt: `date +"Es ist %H:%M:%S"`

30 Prüfen Sie die Rechtetabelle der neuen Datei! Was fällt Ihnen auf?

31 Machen Sie die Datei ausführbar; es ist damit ein selbsterstelltes Shell-Script!

32 Kopieren Sie die Datei `uhr` und geben Sie ihr dabei den Namen `bimbam`!

33 Verschenken Sie `bimbam` an den Superuser (geben Sie ihr einen anderen Besitzer)!

. . . .

Notizen:

. . . .

X	Arbeitsblatt Datei-Rechte	10.1

Situation: Sie sind User **tapico** und gehören zur Gruppe `group` !

1 Annahme: Für das Verzeichnis `/usr` haben Sie die Rechte x und r, an allen Dateien das Recht r. Sie können:

1. ☐ in das Verzeichnis wechseln
2. ☐ Dateien in diesem Verzeichnis löschen
3. ☐ Dateien in diesem Verzeichnis umbenennen
4. ☐ Dateien aus diesem Verzeichnis verschieben
5. ☐ Dateien aus diesem Verzeichnis anschauen
6. ☐ Dateien aus diesem Verzeichnis kopieren

2 Können Sie im Wurzelverzeichnis / Dateien

1. ☐ löschen
2. ☐ neu anlegen
3. ☐ umbenennen

3 Im Dateisystem gebe es im Verzeichnis `/usr` die (ausführbare) Datei `uhr`. In `/usr` haben Sie die Rechte x und r. Der Befehl `ls -l uhr` gibt Ihnen folgende Ausgabe:

```
-rwxr--r--  root   other   124 Mar  4  16:14  uhr
```

Sie können die Datei

1. ☐ löschen
2. ☐ umbenennen
3. ☐ anschauen
4. ☐ drucken
5. ☐ kopieren
6. ☐ starten

4 Welche Rechte haben Sie in Ihrem Heimatverzeichnis, welche Rechte andere Mitglieder?

5 Woran kann es liegen, daß eine Datei nicht löschbar ist?

X	Arbeitsblatt Datei-Rechte	10.2

6 In Ihrem Heimatverzeichnis befindet sich die Datei `alpha`. Sie gehört **root** (Gruppe **other**). Die Rechtetabelle ist `-rwxr--r--` . Sie können die Datei

1. ☐ löschen
2. ☐ umbenennen
3. ☐ anschauen
4. ☐ ändern
5. ☐ kopieren
6. ☐ verschieben

7 Wo befindet sich der physikalische Ort des Dateisystems `proc`?

Notizen:

XI	Arbeitsblatt Fallstudie: Rechte	11.1

Die Firma AMOV GmbH möchte ihre Daten auf einem Server speichern. Dazu soll als erstes ein Verzeichnisbaum unter dem Verzeichnis `/daten` aufgebaut werden. Dabei müssen die Rechte auf die einzelnen Bereiche ermittelt werden.

Stellvertretend für die Mitarbeiter mit gleichen Rechten nehmen Sie zur Durchführung der Fallstudie bitte die folgenden Mitarbeiter.

Dick Tator ist der Chef. Er hat von internen Abläufen eigentlich keine Ahnung, will aber trotzdem alles wissen. Eigentlich müßte er nur im Bereich der Auftragsbearbeitung arbeiten.

Petra Pleite-Geier arbeitet als Geschäftführerin hart und braucht überall Zugriff um im Notfall alles selbst zu machen.

Manuela Magersucht ist die Dame aus der Buchhaltung und auch für die Kundendatenbank sowie für die Auftragsbearbeitung zuständig.

Maria Krohn ist ebenfalls in der Buchhaltung tätig, daneben ist sie aber auch für die Lohnabrechnung zuständig.

Carlo Calvados ist der feurige Spanier aus der Forschungsabteilung. Außer Manuela Magersucht kennt er nur seine Arbeit, die er zusammen mit seinen Kollegen im Forschungsordner speichert.

Karl Lagerhalle, der Lagerverwalter, braucht den Zugriff auf die Lagerdatenbank.

Michael Schrauber braucht als Produktionsleiter Zugriff auf den Lagerbestand, die Produktionsdaten und die Aufträge.

Peter Proxy als Systemadministrator hat als *root* eigentlich überall seine Nase drin, arbeitet als normaler User aber nur in dem Dokumentationsbereich, den alle Lesen dürfen.

Alle Mitarbeiter sollen Zugriff auf den Pool haben um Daten auszutauschen.

1 Erstellen Sie die Liste der Bereiche mit den zugehörigen Verzeichnissen und Gruppen.

2 Erstellen Sie eine Liste der Benutzer mit ihren UIDs, Heimatverzeichnissen und Gruppenzugehörigkeiten. (UIDs ab 700).

3 Legen Sie die Benutzer an.

4 Legen Sie die Gruppen an.

5 Erstellen Sie aus Ihren Überlegungen die Verzeichnisstruktur und vergeben Sie die passenden Rechte.

6 Testen Sie Ihre Installation.

Notizen:

Kapitel 10

Partitionen und Dateien

10.1 Partitionen

Partitionen werden dazu benutzt um eine Festplatte in einzelne Teile zu zerlegen. Eine Festplatte besteht aus mindestens einer Partition und kann bis zu vier Partitionen enthalten. Um mit den Partitionen arbeiten zu können, müssen sie mit dem gewünschten Dateisystem formatiert werden. Bei den Partitionen werden zwei Typen unterschieden: Die primäre und die erweiterte Partition.

Primäre Partition Eine primäre Partition wird formatiert und ist dann in der Lage Daten zu speichern. Ein Rechner kann nur von einer primären Partition booten.

Erweiterte Partition Eine erweiterte Partition kann nicht direkt zum Speichern von Daten verwendet werden. Sie kann nur logische Laufwerke enthalten, die dann aber formatiert werden können. Eine Festplatte kann bis zu vier primäre Partitionen enthalten oder drei primäre und eine erweiterte Partition.

10.1.1 Partitionsnamen

Die Partitionen werden direkt nach dem Typ des Festplattencontrollers benannt. Dabei steht für IDE-Platten der Buchstabe **h** und für SCSII-Platten der Buchstabe **s**. Dann folgt der Buchstabe **d** für *Disk*. Der dritte Buchstabe nummeriert die Festplatte und die vierte Ziffer gibt die Partitionsnummer an. Die erste Partition der ersten Festplatte mit einem IDE-Controller heißt also `hda1`. Mit einem SCSII-Controller würde sie `sda1` heißen.

10.1.2 Planung der Partitionen

Für ein Linux-System benötigt man mindestens zwei Partitionen, eine für die Wurzel (*root*) (`/`) und eine für die Auslagerungspartition (*swap*). Es ist aber oft ratsam mehrere Partitionen einzurichten. Dabei sollte dem Bootverzeichnis `/boot` und den Heimatverzeichnissen `/home` eine eigenen Partition spendiert werden.

Die Vorteile einer Aufteilung auf mehrere Partitionen sind klar.

Name	Controller	Disk	Partition
`hda1`	IDE-Controller	Erste Festplatte	Erste Partition
`sda1`	SCSII-Controller	Erste Festplatte	Erste Partition
`hdb3`	IDE-Controller	Zweite Festplatte	Dritte Partition
`sdc4`	SCSII-Controller	Dritte Festplatte	Vierte Partition

Tabelle 10.1: Beispiele für Partitionsnamen

- Bei Beschädigung einer Partition bleiben die Daten auf den anderen Partitionen unbeschädigt.
- Wenn es notwendig ist eine Partition neu zu formatieren, bleiben die Daten auf den anderen Partitionen erhalten.
- Schnell wachsende Dateien können auf seperate Partitionen ausgelagert werden um ihre Größe besser überwachen zu können.
- Das Updaten oder die Neuinstallation des Betriebssystems ist einfacher, da Programmdateien und Daten getrennt sind.
- Auch das Backup wird durch die Trennung von Programmdateien und Daten erleichtert.
- Auch die Zeiten für notwendige Überprüfungen des Dateisystems werden durch kleinere Partitionen verringert.

Der Nachteil allerdings liegt in der uneffektiven Nutzung des vorhandenen Festplattenplatzes, da ohne weiteres auf einer Partition noch Platz sein kann, während die Partition eines anderen Verzeichnisses schon voll ist. Planen Sie daher vorher mit Bleistift und Papier ihre Partitionen, denn eine nachträgliche Größenänderung ist immer mit Problemen und großem Zeitaufwand verbunden.

Im Gegensatz zu DOS/Windows merkt der Benutzer nichts von der Aufteilung, da es bei Linux keine Laufwerksbuchstaben gibt, sondern die Laufwerke bzw. Partitionen in den Dateibaum eingebunden werden.

Kleine Plattenkapazität

Besitzt das System nur wenig Plattenplatz, so sollten möglichst wenig Partitionen angelegt werden, da jede dieser Partitionen einen ungenutzten Restplatz vorhalten muß.

Beispiel
Sie wollen Linux als Router auf einem alten Pentium-Rechner mit einer 1,2 GB großen Festplatte und 48 MB RAM installieren. Da der Festplattenplatz begrenzt ist, sollten sie die Platte wie folgt aufteilen:

`/boot` Eine 50 MB große Partition stellt sicher, daß alle Kernels unter der Zylinder-1024-Grenze liegen.

`swap` Eine 100 MB große Partition (96 MB) für den Swap-Speicher.

`/` Der Rest der Platte (1050 GB) wird für eine einzige große Wurzel-Partition verwendet, die alle anderen Verzeichnisse enthält.

Sollte sich die ganze Platte unter Zylinder 1024 befinden, dann kann für das Verzeichnis `/boot` auf eine eigene Partition verzichtet werden.

Auf älteren Systemen kann es auch vorkommen, daß mehrere kleinere Platten vorhanden sind. In diesem Fall kann der Verzeichnisbaum auf diese Platten aufgeteilt werden. So kann z. B. für das Verzeichnis `/home` eine eigene Platte verwendet werden. Die Verwendung der Platten ist genau so wie die Verwendung von mehreren Partitionen auf einer Platte.

Große Plattenkapazität

Bei Servern, die mehrere Dutzend GB an Daten speichern müssen, ist natürlich genügen Platz für viele Partitionen. Datensicherung und benötigte Dateisysteme geben dabei die Verwendung der Partitionen vor.

Beispiel
Wenn wir von einem NFS-Server ausgehen, der mit 512 MB RAM und einem Plattenarray mit einer Gesamtkapazität von 100 GB ausgestattet ist, dann wäre diese Aufteilung empfehlenswert.

`/boot` Eine 50 MB große Partition stellt sicher, daß alle Kernels unter der Zylinder-1024-Grenze liegen.

`swap` 512 MB (bzw. 4 x 128 MB) reichen für die Swap-Partition aus, da der Server extra mit viel Speicher ausgestattet wurde um Plattenzugriffe zu verhindern. Eventuell kann die Größe sogar auf 256 MB reduziert werden.

`/` Da möglichst alle wichtigen Verzeichnisse eine eigene Partition bekommen sollen, reichen hier 100 MB aus.

`/usr` Hier sind 2 GB vorgesehen. Praktischerweise können die hier vorhandenen Programme mit den Workstations über NFS geteilt werden.

`/var` 500 MB reichen aus um die Logdateien zu speichern. Auf eine eigene Partition beschränkt, können sie das System bei zu großem und schnellem Wachstum nicht zumüllen.

`/tmp` Für temporäre Aufgaben sind 100 MB an Plattenplatz ausreichend. Auch hier verhindert eine seperate Partition das Überwuchern des Systems.

`/home` Die restlichen 97 GB stehen dann für die Datenspeicherung übers Netzwerk zur Verfügung.

Verzeichnisse sind nicht auf lokale Partitionen beschränkt. NFS-Netzwerkverzeichnisse können die gleichen Aufgaben übernehmen. Früher, als Plattenplatz noch richtig teuer war, wurde oft das Verzeichnis `/usr` auf den Server ausgelagert. Heute ist dies nicht mehr so wichtig. Allerdings erleichert ein gemeinsames `/usr`-Verzeichnis für alle Arbeitsrechner die Wartung (Update) des Systems.

Neben diesen Argumenten kann auch die Datensicherung über die Größe der Partitionen entscheiden. So beschränkt z. B. die Größe der Sicherungsmedien unter Umständen die maximale Größe einer Partition ein.

10.1.3 Swap-Partition

Arbeitsspeicher ist auch heute noch teuer. Deshalb benötigen gerade bei Multitasking-Systemen die Programme oft mehr Speicher als vorhanden ist. Um dem gegenzuwirken arbeiten die Betriebssysteme nicht mit dem physikalischen Arbeitsspeicher sondern mit dem virtuellen Arbeitsspeicher. Der virtuelle Arbeitsspeicher setzt sich aus dem RAM und einem Speicherbereich auf der Festplatte zusammen. Bei Linux wird der Arbeitsspeicher auf der Festplatte durch die Swap-Partition realisiert. Benötigt das Betriebssystem mehr Speicher als vorhanden ist, so lagert es Teile des Arbeitsspeichers auf die Swap-Partition aus.

Dieses Verfahren zur Vergrößerung des Arbeitsspeichers wird als *Paging* bezeichnet. Wenn der physikalischer Speicher knapp wird, werden Memory-Pages (Speicherseiten) auf die Festplatte ausgelagert. Diese Speicherseiten haben meistens eine Größe von 4096 Bytes also 4 KByte. Bei Bedarf werden dann die ausgelagerten Seiten wieder in den physikalischen Arbeitsspeicher geladen.

Natürlich kann ein solcher Swap-Speicher keinen physikalischen Speicher ersetzen. Der Zugriff auf die Festplatte ist im Schnitt um den Faktor 10.000 langsamer als der Zugriff auf RAM-Speicher. Seine gute Funktion zeigt der Auslagerungspeicher, wenn mehrere Programme gleichzeitig laufen. Im Zeitalter der GUI-Programme belegen laufende Programme oft nur Speicher, während Sie die wenig belasten. Dies liegt daran, daß sie die meiste Zeit nur darauf warten, daß der Benutzer endlich etwas tut. Beim Wechsel zwischen den Programmen merkt man deutlich die Verzögerung beim Schreibung- und Rückschreiben der Speicherseiten. Die eigentlich Programmausführung des aktiven Programms ist aber genau so schnell wie auch sonst, da sich seine Speicherseiten im physikalischen Arbeitsspeicher befinden.

Für die Größe der Swap-Partition gibt es eine Faustregel. Der Auslagerungsspeicher sollte immer doppelt so groß sein, wie der eingebaute Arbeitsspeicher. Dabei sollte eine Größe von 16 MB für die Swap-Partitione nicht unterschritten werden um eine gute Funktion des Betriebssystems zu gewährleisten.

Früher war die maximale Größe der Swap-Partition auf 128 MB beschränkt. Allerdings können bis zu 16 Swap-Partitionen eingerichtet werden, so daß eine Gesamtkapazität von 2 GB für die Auslagerungsdatei zur Verfügung stand. Seit Kernel 2.1.117 darf die Swap-Partition bis zu 2 GB groß werden. Die Anzahl der Swap-Partitionen kann bei der Kompilierung des Kernels festgelegt werden. Der entsprechende Eintrag beim Kernel 2.4.18[1] in der Kernel-Headerdatei `/usr/src/linux-2.4.18/include/linux/swap.h` lautet:

```
#define MAX_SWAPFILES 32
```

[1] Nach den Kernelquellen des bei SuSE 8.0 mitgelieferten Originalkernels sowie des SuSE-Kernels.

Danach sollten bis zu 32 Swap Partitionen bei diesem Kernel möglich sein. Maximal unterstützt der Kernel 64 Swap-Partitionen.

Um die Zugriffsgeschwindigkeit zu erhöhen ist es sinnvoll die Swap-Partitionen auf mehrere Platten zu verteilen. Da die einzelnen Partitionen dann zu einem logischen Bereich zusammengezogen werden, bilden sie praktisch ein RAID Level 0. Bei IDE-Platten ist es dann noch von Vorteil, wenn beide Platten an einem seperaten Controller hängen.

Wenn Sie meinen, sie haben genügend Speicher, dann können Sie auch mit der Formel "eins zu eins" arbeiten. Dies ist vor allem bei Serversystemen sinnvoll, die mit extra viel Speicher bestückt worden sind um Festplattenzugriffe zu verringern.

Wenn Sie allerdings meinen zu wenig Platz zu haben, dann können Sie die Swap-Partition auch auf die dreifache Arbeitsspeichergröße setzen.[2]

Sollten Sie auf der Festplatte nicht genügend Speicher oder Partitionsplatz besitzen, so können Sie auch eine Swap-Datei (*swap file*) einrichten. Dies ist allerdings nicht so effektiv wie eine seperate Partition. Sollte ihr Rechner 4 MB oder weniger Arbeitsspeicher besitzen, so benötigen Sie auf jeden Fall eine Swap-Partition.

Beispiele
Ihr System besitzt 8 MB Arbeitsspeicher. Sie legen daher eine Swap-Partition von 16 MB an. Der zur Verfügung stehende virtuelle Speicher liegt dann bei 24 MB.

Sie besitzen einen Rechner mit 128 MB RAM. Nach der Formel sollten Sie einen Swap-Bereich von 256 MB einrichten. Da eine Swap-Partition aber nur 128 MB groß sein darf, legen Sie zwei Partitionen mit jeweils 128 MB an.

Ein Datei-Server wurde für den schnellen Dateizugriff mit 512 MB RAM bestückt. Da Zugriffe auf die Festplatte möglichst vermieden werden sollen, reichen vier Swap-Partitionen á 128 MB.

Ihr alter 486er Rechner soll als Router eingesetzt werden. Da er nur 4 MB RAM besitzt, müssen Sie eine Swap-Partition einrichten um Linux installieren zu können. Dabei sollte die Swap-Partition mindestens eine Größe von 16 MB aufweisen.

10.1.4 fips

Wenn Sie Linux auf einem Rechner installieren wollen, auf dem schon ein Betriebssystem läuft, stoßen Sie meistens auf das Problem, daß kein Platz mehr auf der Festplatte für eine seperate Partition mehr ist. Für diesen Fall gibt es das Programm `fips`. Es ermöglicht die Aufteilung einer FAT16-Partition in zwei Teile. Dabei sollte es zu keinem Datenverlust kommen, trotzdem sollte die Daten mit einem Backup gesichert werden. Vor dem Einsatz von `fips`[3] sollten Sie außerdem genau die Dokumentation lesen.

Kopieren Sie die Dateien `fips.exe` und `restorbb.exe` auf eine Diskette. Eine Kopie Ihres alten MBR (*Master Boot Record*) wird während der Installation auf die Diskette kopiert. Dieses Backup können Sie dazu nutzen die Veränderung wieder rückgängig zu machen. Daher Diskette beschriften und gut verwahren!!!

Gleichzeitig wird der alte MBR auf dem Linux-System als `/boot/boot.0300` für IDE-Platten und als `/boot/boot.0800` für SCSI-Platten abgelegt.

10.1.5 fdisk

Um Partitionen einzurichten wird das Programm `fdisk` verwendet. Sie sollten sich bei der Arbeit mit `fdisk` immer Notizen machen.

```
fdisk [-l] [-u] [PLATTE]
fdisk -s PARTITION
```

[2] Bekannt sind alle drei Regeln (einfacher, zweifacher oder dreifacher Arbeitsspeicher). Je nach Autor finden Sie unterschiedliche Angaben.

[3] Download von `http://sunsite.unc.edu/pub/Linux/system/install`.

Optionen

`-u`	Größenangabe in Sektoren anstatt in Zylindern
`-l`	Ausgabe der Partitionstabelle
`-s`	Ausgabe der Größe der angegebenen Partition

Ein Zylinder (*cylinder*) ist die Menge aller Spuren, die sich an der gleichen Stelle auf jeder Platte befinden. Als Spur (`track`) bezeichnet man die konzentrischen Kreise auf einer Platte der Festplatte.

Sie können Linux auf jeder Festplatte und jeder primären Partition installieren. Wenn Sie keine Festplatte angeben, wird standardmäßig das Gerät `/dev/hda` genommen.

Wenn Sie `fdisk` benutzen, können Sie mit der Taste `<m>` sich eine Übersicht über die Hilfe anzeigen lassen.

```
root@defiant:/ > fdisk /dev/hda

Kommando (m für Hilfe): m
Kommando  Bedeutung
   a   (De)Aktivieren des bootbar-Flags
   b   »bsd disklabel« bearbeiten
   c   (De)Aktivieren des DOS Kompatibilitätsflags
   d   Eine Partition löschen
   l   Die bekannten Dateisystemtypen anzeigen
   m   Dieses Menü anzeigen
   n   Eine neue Partition anlegen
   o   Eine neue leere DOS Partitionstabelle anlegen
   p   Die Partitionstabelle anzeigen
   q   Ende ohne Speichern der Änderungen
   s   Einen neuen leeren »Sun disklabel« anlegen
   t   Den Dateisystemtyp einer Partition ändern
   u   Die Einheit für die Anzeige/Eingabe ändern
   v   Die Partitionstabelle überprüfen
   w   Die Tabelle auf die Festplatte schreiben und das Programm beenden
   x   Zusätzliche Funktionen (nur für Experten)

Kommando (m für Hilfe):
```

Um nun eine Partition anlegen zu können, müssen Sie drei Schritte durchlaufen.

1. Größe der Partition angeben.
2. Den Typ der Partition festlegen.
3. Die Veränderung in die Partitionstabelle schreiben.

Für jede Festplatte, die partitioniert werden soll, muß `fdisk` einmal gestartet werden.

Wenn Sie logische Laufwerke anlegen wollen, müssen Sie nach der Einrichtung der erweiterten Partition den Rechner erst wieder neu starten. Mit `fdisk` wird nämlich nur die Partitionstabelle geschrieben. Das Einrichten der Partitionen erfolgt erst beim Neustarten des Rechners.

Denken Sie daran, `fdisk` für Linux erstellt oder löscht nur Partitionen für das Linux-Betriebssystem. Andere Betriebssysteme können diese Partitionen meistens nicht sehen oder mit Ihnen arbeiten.

Nachdem Sie die Partitionen angelegt haben, muß die Swap-Partition aktiviert werden, die für die Wurzel vorgesehene Partition ausgewählt und die Partitionen formatiert werden.

Beispiele

Der folgende Befehl zeigt den Zustand der Festplatte `/dev/hda` an.

```
root@defiant:/ > fdisk -l

Festplatte /dev/hda: 255 Köpfe, 63 Sektoren, 592 Zylinder
```

```
Einheiten: Zylinder mit 16065 * 512 Bytes

    Gerät boot.  Anfang      Ende    Blöcke   Id  Dateisystemtyp
/dev/hda1   *         1       261   2096451    6  FAT16
/dev/hda2           262       392   1052257+   5  Erweiterte
/dev/hda3           393       576   1477980   83  Linux
/dev/hda4           577       592    128520   82  Linux Swap
/dev/hda5           262       392   1052226    6  FAT16
```

Wenn das Program `fdisk` läuft, zeigt der Befehl `l` zeigt die Liste der Dateisystemtypen.

```
 0  Leer            1c  Verst. Win95 FA 65  Novell Netware  bb  Boot Wizard hid
 1  FAT12           1e  Verst. Win95 FA 70  DiskSecure Mult c1  DRDOS/sec (FAT-
 2  XENIX root      24  NEC DOS         75  PC/IX           c4  DRDOS/sec (FAT-
 3  XENIX usr       39  Plan 9          80  Old Minix       c6  DRDOS/sec (FAT-
 4  FAT16 <32M      3c  PartitionMagic  81  Minix / old Lin c7  Syrinx
 5  Erweiterte      40  Venix 80286     82  Linux Swap      da  Non-FS data
 6  FAT16           41  PPC PReP Boot   83  Linux           db  CP/M / CTOS / .
 7  HPFS/NTFS       42  SFS             84  OS/2 verst. C:- de  Dell Utility
 8  AIX             4d  QNX4.x          85  Linux erweitert df  BootIt
 9  AIX bootfähig   4e  QNX4.x 2nd part 86  NTFS volume set e1  DOS access
 a  OS/2 Bootmanage 4f  QNX4.x 3rd part 87  NTFS volume set e3  DOS R/O
 b  Win95 FAT32     50  OnTrack DM      8e  Linux LVM       e4  SpeedStor
 c  Win95 FAT32 (LB 51  OnTrack DM6 Aux 93  Amoeba          eb  BeOS fs
 e  Win95 FAT16 (LB 52  CP/M            94  Amoeba BBT      ee  EFI GPT
 f  Win95 Erw. (LBA 53  OnTrack DM6 Aux 9f  BSD/OS          ef  EFI (FAT-12/16/
10  OPUS            54  OnTrackDM6      a0  IBM Thinkpad hi f0  Linux/PA-RISC b
11  Verst. FAT12    55  EZ-Drive        a5  FreeBSD         f1  SpeedStor
12  Compaq Diagnost 56  Golden Bow      a6  OpenBSD         f4  SpeedStor
14  Verst. FAT16 <3 5c  Priam Edisk     a7  NeXTSTEP        f2  DOS secondary
16  Verst. FAT16    61  SpeedStor       a9  NetBSD          fd  Linux raid auto
17  Verst. HPFS/NTF 63  GNU HURD / SysV b7  BSDI fs         fe  LANstep
18  AST SmartSleep  64  Novell Netware  b8  BSDI swap       ff  BBT
1b  Verst. Win95 FA
```

Mit `fdisk` kann der Dateisystemtyp einer Partition unter dem Menüpunkt `t` nachträglich geändert werden. Der Dateisystemtyp wird auch beim Formatieren in der Partitionstabelle eingetragen. Wichtig ist diese Funktion dann, wenn eine zerstörte Partitionstabelle rekonstruiert werden soll.

10.2 Mounten

Um mit Dateien zu arbeiten, müssen diese erreichbar sein. Da jede Partition und jedes Speichergerät sein eigenes Dateisystem besitzt, müssen diese an einer zentralen Stelle zusammengeführt werden. Bei Windows wird dies durch die Laufwerksbuchstaben realisiert, die unter *Arbeitsplatz* liegen. Bei Linux muß eine Partition immer als Wurzel `/` ansprechbar sein. Die anderen Partitionen werden dann in das Verzeichnissystem dieser Partition eingebunden. Das Verzeichnis, das die Wurzel des jeweiligen Dateisystems im Verzeichnissystem repräsentiert, wird als **Mount Point** bezeichnet und der Vorgang des Einbindens als **Mounten**. Dabei ist das Einbinden nicht nur auf die Linux-Dateisysteme wie ext2 beschränkt. Es können auch virtuelle Dateisysteme wie `/proc` eingebunden werden oder sogar Dateisysteme, die sich auf anderen Rechnern im Netz befinden. Eine Liste der möglichen Dateisysteme finden Sie im Abschnitt 10.3.

10.2.1 mount

Der `mount`-Befehl bindet ein Dateisystem in den aktuellen Verzeichnisbaum ein.

```
mount [OPTIONEN] [GERÄT] MOUNTPOINT
```

Option	Aktion
`async`	Der I/O-Zugriff erfolgt asyncron
`atime`	Der letzte Zugriff wird in der Inode festgehalten
`auto`	Kann mit der Option `-a` gemountet werden
`defaults`	Entspricht: `rw`, `suid`, `dev`, `exec`, `auto`, `nouser` und `async`.
`dev`	Interpretiert spezielle Zeichen- und Block-Geräte
`exec`	Binärdateien können ausgeführt werden
`noatime`	Der letzte Zugriff wird nicht in der Inode festgehalten (schnellerer Zugriff)
`noauto`	Wird nicht automatisch gemountet
`nodev`	Interpretiert nicht spezielle Zeichen- und Block-Geräte
`noexec`	Binärdateien werden nicht ausgeführt
`nosuid`	Das `suid`- oder `sgid`-bit wird nicht ausgeführt
`nouser`	Ein Benutzer kann das Dateisystem nicht mounten
`remount`	Mountet ein Dateisystem erneut (Wird verwendet um Modifikatoren zu ändern)
`ro`	Das Dateisystem wird nur lesbar gemountet (read only)
`rw`	Das Dateisystem wird schreibbar gemountet (read write)
`suid`	Erlaubt die Benutzung des `suid`- oder `sgid`-bits
`sync`	Alle E/A-Aktionen sollten syncron erfolgen
`user`	Ein Benutzer darf das Dateisystem mounten

Tabelle 10.2: Modifikatoren für `mount -o` und `/etc/fstab`

Optionen

`-a`	Mounte alle Dateisysteme, die in `/etc/fstab` aufgeführt werden.
`-f`	Überprüfe ob das angegebene Dateisystem gemountet werden kann
`-n`	Schreibe die Mount-Informationen nicht in die Datei `/etc/mtab`
`-o OPTION`	Modifikatoren für den Mountvorgang (siehe Tabelle 10.2)
`-r`	Mounten nur mit Leseberechtigung (read only)
`-t DSTYP`	Typ für das zu mountende Dateisystems
`-v`	Angabe der Mount-Informationen
`-w`	Mounten mit Schreibberechtigung (Standard)

Der Aufruf von `mount` ohne Parameter zeigt die Liste aller Dateisysteme an, die in diesem Moment gemountet sind.

```
tapico@defiant:~ > mount
/dev/hda3 on / type ext2 (rw)
proc on /proc type proc (rw)
/dev/hda1 on /c type vfat (rw)
/dev/hda5 on /d type vfat (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=0620)
/dev/fd0 on /a type vfat (rw,noexec,nosuid,nodev,user=tapico)
```

Beim Einbinden eines Dateisystems mit `mount` wird die Datei `/etc/fstab` ausgewertet. Wenn das Gerät in der Datei angegeben ist, so reicht für das Mounten die Angabe des Mount Points aus.
`mount /floppy`
Außerdem werden die Angaben für dieses Gerät aus der Datei entnommen. Änderungen der gegebenen Modifikatoren kann über den Schalter `-o` erfolgen.

Beispiel
Ist der Mount Point in der `/etc/fstab` eingetragen, so reicht ein
`mount /a`
aus.

Ist dies nicht der Fall, so kann das Gerät über
`mount -t vfat /dev/fd0 /a`
gemountet werden. Dies ist aber nur **root** erlaubt.

10.2.2 umount

Der `umount`-Befehl[4] entfernt ein Dateisystem aus dem aktuellen Verzeichnisbaum.

```
umount [OPTIONEN] [GERÄT] [MOUNTPOINT]
```

Optionen	
`-a`	Unmounte alle Dateisysteme, die in `/etc/mtab` aufgeführt werden.
`-n`	Schreibe die Mount-Informationen nicht in die Datei `/etc/mtab`
`-t DSTYP`	Typ für das zu unmountende Dateisystems

Durch den Befehl `umount` wird auch der Befehl `sync` aufgerufen, damit vor dem Entfernen des Dateisystems dieses mit dem Cache abgeglichen wird.

10.2.3 sync

Durch den Befehl `sync` werden die sich im Cache befindlichen Daten auf die Platte geschrieben und damit die Festplatte auf den aktuellen Stand gebracht.

```
sync
```

Der `sync`-Befehl ruft einfach die Kernel-Prozedur `sync` auf. Diese wird auch vom Befehl `umount` zum sauberen Aushängen von Dateisystemen benutzt

10.2.4 /etc/fstab

Die Datei `/etc/fstab` wird von den Befehlen `fsck`, `mount` und `umount` benutzt.

```
tapico@defiant:~ > cat /etc/fstab
/dev/hda3   /                 ext2          defaults         1   1
/dev/hda1   /c                vfat          defaults         0   0
/dev/hda5   /d                vfat          defaults         0   0
/dev/hda4   swap              swap          defaults         0   0
/dev/hdc    /cdrom            iso9660       ro,noauto,user   0   0
/dev/fd0    /floppy           auto          noauto,user      0   0
/dev/fd0    /a                vfat          noauto,user      0   0
proc        /proc             proc          defaults         0   0
```

Die Spalten bedeuten von links nach rechts:

1. Der physikalische Ort des Dateisystems oder Blockgeräts.
2. Der Mount Point. An dieser Stelle wird die Wurzel des Dateisystems in die Verzeichnisstruktur eingebaut.
3. Der Typ des Dateisystems
4. Die Optionen, die beim Mounten des Systems benutzt werden.
5. Die Nummer legt fest, ob das Dateisystem bei einem Backup durch `dump` gesichert werden soll.
6. Die Nummer legt fest, in welcher Reihenfolge `fsck` die Dateisysteme prüft.

[4] Das "n" in `unmount` ist im Lauf der Zeit aus dem Befehl verschollen.

10.2.5 /etc/mtab

In dieser Datei werden alle gerade gemounteten Dateisysteme aufgelistet. Der Befehl `mount` ohne Parameter benutzt diese Datei für seine Ausgabe. Die Befehle `mount` und `umount` verändern die Datei.

```
tapico@defiant:~ > cat /etc/mtab
/dev/hda3 / ext2 rw 0 0
proc /proc proc rw 0 0
/dev/hda1 /c vfat rw 0 0
/dev/hda5 /d vfat rw 0 0
devpts /dev/pts devpts rw,gid=5,mode=0620 0 0
/dev/fd0 /a vfat rw,noexec,nosuid,nodev,user=tapico 0 0
```

Die Ähnlichkeit mit der Ausgabe des Befehls `mount` ist nicht zu übersehen.

10.3 Das Dateisystem

Das Linux-Dateisystem organisiert die Dateien und Verzeichnisse in einer Hierachie. Es ermöglicht das Speichern von und den zufälligen Zugriff auf Dateien, egal ob sich diese Daten auf der Festplatte, der Floppy, einer CD-ROM oder im Netzwerk befinden. Linux unterstützt mehrere Dateisysteme. Einige von ihnen werden in der folgenden Liste aufgezählt.

- `ext2` - Second Extended Filesystem, das aktuelle Standard-Dateisystem für LINUX
- `ext3` - Third Extended Filesystem, das neue Dateisystem für LINUX mit Journaling-Funktionen
- `bfs` - Das SCO BFS Dateisystem.
- `coherent` - Dateisystem der UNIX-Version Coherent (war lange Jahre eine kostengünstige Alternative zu kommerziellen UNIX-Produkten). Wird wohl in Zukunft nicht mehr unterstützt werden.
- `ffs` - Fast Filesystem, das Dateisystem vom Amiga.
- `hpfs` - High Performance Filesystem, das Dateisystem von OS/2. Es kann zur Zeit nur lesend genutzt werden.
- `iso9660` - ISO-9660-Dateisystem, das für die meisten CD-ROMs benutzt wird.
- `jfs` - Das Journaling-Filesystem von IBM.
- `minix` - Minix Dateisystem, Vorläufer von ext. War das ursprüngliche Dateisystem von LINUX.
- `msdos` - Das FAT16-Dateisystem von MS-DOS.
- `ncpfs` - Das Dateisystem der Novell-Server.
- `nfs` - Network File System (NFS). Gestattet den Zugriff auf Dateien via Netzwerk.
- `ntfs` - Dateisystem für Windows NT (New -ähem- Technology). Windows 2000/XP im Moment leider nur lesend.
- `proc` - Das virtuelle Dateisystem, daß der Linux-Kernel für die Verwaltung der Prozesse verwendet.
- `reiserfs` - Das Reiser-Filesytem mit Journaling-Funktion
- `swap` - Das Linux Swap Dateisystem.
- `sysv` - Dateisystem gemäß System V.
- `ufs` - Uniform Filesystem, das Dateisystem, was BSD, SunOS und NeXTstep verwenden. Es wird nur lesend unterstützt.

- `umsdos` - Dateisystem, mit dessen Hilfe LINUX in einer DOS-Partition installiert werden kann. Es ist allerdings sehr langsam.
- `vfat` - Virtual FAT Filesystem, die Erweiterung der FAT für die Verwendung langer Dateinamen.
- `xenix` - Dateisystem von Xenix (eine Entwicklung von Microsoft für den PC. Wurde in späteren Jahren von der Firma SCO übernommen. Xenix war jahrelang das Standard-UNIX-System für den PC). Wird wohl in Zukunft nicht mehr unterstützt werden.

Seit Kernel 2.1.21 sind diese Dateisysteme nicht mehr dabei.

- `ext` - Extended Filesystem, ein Vorläufer von ext2
- `xiafs` - Ein nach seinem Entwickler benanntes Dateisystem, das ext2 sehr ähnlich ist.

Nachdem Sie also die Partitionen angelegt haben, muß die Partition noch formatiert werden. Dafür wird der Befehl `mkfs` verwendet.

10.3.1 mkfs

Das Kommando `mkfs` formatiert eine Partition mit dem angegebenen Dateisystem.

```
mkfs [-t DSTYP] GERÄT [BLOCKS]
```

Optionen

`-t DSTYP`	Legt den Dateisystemtyp fest
`-v`	Zeigt alle Kommandos an
`-c`	Überprüft nach defekten Blöcken vorm Anlegen des Dateisystems
`-l DATEINAME`	Benutzt die angegeben Datei als Liste für bekannte defekte Blöcke

Normalerweise ruft `mkfs` andere Kommandos auf, die für das jeweilige mit `-t` angegebene Dateisystem programmiert wurden. Z. B. sind dies `mkfs.ext2`, `mkfs.msdos` und `mkfs.minix`. Sie können diese Programme auch direkt nutzen. Natürlich darf für diesen Vorgang der Datenträger nicht gemountet sein. Normalerweise kann `mkfs` nur von dem Benutzer *root* ausgeführt werden. Um normalen Benutzern trotzdem das Formatieren von Disketten zu erlauben, muß *root* ein passendes Skript (9.1.4) zur Verfügung stellen.

Beispiele
Der folgenden Befehl erstellt das ext2-Dateisystem auf der dritten Partition der ersten Festplatte.
`mkfs -t ext2 /dev/hda3 1477980`
Heute wird die Anzahl der Blöcke automatisch ermittelt. Es kann aber `mkfs`-Versionen geben, die die Angabe der Blöcke verlangt. Mit dem Befehl `fdisk` kann diese ermittelt werden. Dabei muß man sehr vorsichtig sein, da falsche Angaben zu einem Datenverlust auf einer anderen Partition führen kann.

10.3.2 fdformat

Für die Formatierung der Disketten kann auch das Kommando `mkfs` verwendet werden. Allerdings sollte man vorher mit `fdformat` eine Low-Level-Formatierung durchführen, die die Sektoren und Spuren auf der Diskette anlegt.

```
fdformat [OPTIONEN] GERÄT
```

Dabei wird für das Diskettenlaufwerk A: die Bezeichnung `/dev/fd0` und für Laufwerk B: die Bezeichnung `/dev/fd1` verwendet. Es können über andere Gerätenamen verschiedene Formatierungen durchgeführt werden. Hier eine Liste von Gerätenamen[5] für das Laufwerk A:.

[5] Eine Erläuterung zu den Geräten finden Sie in den Kernelquellen in der Datei `/usr/src/linux/Documentation/devices.txt`.

```
root@defiant:/dev > ls fd0*
fd0        fd0h1440  fd0h360  fd0h880   fd0u1600  fd0u1760  fd0u3200  fd0u720
fd0CompaQ  fd0h1476  fd0h410  fd0u1040  fd0u1680  fd0u1840  fd0u3520  fd0u800
fd0d360    fd0h1494  fd0h420  fd0u1120  fd0u1722  fd0u1920  fd0u360   fd0u820
fd0h1200   fd0h1600  fd0h720  fd0u1440  fd0u1743  fd0u2880  fd0u3840  fd0u830
```

Ein normaler Benutzer kann allerdings nur das Standardformat (`/dev/fd0` und `/dev/fd1`) für die Formatierung verwenden. Außerdem darf das Laufwerk nicht gemountet sein, was bei einer vorformatierten Diskette durchaus möglich ist.

10.3.3 Swap-Space einrichten

Der Auslagerungsspeicher kann als Swap-Partition oder als Swap-Datei angelegt werden.

Anlegen der Swap-Partition

Um eine zusätzliche Swap-Partition einzurichten müssen Sie auf der Festplatte eine neue Partition anlegen. Dies können Sie mit dem Befehl `fdisk` (10.1.5) erledigen.

Nehmen wir einfach mal an, daß Sie eine zweite Festplatte (`/dev/hdb`) in Ihren Rechner eingebaut haben. Um die Swap-Performance zu verbessern, sollten Sie auf dieser zweiten Platte auch eine Swap-Partition einrichten. Starten Sie also das Programm `fdisk`.

```
enterprise:~ # fdisk /dev/hdb

The number of cylinders for this disk is set to 5005.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Befehl (m für Hilfe):
```

Mit dem Befehl `n` legen wir nun eine neue Partition an. Und zwar eine primäre Partition mit der Größe von 256 MB.

```
Befehl (m für Hilfe): n
Befehl  Aktion
   l      Logische Partition (5 oder größer)
   p      Primäre Partition (1-4)
p
Erster Zylinder (1-5005) [Standardwert: 1]:
Benutze den Standardwert 1
Letzter Zylinder oder +Größe, +GrößeK oder +GrößeM (1-5005) [Standardwert: 5005]: +256M
```

Mit dem Befehl `p` überprüfen wir den Eintrag in der Partitionstabelle und notieren uns die Werte.

```
Befehl (m für Hilfe): p

Festplatte /dev/hdb: 255 Köpfe, 63 Sektoren, 5005 Zylinder
Einheiten: Zylinder mit 16064 * 512 Bytes

    Gerät boot.  Anfang      Ende    Blöcke   Id  Dateisystemtyp
/dev/hdb1             1        34    273088   82  Linux Swap
```

Nun brauchen wir nur noch mit `w` die Partitionstabelle zu schreiben und das Programm zu verlassen. Danach muß der Rechner neu gestartet werden, damit die neue Partitionstabelle übernommen wird.

Anlegen einer Swap-Datei

Um eine Swap-Datei zu erstellen, müssen Sie eine Datei öffnen und soviele Bytes hineinschreiben, wie die Swap-Datei groß sein soll. Hierfür können Sie gut den Befehl `dd` (4.5.4) verwenden.

```
enterprise:~ # dd if=/dev/zero of=/swap bs=1024 count=16384
16384+0 Records ein
16384+0 Records aus
enterprise:~ # ls -l /swap
-rw-r--r--    1 root     root     16777216 Nov 17 12:15 /swap
```

Das Gerät `/dev/zero` ist praktisch das Gegenteil von `/dev/null`. Während `/dev/null` praktisch ein WOM (Write Only Memory) ist, der alle Informationen schluckt und Sie nicht wieder hergibt, ist `/dev/null` ein unendlich großer Datenträger, auf dem aber nur Nullen stehen. Der obige Befehl liest praktisch 16384 Blöcke mit 1024 Bytes von diesem Gerät und schreibt Sie in die Datei `/swap`.

Nach dem Anlegen der Datei sollten Sie sicherheitshalber mit dem Befehl `sync` (10.2.3) die Dateisysteme wieder synchronisieren.

Auf keinen Fall darf die Swap-Datei eine Lücke besitzen. Die Blöcke der Datei müssen sich hintereinander auf der Platte befinden.

Swap-Bereich formatieren

Nach dem Anlegen der Partition bzw. der Datei muß diese für den Einsatz als Auslagerungsspeicher vorbereitet werden, d. h. formatiert werden. Hier kommt der Befehl `mkswap` zum Einsatz.

```
mkswap -c GERÄTENAME
```

`GERÄTENAME` steht für den Namen der Partition bzw. der Datei. Der Schalter `-c` ist optional und bewirkt, daß der Swap-Bereich bei der Formatierung auf fehlerhafte Blöcke untersucht wird.

Nehmen wir als Beispiel mal die oben angelegte Swap-Partition. Mit dem folgenden Befehl wird sie als Swap-Bereich formatiert.

```
enterprise:~ # mkswap -c /dev/hdb1
Swapbereich Version 1 mit der Größe 279642112 Bytes wird angelegt
```

Die Swap-Datei wird analog angelegt. Nur daß anstatt des Gerätenamens, der Dateiname angegeben wird.

```
enterprise:~ # mkswap -c /swap
Swapbereich Version 1 mit der Größe 16773120 Bytes wird angelegt
enterprise:~ # sync
```

Nach dem Einrichten einer Swap-Datei sollten Sie immer das Dateisystem mit `sync` synchronisieren um sicherzugehen, daß die Informationen auch physikalisch auf die Platte übertragen worden sind. Dies ist bei der Swap-Partition natürlich nicht nötig.

Swap-Bereich aktivieren und deaktivieren

Damit der neue Swap-Bereich auch vom System benutzt wird, muß er aktiviert werden. Dafür ist der Befehl `swapon` verantwortlich. Analog dazu können Sie natürlich mit dem Befehl `swapoff` einen Swap-Bereich auch deaktivieren. Anhand des Befehls `free` (12.6.4) kann die Veränderung des Swap-Bereichs beobachtet werden.

Als Beispiel schauen wir uns doch mal die Einbindung unserer Swap-Datei ins System an.

```
enterprise:~ # free
             total       used       free     shared    buffers     cached
Mem:        513448     466992      46456          0      63500     258672
-/+ buffers/cache:     144820     368628
Swap:      1028120          0    1028120
enterprise:~ # swapon /swap
enterprise:~ # free
             total       used       free     shared    buffers     cached
Mem:        513448     467032      46416          0      63528     258672
-/+ buffers/cache:     144832     368616
Swap:      1044496          0    1044496
enterprise:~ # swapoff /swap
enterprise:~ # free
             total       used       free     shared    buffers     cached
Mem:        513448     467036      46412          0      63536     258672
-/+ buffers/cache:     144828     368620
Swap:      1028120          0    1028120
```

Auf jeden Fall sollten Sie eine Swap-Datei nur dann löschen, wenn Sie sie vorher mit `swapoff` deaktiviert haben.

Wie üblich ist dieser Befehl nicht von Dauer. Nach einem Reboot ist der Swap-Bereich wieder inaktiv. Die Swap-Bereich müssen also bei jedem Start initalisiert werden. Dies geschieht mit dem Befehl `swapon -a` in einer der Startdateien.

`swapon -a` liest die Informationen aus der Datei `/etc/fstab` aus. Wenn dort Informationen des Typs

```
/dev/hda6       swap    swap    sw 0 0
```

bzw.

```
/dev/hdb1       swap    swap    pr=42 0 0
```

dann werden die jeweiligen Swap-Bereiche beim Booten initialisiert. Sie sollten für jeden neu angelegten Swap-Bereich einen Eintrag zur `/etc/fstab` hinzufügen.

10.3.4 mkswap

Das Kommando `mkswap` bereitet eine Partition oder eine Datei für den Einsatz als Swap-Bereich vor.

```
mkswap [OPTIONEN] GERÄT [GRÖSSE]
```

Der Parameter `GERÄT` kann die Bezeichnung für eine Partition (z. B. `/dev/hdb3`) oder der Name einer Datei sein.

Die Angabe der Größe des Swap-Bereichs ist heute nicht mehr nötig und existiert nur noch aus Kompatibilitätsgründen. Der Befehl nutzt den zur Verfügung stehenden Platz in der Partition bzw. in der angelegten Datei voll. Eine falsche Angabe der Größe kann zu Schäden auf der Festplatte führen. Also lieber Finger weg von diesem Parameter.

Optionen	
`-c`	Testen auf fehlerhafte Blocks vorm Formatieren (*check*)
`-f`	Erlaubt auch die Ausführung mit unsinnigen Parametern (*force*)
`-p GRÖSSE`	Größe der verwendeten Speicherseiten (*page*)
`-v0`	Erstellt einen Swap-Bereich im alten Format
`-v1`	Erstellt einen Swap-Bereich im neuen Format (Standard)

10.3.5 swapon

Das Kommando `swapon` initalisiert Geräte und Dateien für den Einsatz als Swap-Bereich. Normalerweise wird der Befehl in einer der Startdateien ausgeführt um den Swap-Bereich immer zur Verfügung zu stellen.

```
swapon [OPTIONEN]
swapon [OPTIONEN] SWAPBEREICH
```

Optionen	
`-h`	Hilfe (*help*)
`-V`	Versionsnummer (*version*)
`-s`	Übersicht über die Swap-Bereiche (*summary*)
`-a`	Alle Swap-Bereiche aus `/etc/fstab` mit der Option `sw` bzw. `pri` werden initialisiert. (*all*)
`-p PRIORITÄT`	Gibt die Priorität des Swap-Bereichs an (Wert zwischen 0 und 32767), steht seit Kernel 1.3.2 zur Verfügung (*priority*)

Die Option sich eine Übersicht über die Swap-Bereiche anzeigen zu lassen steht nur dann zur Verfügung, wenn `/proc/swaps` existiert. Dies ist erst seit Kernel 2.1.25 der Fall.

```
enterprise:/home/ole # ls -l /proc/swaps
-r--r--r--    1 root     root            0 Nov 17 13:38 /proc/swaps
enterprise:/home/ole # cat /proc/swaps
Filename                        Type            Size    Used    Priority
/dev/hdb6                       partition       1028120 0       42
/swap                           file            16376   0       -2
enterprise:/home/ole # swapon -s
Filename                        Type            Size    Used    Priority
/dev/hdb6                       partition       1028120 0       42
/swap                           file            16376   0       -2
```

Um den Befehl jedesmal beim Starten ausführen zu können, wird er in einer der Startdateien des Systems eingetragen. Diese befinden sich meistens im Verzeichnis `/etc/rc.d`. Unter SuSE (8.0) ist `/etc/rc.d` ein Link auf `/etc/init.d`. Dort findet sich der Eintrag in den Dateien `boot.localfs` und `boot.swap`.

```
enterprise:/etc # ls -ld init.d rc.d
drwxr-xr-x   11 root     root         4096 Jul 15 21:54 init.d
lrwxrwxrwx    1 root     root            6 Mai  2  2002 rc.d -> init.d
enterprise:/etc # cd init.d/
enterprise:/etc/init.d # grep -l swapon * 2> /dev/null
boot.localfs
boot.swap
enterprise:/etc/init.d # grep -n swapon * 2> /dev/null
boot.localfs:29:        swapon -a &> /dev/null
boot.swap:26:   # .. this should work know with the new swapon behavio(u)r
boot.swap:29:   swapon -a &> /dev/null
```

10.3.6 swapoff

Wenn ein Swap-Bereich eingebunden werden kann, dann muß er auch aus dem System entfernt werden können. Dafür ist der Befehl `swapoff` verantwortlich.

```
swapoff [OPTIONEN]
swapoff SWAPBEREICH
```

Optionen

`-h`	Hilfe (*help*)
`-V`	Versionsnummer (*version*)
`-s`	Übersicht über die Swap-Bereiche (*summary*)
`-a`	Alle Swap-Bereiche aus `/etc/fstab` mit der Option `sw` bzw. `pri` werden aus dem System entfernt. (*all*)

10.4 Verwaltung

Linux unterscheidet folgende Dateiarten

- gewöhnliche Dateien (ordinary files)
- Verzeichnisse (directories)
- Gerätedateien (devices)
- Named Pipes (FIFOs)
- Sockets
- Verweise (Links)

Gewöhnliche Dateien sind die Dateien, die mit einer mehr oder weniger gewöhnlichen Anwendung erzeugt werden - der Text, der mit einer Textverarbeitung erstellt wird, die Daten einer Datenbank-, die Tabelle eines Kalkulationsprogramms oder auch die Zeilen eines Shell-Skripts, Tabellen, die von LINUX zu diversen Verwaltungszwecken benutzt werden, wie etwa die Paßwortdatei `/etc/passwd` und nicht zuletzt Programmdateien. Gewöhnliche Dateien sind solche Dateien, deren Inhalt für den einzelnen Anwender direkt oder indirekt von Interesse ist (am Text den man geschrieben hat ist man direkt interessiert, der Inhalt der Datei `/etc/passwd` ist für jeden Anwender deshalb von Belang, weil über ihn der Zugang zum System ermöglicht sowie ein Teil der persönlichen Arbeitsumgebung bereitgestellt wird).

Verzeichnisse ermöglichen eine strukturierte Dateiverwaltung. Die Organisation der Daten wird durch sie wesentlich vereinfacht. Verzeichnisse sind Tabellen mit den Namen von Dateien sowie einem Vermerk, über den das System den Zugriff auf die Daten in den Dateien steuert. Die Anzahl der gewöhnlichen Dateien ist schon in einem Einplatzsystem wie DOS oder Windows 95 so groß, daß ein Ordnungsprinzip erforderlich wird. So war ein wesentlicher Grund für die erste Überarbeitung des Betriebsystems DOS (Einführung der Version 2.0) die Verwendung von Festplatten für den IBM PC-XT. Diese ermöglichten die Speicherung großer Datenmengen. Die Hierarchie der DOS-Dateiverwaltung kannte bis dahin in Anlehnung an das System CP/M Laufwerke (Disketten!) und Dateien. Da nun die Anzahl der Dateien sehr hoch sein konnte, führten die Microsoft-Programmierer in DOS die Verzeichnisse als Verwaltungsinstrument hinzu. Somit kennt DOS drei Hierarchieebenen in der Dateiverwaltung: Laufwerk (aus CP/M geerbt), Verzeichnisse und - natürlich - gewöhnliche Dateien. Daß dabei die Verzeichnisse und ihr Handling im wesentlichen aus Unix übernommen wurden, erleichtert es den heutigen LINUX-Programmierern, den Zugriff auf DOS-Daten zu realisieren; denn Dateien und Verzeichnisse sind in DOS und LINUX prinzipiell sehr ähnlich.

Gerätedateien vereinfachen den Zugriff des Anwenders auf die Hardware. Sie sind im Prinzip Verweise auf die entsprechenden Treiber im LINUX-Kernel. Sollen etwa Daten auf den Bildschirm ausgegeben werden, so kann der Prozeß, der diese Aufgabe zu erledigen hat, die Daten einfach in die zugehörige Gerätedatei, dies könnte z. B. `tty01` sein, kopieren. Dadurch wird der zuständige Gerätetreiber im Kernel zur Ausgabe auf dem Bildschirm veranlaßt.

Named Pipes oder **FIFO**s sind eines der vielen Mittel, die LINUX für die Kommunikation zwischen Prozessen (kurz **IPC** für *Inter Process Communication*) bereitstellt. Vereinfacht gesagt ist ein Prozeß ein laufendes Programm. Solch ein Prozeß kann Daten in eine FIFO schreiben, die dann von einem anderen Prozeß gelesen werden können. Eine Variante dieser Möglichkeit des Datenaustausches zwischen Programmen kennen wir schon. Eine Kommandozeile wie

`ls -l | more`

dürfte Ihnen in ihrer Bedeutung klar sein.[6] Der Befehl `ls` leitet seine Daten statt auf den Bildschirm an den

[6]Falls nicht, arbeiten Sie Kapitel 5 noch einmal durch.

Pager `more` weiter, der sie so bearbeitet, daß er sie bildschirmseitenweise anzeigt. Bei Linux handelt es sich um temporäre Objekte, die von Prozessen benutzt werden können, die durch ein und dieselbe Befehlszeile gestartet wurden. Named Pipes dagegen werden als besondere Dateien angelegt und können von Prozessen benutzt werden, die unabhängig voneinander sind.

Sockets dienen, wenn man so will, ebenfalls der IPC. Dabei geht es aber nicht um Kommunikation zwischen Prozessen in einem LINUX/Unix-Rechner. Sockets ermöglichen Kommunikation über ein Netz hinweg. Dabei kann durchaus einer der beteiligten Kommunikationsteilnehmer ein Nicht-Unix -Programm sein, d.h. Sockets ermöglichen durchaus Kommunikation zwischen einem Prozeß, den ein Benutzer auf einem LINUX-Rechner startet einerseits, und einer DOS-Anwendung auf einem weit entfernten PC andererseits. Da nun DOS als Einplatzsystem keine Prozesse kennt, lassen sich Sockets nicht eindeutig der IPC zuordnen. Im wesentlichen steckt hinter den Sockets jedoch ein dem Pipelining verwandtes Prinzip des verbindungsorientierten Datenaustausches. Sie ermöglichen also die Weitergabe von Daten, in diesem Falle sogar via Netzwerk.

Links sind keine eigentlichen Dateien, sondern lediglich Verweise auf solche. Im Zusammenhang mit den Inodes werden wir Links genauer untersuchen können.

10.4.1 mkfifo

Erzeugen kann man FIFOs mit dem Befehl `mkfifo` (*make fifo*).

`mkfifo [OPTIONEN] NAME`

Die Befehlszeile
`mkfifo testpipe`
würde z. B. eine Named Pipe mit Namen `testpipe` erzeugen. Genutzt werden könnte sie dann wie folgt:
`ls /dev > testpipe & less < testpipe`
`ls /dev` zeigt die Namen aller Gerätedateien (genauer: aller Dateien des Verzeichnisses `/dev`). In diesem Fall werden die Daten in die Named Pipe umgeleitet statt auf dem Bildschirm ausgegeben. Der Befehl `less` liest die Daten aus `testpipe` und gibt sie bildschirmseitenweise aus. Das `&` zwischen den Befehlen sorgt dafür, daß beide Teilbefehle nacheinander als Prozesse gestartet werden, allerdings ohne daß auf ihr Ende gewartet werden müßte. Insgesamt werden in diesem Beispiel also die Namen aller Gerätedateien bildschirmseitenweise angezeigt. Dieses Beispiel stellte die Named Pipes nur prinzipiell vor. Als Benutzer wird man in der gegebenen Situation eine gewöhnliche Pipe vorziehen:
`ls /dev | less`

10.4.2 Dateinamen

Das Dateisystem von MS-DOS beschränkt die Größe des Namens einer Datei auf acht Zeichen plus einem mit einem Punkt abgetrennten Dateikürzel mit drei Zeichen. Im Gegensatz dazu erlaubte schon das erste UNIX-Dateisystem 14 Zeichen. Das LINUX-Dateisystem ext2 (*extended filesystem, version 2*) erlaubt bis zu 255 Zeichen. Dabei ist der Punkt nur eines von vielen Zeichen und kann beliebig oft angewendet werden.

Im Gegensatz zu MS-DOS unterscheidet LINUX bei Dateinamen zwischen Groß- und Kleinschreibung. `TAPICO`, `Tapico` und `tapico` sind drei verschiedene Dateinamen. Aus praktischen Gründen werden die Dateinamen meistens klein geschrieben. Bei Pfadangaben werden die einzelnen Verzeichnisse durch einen Schrägstrich `/` (*slash*) getrennt. Daher ist dieses Zeichen für Dateinamen verboten. Auch das Minuszeichen kann am Anfang einer Datei Probleme bereiten. So stellt sich bei
`ls -fault`
die Frage, ob nun die Datei `-fault` oder die Schalter `f`, `a`, `u`, `l` und `t` gemeint sind.

Probleme bereiten auch die nichtdruckbaren (ASCII-)Zeichen 0 bis 32 sowie
`< > | * ? [ ] { } ( ) $ ‘ ´ " ! ; \ # ^`
Diese Zeichen sind nicht verboten, können aber große Probleme bereiten und sollten deshalb auch vermieden werden.

10.4.3 Die Datei

Was gehört eigentlich zu einer Datei?

1. Der Inhalt der Datei
2. Ein Link
   (a) Der Name der Datei
   (b) Die eindeutige Inode-Nummer
3. Der Inhalt der Inode (die eigentliche Verwaltungsinformation)
   (a) Dateityp (gewöhnliche Datei, Verzeichnis, Gerätedatei, ...)
   (b) Rechtetabelle
   (c) Anzahl der Links (bzw. Dateinamen)
   (d) Dateibesitzer (uid)
   (e) Besitzergruppe (gid)
   (f) Größe in Byte (bei Geräteadressen statt dessen Major und Minor Device number)
   (g) Datum der letzten Änderung
   (h) Datum des letzten Zugriffs
   (i) Datum der letzten Änderung dieser Verwaltungsinformationen
   (j) Attribute (nur für das Dateisystem ext2)
   (k) physikalischer Ort der Speicherung (Blöcke in denen die Datei abgelegt ist)

Die Inode speichert alle Verwaltungsinformationen über die Datei. Da die Benutzer in aller Regel besser mit Namen als mit Nummern arbeiten können, bietet Linux folgende einfache Schnittstelle zwischen Benutzer und Dateisystem an. Der Benutzer verwendet in den Befehlen einen Dateinamen. Die benutzte Shell, z. B. die `bash`, ermittelt über den Verzeichniseintrag, den Link, die zugehörige Inode. Jeder Dateiname ist mit genau einer Inode verbunden. Dagegen kann eine Inode durch mehrere Dateiname referenziert werden. Deshalb kann eine Datei nicht nur einen Dateinamen besitzen. Wird die Inode beschädigt, dann kann auf die Datei nicht mehr zugegriffen werden. Um die Inodes der Dateien zu bestimmen, wird der Schalter `-i` des Befehls `ls` eingesetzt.

Beispiel

```
tapico@defiant:/ > ls -aFi
      2 ./           1 d/           11 lost+found/    10241 tmp/
      2 ../      38914 dev/     124929 mnt/             165 trellwan
 305412 a/        2049 etc/     126977 opt/           83969 usr/
  59393 bin/    118785 floppy/       1 proc/          12289 var/
  51201 boot/   120833 home/     69633 root/
      1 c/      117000 kurs/     55297 sbin/
 112641 cdrom/   65537 lib/     211184 seminare/
```

Besitzen zwei Dateinamen die gleiche Inode, so handelt es sich hierbei um ein und dieselbe Datei. Auffällig ist die Inode 1 bei `/proc`, `/c` und `/d`. Bei `proc` handelt es sich um ein virtuelles Dateisystem zur Prozessüberwachung. `/c` und `/d` sind eingebundene FAT16-Dateisysteme von Windows. Ihnen wird daher auch keine "richtige" Inode zugewiesen.

10.4.4 stat

Der Befehl `stat` zeigt die Informationen der Inode an.

```
stat DATEINAME
```

Allerdings ist er nicht auf jedem System zu finden. Bei SuSE muß das Paket explizit installiert werden.

10.5 Links

Im Prinzip ist ein Link nur ein weiterer Name für die Datei. Bei den Links wird zwischen dem harten Link und dem symbolischen (weichen) Link unterschieden.

Ein Link kann dazu benutzt werden einem anderen Benutzer Zugriff auf eine Datei zu gewähren. Als Erstes bekommt der andere Benutzer Rechte auf die Datei und evt. auf das Verzeichnis. Danach kann er einen Link auf die Datei in seinem Heimatverzeichnis anlegen. Dadurch erhält er einen vereinfachten Zugriff auf den Inhalt der Datei.

Durch die Benutzung von Links brauchen oft keine Kopien von Dateien erstellt werden. Dies spart Speicherplatz und erleichtert die Aufgabe des Systemadministrators alle Dateien auf dem neuesten Stand zu halten.

Daneben können große Verzeichnisbäume übersichtlicher gestaltet werden. Z. B. landen alle Auswertungsergebnisse in einem Verzeichnis `data`. In den Verzeichnissen `helios` und `ulysses` können dann Links auf die entsprechenden Datendateien im Verzeichnis `data` gemacht werden.

10.5.1 Harte Links

Immer wenn Sie eine Datei erzeugen wird auch ein Link zu dieser Datei erzeugt. Er ermöglicht den Zugriff auf die Datei, weil er direkt auf die Inode der Datei zeigt. Der Befehl `rm` (4.5.7) löscht übrigens nicht direkt die Datei, sondern den Link, der auf die Inode der Datei zeigt.

Einen Link können Sie nur auf eine existierende Datei anlegen. Er ist ein weiterer Name für die gleiche Datei. Dies wird deutlich wenn Sie sich die Links mit `ls -i` anzeigen lassen. Beide Links zeigen auf die gleiche Inode. Sie unterscheiden sich nur im Namen und/oder ihrem Ort im Verzeichnisbaum.

Beispiel
Auf die existierende Datei `clarkkent` wird ein harter Link namens `superman` angelegt. Wird nun `superman` durch z. B. den Editor `joe` editiert, so finden sich die Änderungen auch in `clarkkent` wieder.

```
tapico@defiant:~/links > ls -il
insgesamt 2
 111099 -rw-r--r--   2 tapico   users          46 Jan 14 13:41 clarkkent
 111099 -rw-r--r--   2 tapico   users          46 Jan 14 13:41 superman
```

In der dritten Spalte sehen Sie die Anzahl der harten Links, die auf die Datei zeigen. Lösche Sie nun einen der beiden Links, so bleibt die Datei trotzdem bestehen. Erst wenn alle Links auf die Datei gelöscht sind, die Zahl also auf Null steht, wird die Datei auf wirklich gelöscht.

Es gibt aber auch Einschränkungen für harte Links. So müssen sich alle Links auf dem gleichen Dateisystem befinden. Außerdem ist es einem normalen Benutzer nicht möglich einen Hardlink auf ein Verzeichnis einzurichten. Dies ist nur **root** vorbehalten.

10.5.2 Symbolische Links

Im Gegensatz zu dem harten Link zeigt der symbolische Link nur indirekt auf eine Datei, da er auf einen Namen (harten Link) der Datei zeigt. Deshalb ist er auch ein neuer Eintrag mit eigener Inode im Verzeichnisbaum.

Symbolische Links werden dann eingerichtet, wenn die Grenzen der harten Links umgangen werden sollen. Symbolische Links können eingerichtet werden für

- Verzeichnisse durch einen normalen Benutzer.
- Dateien, die nicht existieren.
- Dateien, die sich auf einem anderen Dateisystem befinden.

Wie im Beispiel unten zu sehen, ändert sich das Aussehen des symbolischen Links bei Verwendung des Befehls `ls -l`. Der Linkname wird durch einen Pfeil und den Namen der Zieldatei ergänzt.

```
tapico@defiant:~/links > ls -il
insgesamt 2
 111099 -rw-r--r--   2 tapico   users          46 Jan 14 13:41 clarkkent
 111101 lrwxrwxrwx   1 tapico   users           5 Jan 14 13:51 kurs -> /kurs
 111100 lrwxrwxrwx   1 tapico   users           9 Jan 14 13:50 superheld -> clarkkent
 111099 -rw-r--r--   2 tapico   users          46 Jan 14 13:41 superman
```

Die Funktion der Befehle `cd` und `pwd` kann sich bei gelinkten Verzeichnissen je nach Shell unterscheiden. In der `bash` zeigt, wenn wir dem Link `kurs` gefolgt sind, z. B. der Befehl `pwd` das Arbeitsverzeichnis über die verlinkte Struktur an. Der Befehl `cd ..` führt auch in das Verzeichnis mit dem Link zurück.

```
tapico@defiant:~/links > cd kurs
tapico@defiant:~/links/kurs > pwd
/home/tapico/links/kurs
tapico@defiant:~/links/kurs > cd ..
tapico@defiant:~/links > pwd
/home/tapico/links
```

In anderen Shells führt der Link an den Ort der normalen Struktur. Das aktuelle Verzeichnis ist dann nicht mehr durch den symbolischen Link gegeben, sondern durch den harten Link des Verzeichnisses.

10.5.3 ln

Das Kommando `ln` legt einen harten oder einen symbolischen Link an.

`ln [OPTIONEN] DATEINAME NEUERLINK`

Ist `NEUERLINK` ein Verzeichnis und die Optionen `-d` und `-F` sind nicht gegeben, so wird in dem Verzeichnis ein Link mit dem Namen `DATEINAME` angelegt.

Optionen	
`-d`	Legt einen Link auf ein Verzeichnis an (Nur Superuser)
`-F`	Legt einen Link auf ein Verzeichnis an (Nur Superuser)
`-s`	Legt einen symbolischen Link an

Einen harten oder symbolischen Link können Sie durch den Befehl `rm` entfernen.

10.6 Festplattennutzung

Die Einrichtung, Formatierung und Einbindung einer Partition in das System hat einen gravierenden Nachteil. Die Benutzer können nun dort Dateien speichern und füllen dort in kürzester Zeit den vorhandenen Speicherplatz[7]. Damit Linux vernünftig arbeiten kann, sollte zwischen 5 und 30 Prozent der Plattenkapazität frei sein. Es ist also zwingend notwendig den zur Verfügung stehenden Plattenspeicher zu kontrollieren.

10.6.1 du

Das Kommando `du` (*Disk Usage*) zeigt den Platz an, den eine Datei oder ein Verzeichnis mit seinen Dateien und Unterverzeichnissen einnimmt.

`du [OPTIONEN] [DATEINAME]`

[7] Hier sei auf eins der ewigen EDV-Gesetze hingewiesen: Jede Festplatte jeder beliebigen Größe wird innerhalb kurzer Zeit zu 90% belegt sein.

Wird du ohne Dateinamen aufgerufen, dann zeigt es die Daten des aktuellen Arbeitsverzeichnisses an.

Optionen	
-a	Zeige die Nutzung für einzelne Dateien
-b	Anzeige des Speicherplatzes in Bytes
-c	Zeigt die Summe an
-h	Zeigt durch einen Buchstaben die verwendete Einheit an
-k	Anzeige des Speicherplatzes in Kilobytes (normal)
-m	Anzeige des Speicherplatzes in Megabytes
-l	Zählt auch die Links
-s	Gibt nur die totale Summe aus

Ohne den Schalter -a zeigt du nur die Verzeichnisse.

Beispiel

```
tapico@defiant:~ > du -c test
1259    test/mytest/textfilter
1263    test/mytest
14761   test/log
17453   test
17453   insgesamt
```

10.6.2 df

Der Befehl df zeigt die Nutzung der Partition an.

```
df [OPTIONEN] [DATEINAME]
```

Wird ein Dateiname angegeben, so zeigt df die Daten der Partition an, auf der die Datei liegt. Ansonsten zeigt er eine Liste aller Partitionen an.

Optionen	
-a	Liefert Informationen über alle Dateisysteme (normal) (*all*)
-h	Zeigt durch einen Buchstaben die verwendete Einheit an(*human readable*)
-i	Zeigt die Nutzung der Inodes an(*inode*)
-k	Anzeige des Speicherplatzes in Kilobytes (normal)(*kilobyte*)
-m	Anzeige des Speicherplatzes in Megabytes(*megabyte*)
-t DSTYP	Zeigt nur Dateisysteme vom Typ DSTYP(*typ*)
-T	Zeigt den Dateisystemtyp für jeden Eintrag(*typ*)
-x DSTYP	Zeigt keine Dateisysteme vom Typ DSTYP(*exclude typ*)
--sync	Startet erst das sync-Kommando

Als Information liefert df:

- Die Größe des Geräts
- Die Anzahl der freien Blöcke
- Die Anzahl der belegten Blöcke
- Den Anteil der freien Blöcke am Speicher in Prozent
- Den Namen des Geräts

Beispiel

```
tapico@defiant:~ > df
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/hda3              1430014   1226061    130054  90% /
/dev/hda1              2096160   1367744    728416  65% /c
/dev/hda5              1052064    363392    688672  35% /d
```

10.6.3 Aufräumen des Systems

Was also tun, wenn das Dateisystem zu voll geworden ist?

- Finden Sie große Dateien, die nicht mehr benötigt werden. Log-, Mail- und News-Dateien können sehr groß werden. Auch die durch Abstürze erstellen Speicherabbildungen (*core dumps*) können sehr groß werden.
- Räumen Sie die temporären Verzeichnisse auf. Das sollte sowieso zu den normalen Wartungsaufgaben gehören.
- Selten genutzte Dateien sollten komprimiert werden und möglichst auf andere Geräte ausgelagert werden.
- Verzeichnisse sollten nicht so viele Dateien enthalten. Kleinere Verzeichnisse sind effektiver.
- Die Verzeichnisse an sich können nur größer werden. Je mehr Dateien vorhanden sind, desto größer wird der Verwaltungsplatz. Werden die Dateien gelöscht, bleibt die Verwaltungsdatei trotzdem so groß wie vorher. Das Verzeichnis muß gelöscht und anschließend wieder erstellt werden, um den Platz zu reduzieren.
- Ist nur eine Partition zu voll, können Dateien auf andere Partitionen ausgelagert werden.
- Eine zweite Möglichkeit ist es die Partition zu vergrößern. Dabei muß von den Daten ein Backup erstellt werden. Dann wird die Partition gelöscht und in dem neuen freien Platz (wahrscheinlich eine neue Platte) neu erstellt. Dann wird das Backup wieder zurückgespielt.

Notizen:

Notizen:

XII Arbeitsblatt Dateisystem 1 12.1

1 Welche Dateinamen sind richtig, problematisch oder falsch?

Dateiname	R	P	F	Dateiname	R	P	F
`abc.txt`				`Dies ist Falsch`			
`Textdatei`				`ls -l`			
`-usr`				`*.*`			
`c:\dos`				`this is right`			
`pillepallename`				`\">$<[!]`			
`..`				`...`			
`../..`				`calprobedatei1`			
`katalog_1.2.1`				`abc/def/ghi`			
`-123`				`123`			
`mc.man.gz`				`194.195.155.60`			
`..home`				`../home`			

2 Loggen Sie sich als *walter* auf der Konsole 1 ein.

3 Kopieren Sie den Inhalt der Verzeichnisse `yast2`, `mc` und `cron` aus dem Verzeichnis `/usr/share/doc/packages` in Ihr Heimatverzeichnis.

4 Lassen Sie sich den Speicherbedarf der Dateien in Ihrem Heimatverzeichnis anzeigen.

5 Lassen Sie sich eine Übersicht über den freien Platz auf den Partitionen anzeigen.

6 Welchen Platz benötigen die Verzeichnisse `/etc`, `/home` und `/usr/bin`?

7 Loggen Sie sich als *root* auf der Konsole 2 ein.

8 Wiederholen Sie Aufgabe 6 unter *root*. Gibt es Unterschiede im Ergebnis?

9 Legen Sie im freien Speicherbereich bzw. im freien Bereich der erweiterten Partition eine Partition von 500 MB an und formatieren Sie sie mit dem Dateisystem ext2.

10 Mounten Sie die Platte als *root* auf dem neuangelegten Mountpunkt `/usr/extra`.

11 Wie groß ist die Platte laut `fdisk`?

12 Lassen Sie sich eine Übersicht über den freien Platz auf den Partitionen anzeigen und vergleichen Sie ihr Ergebnis mit dem Ergebnis der vorherigen Aufgabe.

13 Lassen Sie sich die Liste der Swap-Bereiche anzeigen.

14 Legen Sie im freien Speicherbereich bzw. im freien Bereich der erweiterten Partition eine Swap-Partition von 256 MB an und aktivieren Sie sie.

15 Lassen Sie sich die Liste der Swap-Bereiche erneut anzeigen.

16 Wieviel Swap-Speicher steht ihnen laut `free` zur Verfügung?

17 Starten Sie den Rechner neu.

18 Führen Sie die Aufgaben 12 und 13 erneut durch und vergleichen Sie die jetzigen Ergebnisse mit den vorherigen.

XII	Arbeitsblatt Dateisystem 1	12.2

19 Richten Sie das System so ein, daß die neue Partition und der neue Swap-Bereich permanent zugänglich sind.

20 Testen Sie die neuen Einstellungen, indem Sie den Rechner neu starten und die Aufgaben 12 und 13 erneut durchführen.

21 Deaktivieren Sie wieder den neuen Swap-Bereich permanent.

22 Ein Mitarbeiter steht vor dem Problem, daß er ein-, zweimal die Woche mehr Auslagerungsspeicher für komplizierte Rechenoperationen benötigt. Platz genug ist auf seiner Festplatte. Leider ist aber kein freier Bereich für eine Partition mehr vorhanden. Lösen Sie das Problem so, daß der User eigenständig den neuen Swap-Bereich aktivieren und deaktivieren kann.

Notizen:

XIII	Arbeitsblatt Arbeiten mit Disketten	13.1

Sie benötigen für die Aufgaben zwei 3,5"-Disketten.

1 Loggen Sie sich als root an Terminal 1 ein.

2 Legen Sie den User *ardan* mit Home-Verzeichnis an. Geben Sie ihm das Paßwort *victor*.

3 Lassen Sie sich alle möglichen Gerätetypen für das Diskettenlaufwerk A: anzeigen.

4 Legen Sie die erste Diskette ins Diskettenlaufwerk ein. Führen Sie eine Low-Level-Formatierung (1440 KB) durch.

5 Erstellen Sie ein ext2-Dateisystem auf der Diskette.

6 Lassen Sie sich den Inhalt der Datei `/etc/mtab` anzeigen. Welche Laufwerke sind gemountet?

7 Mounten Sie die Diskette auf das Verzeichnis `/floppy`.

8 Lassen Sie sich erneut den Inhalt der Datei `/etc/mtab` anzeigen. Welche Laufwerke sind gemountet?

9 Loggen Sie sich an Terminal 2 als `ardan` ein. Versuchen Sie das Diskettenlaufwerk auf `/floppy` zu mounten.

10 Wechseln Sie zu Terminal 1. Unmounten Sie das Diskettenlaufwerk. Wechseln Sie zu Terminal 2 und versuchen Sie erneut das Diskettenlaufwerk zu mounten.

11 Lassen Sie sich die Auslastung aller Laufwerke anzeigen.

12 Unmounten Sie das Diskettenlaufwerk.

13 Wechseln Sie zu Terminal 1 und legen Sie das Verzeichnis `/a` an. Ändern Sie die Rechte des Verzeichnisses so, daß alle volle Rechte besitzen.

14 Mounten Sie die Diskette auf das Verzeichnis `/a`.

15 Unmounten Sie die Diskette wieder. Wechseln Sie zu Terminal 2. Versuchen Sie nun als `ardan` das Diskettenlaufwerk zu mounten.

16 Wechseln Sie wieder zu Terminal 1 und fügen Sie in die Datei `/etc/fstab` einen Eintrag hinzu, so daß jeder das Diskettenlaufwerk auf `/a` mounten kann. Testen Sie dies.

17 Formatieren Sie nun die Diskette 2 mit 1440 KB und erstellen ein DOS-Dateisystem darauf.

18 Ändern Sie die Datei `/etc/fstab` so, daß auf das Verzeichnis `/a` nur vfat-Dateisysteme gemountet werden können.

19 Mounten Sie Diskette 2 auf `/a`.

20 Versuchen Sie nun Diskette 1 auf `/a` zu mounten. Was passiert?

21 Führen Sie erneut eine Low-Level-Formatierung für Diskette 1 ein durch. Wählen Sie dabei das größtmögliche Format. Erstellen Sie ein ext2-Dateisystem auf der Diskette. Wieviel Platz steht Ihnen nun auf der Diskette zur Verfügung.

22 Erstellen Sie ein minix-Dateisystem auf der Diskette. Wieviel Platz steht Ihnen nun zur Verfügung?

23 Erstellen Sie ein DOS-Dateisystem auf der Diskette. Wieviel Platz steht Ihnen nun auf der Diskette zur Verfügung.

24 Kann ein Windows-Rechner dieses Diskette lesen? Welche Kapazität kann ein Windows-Rechner maximal verkraften?

Notizen:

Kapitel 11

Dateisysteme und Disk Quotas

11.1 Aufbau des ext2-Dateisystems

Das ext2-Dateisystem besteht aus dem Bootblock und den Blockgruppen. Die Blockgruppen sind wiederum aufgeteilt in einen Superblock, einer Liste der Blockgruppenbeschreibungen, Bitmaps für Blöcke und Inodes, Inode-Tabelle und den Datenblöcken.

11.1.1 Bootblock

Der Bootblock liegt im ersten Block (Block 0) des Dateisystems. Er enthält ein Programm zum Starten und Initialisieren des gesamten Systems. Meistens enthält nur das erste Dateisystem einen Bootblock. Beim ext2-Dateisystem ist der Bootblock 1024 Zeichen groß.

11.1.2 Blockgruppe

Jede Blockgruppe, wie in der Abbildung 11.1 zu sehen, besteht aus sechs Komponenten:

- Superblock
- Liste der Blockgruppenbeschreibungen
- Bitmap für Blöcke
- Bitmap für Inodes
- Inode-Tabelle
- Datenblöcke

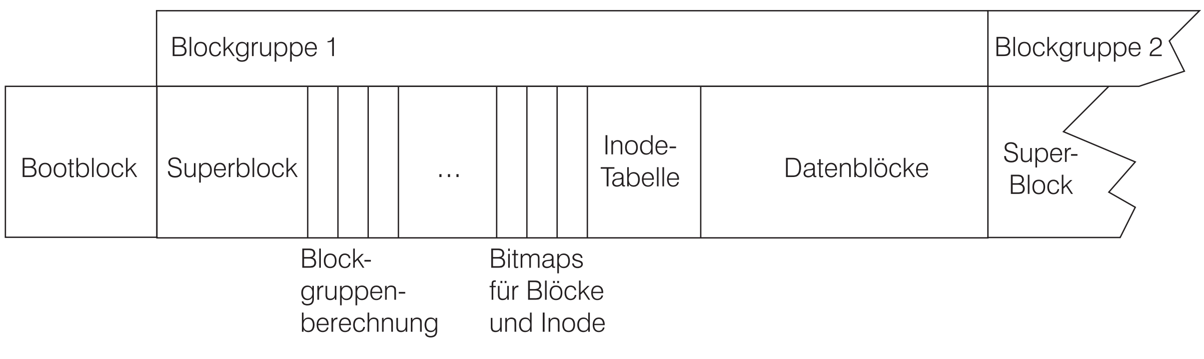

Abbildung 11.1: Aufbau des ext2-Dateisystems

Die Standardgröße für eine Blockgruppe liegt bei 8192 Blöcken und 2048 Inodes, von denen acht reserviert sind.

11.1.3 Superblock

Die Informationen über das gesamte Dateisystem liegen im Superblock. Wenn dieser Superblock beschädigt wird, kann das Dateisytem nicht mehr ins Dateisystem eingebunden (gemountet) werden. Deswegen werden in regelmäßigen Intervallen Backup-Kopien dieses Blocks im Dateisystem angelegt. Im Normalfall erfolgt das alle 8192 Blöcke. Das heißt die Daten befinden sich in Block 8193, 16385, 24577 etc. Die Angabe
`Blocks per group:          8192`
des Befehls `dumpe2fs` gibt den für das Gerät aktuellen Wert an.

Die Strukturdefinition des Superblocks finden Sie in der Datei `/usr/include/linux/ext2_fs.h` im Abschnitt `Structure of the super block`. Die wichtigsten Informationen sind:

- Gesamtgröße des Dateisystems in Blöcken bzw. Inodes
- Anzahl der freien Blöcke
- Anzahl der Blöcke für die Inodes
- Anzahl der freien Inodes
- Adresse des ersten Datenblocks
- Größe eines Datenblocks
- Größe eines Teildatenblocks
- Größenbestimmung für eine Blockgruppe
- Zeitpunkt des Einbinden des Dateisystems
- Zeit der letzten Änderung
- Anzahl der Einbindungen (*mount counts*)
- Maximale Anzahl der Einbindungen bevor ein Dateisystemcheck durchgeführt wird
- Versionsnummer des Dateisystems
- Magische Nummer: für ext2 lautet sie `0xEF53`.
- Nummer des einrichtenden Betriebssystems
- Status des Dateisystems: sauber oder mit Fehlern
- Verhalten bei Fehlererkennung
- Zeitpunkt der letzten Prüfung
- Maximaler Zeitraum zwischen zwei Prüfungen

11.1.4 dumpe2fs

Der Befehl `dumpe2fs` gibt den Superblock und die Blockgruppeninformationen eines Gerätes aus.

```
dumpe2fs [OPTIONEN] GERÄT
```

Der Befehl steht nur dem Superuser zur Verfügung. Als Beispiel hier die Ausgabe eine Befehls ohne die Übersicht über die Blockgruppen.

```
root@defiant:/ > dumpe2fs /dev/hda3
Filesystem volume name:   <none>
Last mounted on:          /mnt
Filesystem UUID:          0c7c1f82-510a-11d3-8c57-e447ecb45d66
Filesystem magic number:  0xEF53
Filesystem revision #:    0 (original)
Filesystem features:     (none)
Filesystem state:         not clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              370688
Block count:              1477980
Reserved block count:     73899
Free blocks:              203986
Free inodes:              269515
First block:              1
Block size:               1024
Fragment size:            1024
Blocks per group:         8192
Fragments per group:      8192
Inodes per group:         2048
Inode blocks per group:   256
Last mount time:          Mon Jan  8 20:45:40 2001
Last write time:          Mon Jan  8 21:19:10 2001
Mount count:              20
Maximum mount count:      20
Last checked:             Mon Dec 18 20:44:00 2000
Check interval:           15552000 (6 months)
Next check after:         Sat Jun 16 21:44:00 2001
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
```

11.1.5 tune2fs

Die Einstellungen für das Dateisystem können über den Befehl `tune2fs` eingestellt werden.

`tune2fs [OPTIONEN] GERÄT`

Optionen	
`-c MOUNTCOUNTS`	Stellt die Anzahl der maximalen Mountcounts zwischen zwei Dateisystemchecks ein
`-C MOUNTCOUNTS`	Trägt die Anzahl der erfolgten Mountcounts ein

Die weiteren Schalter entnehmen Sie bitte den Man-Pages.

Beispiele

```
root@defiant:/> tune2fs -c 200 /dev/hda5
```

stellt die Anzahl der maximalen Mountvorgänge zwischen zwei Dateisystemchecks für das Gerät `/dev/hda5` auf 200 ein.

11.1.6 Liste der Blockgruppenbeschreibungen

Hier werden die wichtigsten Kenndaten der Blockgruppe festgehalten. Dies sind

- Blocknummer für Block-Bitmap
- Blocknummer für Inode-Bitmap

- Blocknummer für Inode-Tabelle
- Anzahl der freien Blöcke
- Anzahl der freien Inodes
- Anzahl der Verzeichnisse

Damit die Verzeichnisse gleichmäßig über die Blockgruppen verteilt werden, werden neue Verzeichnisse in dem Block angelegt, der am wenigsten Verzeichnisse enthält.

11.1.7 Block- und Inode-Bitmap

Die Bitmaps für die Verwaltung der Blöcke und Inodes sind auf einen logischen Block beschränkt. Für jeden Block wird ein Bit benötigt. Daher ist eine Blockgruppe auf $1024 \cdot 8 = 8192$ Blöcke beschränkt.

11.1.8 Inode-Liste

Die Inode-Liste besteht aus vielen aneinandergereihten Inodes, die alle exakt die gleiche Größe haben. Im Standardfall sind es 2048 Stück, die innerhalb einer Blockgruppe gespeichert werden. Die Inode ist der Dreh- und Angelpunkt des Dateizugriffs.

- Dateityp (gewöhnliche Datei, Verzeichnis, Gerätedatei, ...)
- Rechtetabelle
- Anzahl der Links (bzw. Dateinamen)
- Dateibesitzer (uid)
- Besitzergruppe (gid)
- Größe in Byte (bei Geräteadressen statt dessen Major und Minor Device number)
- Datum der letzten Änderung
- Datum des letzten Zugriffs
- Datum der letzten Änderung dieser Verwaltungsinformationen
- Attribute (nur für das Dateisystem ext2)
- physikalischer Ort der Speicherung (Blöcke in denen die Datei abgelegt ist)

Adressierung des Speicherblocks

In der Inode befinden sich auch die Informationen über die für die Datei verwendeten Datenblöcke. Diese Tabelle besitzt 15 Einträge.

Die ersten zwölf Einträge verweisen direkt auf jeweils einen Datenblock. Im Normalfall ist dieser logische Block 1024 Bytes groß. Es können also direkt 12 KB adressiert werden. Der Zugriff ist aufgrund der Mechanik des Caches sehr schnell.

Der dreizehnte Eintrag weißt auch auf einen Datenblock. Dieser Datenblock enthält aber wiederum bis zu 256 Adressen von Datenblöcken. Es sind also nun 268 KByte adressierbar.

Der vierzehnte Eintrag weißt auf einen Datenblock der 256 Einträge vom Typ des dreizehnten Eintrags enthält. Er adressiert hier also $256 \cdot 256$ Blöcke mit 64 MByte Daten.

Der fünfzehnte Eintrag weißt auf einen Datenblock mit 256 Einträgen vom Typ des vierzehnten Eintrags. Hiermit werden also $256 \cdot 256 \cdot 256$ Blöcke mit ca. 16,6 GByte adressiert.

11.2 Pflege des Dateisystems

Nachdem Sie nun mit viel Mühe die Partitionen und Ihr System eingerichtet haben, sollten Sie Ihr Dateisystem auch pflegen, damit es gesund bleibt. Wie beim Menschen ist Vorsorge besser als Nachsorge.

11.2.1 fsck

Das wichtigste Programm für die Wartung des Dateisystems ist `fsck`. Wie auch `mkfs` (10.3.1) ist `fsck` ein Frontend zu dem Systemcheck-Programm des jeweiligen Dateisystemtyps. Es wird zur Überprüfung des Dateisystems und zur Behebung von Unstimmigkeiten verwendet.

In den meisten Fällen basiert ein solcher Dateisystemfehler auf einem Systemabsturz. Der Kernel ist dann nicht mehr in der Lage die im Cache zwischengespeicherten Daten auf die Platte zu schreiben.

`fsck [OPTIONEN] [-t DSTYP] [GERÄT]`

Optionen	
`-a`	Das Programm läuft ohne Rückfragen ab (Vorsicht!)
`-c`	Untersucht auf defekte Blöcke

Beispiel
Um das ext2-Dateisystem der zweiten Partition der ersten IDE-Platte zu testen geben Sie den folgenden Befehl ein.
`fsck -t ext2 /dev/hda2`
Dies führt dazu, daß das Programm `e2fsck` aufgerufen wird.

Ist dieses Dateisystem das Wurzel-Dateisystem, dann fragt `e2fsck` nach, ob der Check auch durchgeführt werden soll. Beim Check werden getestet:

- Inodes
- Blöcke
- Dateigröße
- Verzeichnisstruktur
- Links

11.2.2 e2fsck

Das Tool `e2fsck` ist das Dateisystem-Checkprogramm für das ext2-Dateisystem.

`e2fsck [OPTIONEN] GERÄT`

Optionen	
`-b BLOCK`	Die zu benutzende Kopie des Superblocks
`-f`	Führt den Check auch durch, wenn das System sauber erscheint.

Zum Testen können alle Dateisysteme außer dem Wurzel-System unmountet werden. Das Wurzel-System kann in folgenden Situationen gestestet werden.

- Im Modus “Single-User” und “Read-Only”
- Booten mit der Diskette
- Während des normalen Systemstarts

Reparatur des Superblocks
Sollte der Superblock beschädigt sein, so kann das Dateisystem nicht gemountet werden. Für den Dateisystemcheck geben wir deshalb eine Kopie des Superblocks an. Für ein ext2-Dateisystem lautet der Befehl dann:
`e2fsck -f -b 16385 /dev/hda3`
Die Option `-f` muß gesetzt werden, da das Programm sonst die Kopie des Superblocks für den echten hält und, da dieser ja in Ordnung ist, den Test beendet.

11.2.3 badblocks

Mit dem Tool `badblocks` können Sie ein Gerät, in den meisten Fällen eine Festplatte oder Diskette, auf defekte Datenblocks untersuchen.

```
badblocks [OPTIONEN] GERÄT [LETZTER-BLOCK] [ERSTER-BLOCK]
```

Mit der Angabe von ERSTER-BLOCK und LETZTER-BLOCK können sie auch einen Bereich auf dem Gerät bestimmen, welcher untersucht werden soll. Falls nichts angegeben wird, wird vom ersten bis letzten Block alles getestet.

Das Programm `badblock` sollte möglichst nur auf nicht gemounteten Geräten gestartet werden.

Optionen	
`-b GROESSE`	Größe der Datenblöcke in Bytes
`-c ZAHL`	Anzahl der Blocks, die auf einmal getestet werden
`-f`	Erlaubt auch die Durchführung bei gemounteten Geräten. Vorsicht!
`-i DATEI`	Liest Liste mit bekannten defekten Blocks ein
`-o DATEI`	Schreibt gefundene defekte Block in die angegebene Datei
`-p VERSUCHE`	Anzahl der Wiederholungen, wenn kein defekter Block gefunden wurde. Defaultwert ist 0
`-n`	Nicht zerstörender Lese-und-Schreibtest, Defaulteinstellung
`-s`	Zeigt Fortschritt in Form der abgearbeiteten Blocks
`-v`	"Blubbermodus"
`-w`	Schreibmodus-Test: dieser Test löscht vorhandene Daten

Beispiel

Um eine Diskette zweimal auf defekte Sektoren zu untersuchen wird folgender Befehl eingegeben.

```
enterprise:~ # badblocks -p 2 -s /dev/fd0
Checking for bad blocks (read-only test): done
Checking for bad blocks (read-only test): done
```

11.3 Auf der Suche nach der verlorenen Datei

Wie Sie schon in Abschnitt 4.4 und Abbildung 4.1 gesehen haben, kann in diesem weit verzweigtem Verzeichnisbaum eine Datei leicht mal verloren gehen. Um diese verlorene Datei wieder zu finden, stehen Ihnen einige Tools zur Verfügung. Die Tools `whereis` (6.3.1) und `which` (6.3.2) haben Sie schon in dem Kapitel über die Hilfe kennengelernt. Neben diesen Befehlen gibt es noch `find` und `locate`.

11.3.1 find

Dieses Tool wird dazu benutzt das angegebene Verzeichnis und seine Unterverzeichnisse nach Dateien zu durchsuchen, die den mitgegebenen Kriterien entsprechen.

```
find [PFAD] [SUCHKRITERIEN]
```

Kriterien	Aktion
`-atime ±N`	Findet Dateien auf der Basis des letzten Zugriffs
`-ctime ±N`	Findet Dateien auf der Basis der letzten Änderung des Verzeichniseintrags
`-exec BEFEHL`	Ausführen eines `BEFEHL` für jede gefundene Datei
`-group GRUPPE`	Findet Dateien die zu einer bestimmten `GRUPPE` (GID) gehören
`-inum N`	Findet Dateien mit der Inode `N`
`-links N`	Findet Dateien mit `N` Links
`-mount`	Findet Dateien nur im gleichen Dateisystem
`-mtime ±N`	Findet Dateien auf der Basis der letzten Änderung
`-name MUSTER`	Findet Dateien, auf deren Dateiname das `MUSTER` zutrifft.
`-newer DATEI`	Findet Dateien, die jünger als die angegebene `DATEI` sind.
`-ok BEFEHL`	Ausführen eines `BEFEHL` für jede gefundene Datei mit vorheriger Sicherheitsabfrage
`-perm NNN`	Findet Dateien, auf die Rechtemaske `NNN` zutrifft.
`-print`	Zeigt die gefundenen Dateien mit ihrem Pfad an. (Standard)
`-size N[c]`	Findet Dateien in Abhängigkeit von ihrer Größe (`N` Blocks oder `N` Zeichen).
`-type C`	Findet Dateien vom gegebenen Typ:
	b = Blockgerät, c = Zeichengerät, d = Verzeichnis, l = Link oder f = normale Datei
`-user BENUTZER`	Findet die Dateien des angegebenen Benutzers

Tabelle 11.1: Suchkriterien des `find`-Befehls

Werden keine Kriterien angegeben, so zeigt `find` alle Dateien im Verzeichnis und seinen Unterverzeichnissen an. Wird kein Pfad angegeben, so dient das aktuelle Arbeitsverzeichnis als Startverzeichnis. Um die Anzahl der gefundenen Dateien einzuschränken, können Kriterien für die Eigenschaften der gesuchten Dateien angegeben werden. Tabelle 11.1 liefert einen Überblick über die wichtigsten Kriterien.

Einzelne Suchkriterien sind logisch verknüpfbar, d. h., es können Suchkriterien aus verschiedenen Teilbedingungen zusammengesetzt werden. Numerische Angaben als Bestandteil eines Suchkriteriums lassen sich auf drei Weisen benutzen: `N`, `+N` und `-N`, wobei `N` eine ganze Zahl ist. Das Kriterium `-links` veranlaßt z. B. eine Suche nach Dateien mit einer bestimmten Anzahl von Links und erfordert eine entsprechende numerische Angabe. So bedeuten

`-links 3`	Suche nach Dateien mit 3 Links
`-links +3`	Suche nach Dateien mit mehr als 3 Links
`-links -3`	Suche nach Dateien mit weniger als 3 Links

Beispiele

`find -name "super*"`
Findet alle Dateien im aktuellen Verzeichnis und darunter, die mit “super” beginnen. Dabei sollte das Suchmuster in Anführungszeichen stehen.

`find /usr -type d -print`
sucht nach allen Verzeichnissen, die unterhalb von `/usr` existieren und zeigt ihre Pfadnamen an.

`find / -name tty* -print`
liefert die Pfade aller Dateien des gesamten Dateisystems, die mit “tty” beginnen.

`find / -user 406 -ok rm {} \;`
sucht im gesamten Dateisystem nach Dateien des Benutzers mit der uid 406. Gefundene Dateien werden bei positiver Beantwortung einer Sicherheitsabfrage gelöscht.

`find . -newer .version_time -print`
zeigt alle Dateien des aktuellen Verzeichnisses, die jüngeren Datums sind als die Datei `.version_time`.

`find / -size +10k -user tapico -print`
zeigt alle Dateien die größer als 10 kB sind und dem Benutzer **tapico** gehören.

`find ~ -atime -10 -print`
findet alle Dateien, auf die innerhalb der letzten 10 Tage zugegriffen wurde.

Bei Angabe von mehreren Kriterien wird immer von einer UND-Verknüpfung ausgegangen.
`find / -size +10k -user tapico -print`

zeigt alle Dateien die größer als 10 kB sind und dem Benutzer **tapico** gehören.

Dies kann durch `-a` betont werden.
`find / -size +10k -a -user tapico -print`

Eine ODER-Verknüpfung erreicht man durch `-o`.
`find ~ -name '*.htm' -o -name '*.html' -print`
sucht nach allen Dateien, die auf ".htm" oder ".html" enden.

Die Negation eines Kriteriums wird durch das `!` erreicht.
`find / -name !'*~' -print`
findet alle Dateien, die nicht mit einem Tilde ~ enden.

11.3.2 locate

Während `find` den gesamten Verzeichnisbaum durchsucht, verwendet `locate` für seine Suche die Datenbank `/var/lib/locatedb`.

```
locate [DATEINAME]
```

Der Befehl `locate XYZ` ist wesentlich schneller als `find / -name XYZ`. Allerdings muß auch immer die Datenbank auf dem neuesten Stand sein.

Beispiel
`locate *.ps`
zeigt alle Dateien an, die mit ".ps" enden.

11.3.3 /var/lib/locatedb

Die Datei `/var/lib/locatedb` enthält eine Liste aller Dateien des Dateisystems zu einem bestimmten Zeitpunkt. Sie ist eine große Binärdatei, die Sie mit
`od -c /var/lib/locatedb | less`
betrachten können.

Mit dem Befehl `updatedb` wird die Datenbank auf den neuesten Stand gebracht.

11.3.4 updatedb

Das Kommando `updatedb` bringt die für `locate` wichtige Datenbank `/var/lib/locatedb` auf den neuesten Stand.

```
updatedb [OPTIONEN]
```

Optionen

`--localpaths`	Die lokalen Verzeichnisse werden mit eingebunden
`--netpaths`	Die Netzwerk-Verzeichnisse werden mit eingebunden
`--prunepaths LISTE`	Liste der ausgeschlossenen Verzeichnisse
`--output DATEI`	Name der Ausgabedatei

Für eine generelle Konfiguration des Befehls kann eine Datei `updatedb.conf` benutzt werden.

In vielen Distributionen wird das Update automatisch durch ein mit `cron` gestartetes Skript durchgeführt. Bei S.u.S.E. Linux 6.4 ist das Skript `aaa_base_updatedb` verantwortlich.

```
root@defiant:/etc/cron.daily > cat aaa_base_updatedb
#!/bin/sh

#
# paranoia settings
```

```
#
umask 022

PATH=/sbin:/bin:/usr/sbin:/usr/bin
export PATH

#
# get information from /etc/rc.config
#
if [ -f /etc/rc.config ] ; then
    . /etc/rc.config
fi

#
# update database for locate
#
if [ -n "$RUN_UPDATEDB" -a "$RUN_UPDATEDB" = "yes" -a \
     -x /usr/bin/updatedb ] ; then

    # avoid error messages from updatedb when using user nobody for find.
    cd /

    PARAMS="`test -n "$RUN_UPDATEDB_AS" && \
            fgrep localuser /usr/bin/updatedb > /dev/null && \
            echo --localuser=$RUN_UPDATEDB_AS`"

    PARAMS="$PARAMS `test -n "$UPDATEDB_PRUNEPATHS" && \
            echo --prunepaths=\'$(eval echo $UPDATEDB_PRUNEPATHS)\'`"

    PARAMS="$PARAMS `test -n "$UPDATEDB_NETUSER" && \
            echo --netuser=$UPDATEDB_NETUSER`"

    PARAMS="$PARAMS `test -n "$UPDATEDB_NETPATHS" && \
            echo --netpaths=\'$(eval echo $UPDATEDB_NETPATHS)\'`"

    eval /usr/bin/updatedb $PARAMS 2> /dev/null
fi

exit 0
```

11.4 Disk Quotas

Disk Quotas sind ein Mittel um den Platz, den jeder Benutzer oder Gruppe benutzen kann, einzuschränken. Sie erleichtern dem Administrator die Verwaltung des Systems. Diese Speichernutzungsreglementierung arbeitet Partitionsweise. Für die Partitionen können verschiedenen Quoten oder gar keine festgelegt werden. Um diese Funktion nutzen zu können, muß die Disk-Quota-Unterstützung auch in den Kernel einkompiliert werden.

Auch können für einzelne Benutzer oder für Gruppen Quoten festgelegt werden. Diese Quoten sind unabhängig voneinander. Erreicht das Kontingent der Gruppe oder des Benutzers das Limit, kann der Benutzer keine Daten mehr speichern.

Sollte ein Benutzer durch die Gruppenbegrenzung keinen Platz mehr haben, kann er das dadurch umgehen, daß er die Gruppe wechselt.

Im Normalfall sind die Festplattenquoten abgeschaltet. Um sie wieder zu aktivieren ist ein Eintrag für die Partition in `/etc/fstab` notwendig. Also Option wird der Platte der Eintrag `usrquota` für Benutzer und `grpquota` für Gruppen hinzugefügt.

```
/dev/hda1      /         ext2     defaults,grpquota          1  1
/dev/hda2      /home     ext2     defaults,usrquota          1  1
/dev/hda3      /server   ext2     defaults,usrquota,grpquota 1  1
```

Der Superuser muß dann in der Wurzel der Partition eine Datei `quota.user` bzw. `quota.group` einrichten. Diese Dateien sollten nur für den Superuser schreib- und lesbar sein.

```
root@defiant: ~> touch /quota.user /quota.group
root@defiant: ~> chmod 600 /quota.user /quota.group
```

Danach muß das System neu gestartet werden, um die Änderungen im System gültig zu machen. Um die Quoten einzuschalten wird das Kommando `quotaon` und um die Quoten auszuschalten das Kommando `quotaoff` benutzt. Um die Quoten einzurichten, wird der Befehl `edquota` verwendet. Dabei kann die Anzahl der Inodes, der Speicherplatz oder beides als Begrenzung dienen. Dabei spielen drei Parameter eine wichtige Rolle.

Weiche Limit (*soft limit*): Die durch das weiche Limit gesetzte Grenze kann für eine bestimmte Zeit überschritten werden.

Harte Limit (*hart limit*): Nach Erreichen des harten Limits ist es nicht mehr möglich Dateien zu erstellen. Es gibt keine Gnadenfrist.

Gnadenfrist (*grace period*): Für diesen Zeitraum kann das weiche Limit überschritten werden. Standardeinstellung sind sieben Tage.

11.4.1 quota

Mit dem Befehl `quota` kann der Superuser die Quoten überprüfen. Der normale Benutzer kann nur seine Quoten und die seiner Gruppe sehen.

```
quota [OPTIONEN] [WER]
```

Normal zeigt `quota` die Quoten der in `/etc/fstab` eingetragenen Dateisysteme.

Optionen	
`-u BENUTZER`	Zeigt die Quoten des angegebenen Benutzers
`-g`	Zeigt neben den persönlichen Quoten auch die Quoten der Gruppen, in denen der Benutzer Mitglied ist
`-q`	Zeigt nur Dateisysteme an in denen die Quote überschritten wurden.

11.4.2 quotaon

Der Befehl `quotaon` startet die Quotenüberwachung.

```
quotaon [OPTIONEN]
```

Optionen	
`-a`	Die Quoten werden für alle Dateisysteme mit rw-Recht aus der `/etc/fstab` aktiviert. Dies sollte jedesmal beim Boot-Vorgang passieren.
`-v`	Zeigt die Informationen über alle Systeme, auf denen die Quoten aktiviert sind.
`-u`	Schaltet die Quoten für benannte User an
`-q`	Schaltet die Quoten für benannte Gruppen an

Die Quoten sollten beim Booten jedesmal aktiviert werden. Deshalb muß das Init-Skript um einen Eintrag erweitert werden.

```
# Check quota and then turn quota on.
if [ -x /usr/sbin/quotacheck ]
  then
    echo "Checking quotas. This may take some time."
```

```
    /usr/sbin/quotacheck -avug
    echo " Done."
  fi
if [ -x /usr/sbin/quotaon ]
  then
    echo "Turning on quota."
    /usr/sbin/quotaon -avug
  fi
```

11.4.3 quotaoff

Der Befehl `quotaoff` beendet die Quotenüberwachung.

`quotaoff [OPTIONEN]`

Optionen

`-a`	Die Quoten werden für alle Dateisysteme mit rw-Recht aus der `/etc/fstab` deaktiviert. Dies sollte jedesmal beim Shutdown-Vorgang passieren.
`-v`	Zeigt die Informationen über alle Systeme, auf denen die Quoten deaktiviert sind.
`-u`	Schaltet die Quoten für benannte User aus
`-q`	Schaltet die Quoten für benannte Gruppen aus

11.4.4 edquota

Das Kommando `edquota` ist für das Anpassen der Quoten zuständig. Es öffnet dafür den Standardeditor (Variable `EDITOR`).

`edquota [OPTIONEN]`

Optionen

`-u BENUTZER`	Editiert die Quote des angegebenen Benutzers
`-g BENUTZER`	Editiert die Quote der angegebenen Gruppe
`-p BENUTZER`	Dupliziert die Quote des Benutzers (mit `-u`) oder die Quote der Gruppe (mit `-g`).
`-t`	Editiert die weichen Limits

Der Befehl `edquota -u tapico` öffnet den Abschnitt der Datei `quota.user` im Standardeditor.

```
Quotas for user tapico:
  /dev/hda3: blocks in use: 2594, limits (soft = 5000, hard = 6500)
           inodes in use: 356, limits (soft = 1000, hard = 1500)
```

Jedes Limit kann dabei drei verschiedene Werte annehmen.

- Die Zahl `0` steht für kein Limit.
- Die Zahl `-1` steht für die Standardeinstellungen.
- Jede ganze Zahl größer als Null steht für die Größe in KByte oder Stück.

Die Standardeinstellungen werden in der Datei `quota.h` eingestellt.

11.4.5 repquota

Dieser Befehl kann dazu benutzt werden den Status der Quoten zu überprüfen.

`repquota [OPTIONEN] [DATEISYSTEM]`

Dabei werden die Nutzung und die Quoten der Partition angezeigt.

Optionen

`-a`	Zeigt alle Dateisysteme an, die in `/etc/fstab` eingetragen sind.
`-v`	Zeigt alle Quoten an, selbst wenn der Benutzer keinen Platz auf der Partition beansprucht.
`-g GRUPPE`	Zeigt die Quoten für die angegebene Gruppe an.
`-u BENUTZER`	Zeigt die Quoten für den angegebenen Benutzer an. Nur der Superuser kann dies für andere Benutzer machen.

11.4.6 quotacheck

Der Befehl `quotacheck` aktualisiert die Quoteninformationen, also wieviel Platz von den einzelnen Benutzern und Gruppen belegt wird.

```
quotacheck [OPTIONEN]
```

Normalerweise wird der Befehl nach einem `fsck` ausgeführt.

Optionen

`-v`	Zeigt Informationen über den Verlauf des Programms an
`-u`	Führt einen Benutzerscan aus. (UID als Argument)
`-g`	Führt einen Gruppenscan aus. (GID als Argument)
`-a`	Überprüft alle Dateisysteme, die in der `/etc/fstab` stehen.
`-R`	Im Zusammenhang mit `-a` verwendet, um alle Systeme außer dem Wurzelverzeichnis zu testen.

11.4.7 quotastats

Der Befehl `quotastats` kalkuliert den benutzten Platz für jeden Benutzer und für jede Gruppe. Die Ergebnisse werden in den jeweiligen Dateien `quota.user` und `quota.group` gespeichert.

11.4.8 Einrichten von Disk Quotas mit SuSE

Um Disk Quotas nutzen zu können, müssen diese in den Kernel einkompiliert sein, was beim SuSE-Kernel der Fall ist. Die Quota-Utilities befinden sich im Paket *quota*, das auch installiert werden muß.

In diesem Beispiel wird die Partition `/dev/hdb1` mit dem Dateiystem ext2 am Mountpunkt `/daten` in das Dateisystem eingebunden. Für diese Partition sollen Disk Quotas für Benutzer angelegt werden. Daher muß in der `/etc/fstab` der Eintrag für `/dev/hdb1` um die Option `usrquota` ergänzt werden.

```
/dev/hdb1       /daten  ext2    defaults,usrquota 1 3
```

Im Wurzelverzeichnis der Partition muß nun die Verwaltungsdatei für die Disk Quotas eingesetzt werden. Bei SuSE kommt "aquota" zum Einsatz, daher der abweichende Name. Auf jeden Fall darf nur *root* die Datei lesen und schreiben.

```
voyager:~ # touch /daten/aquota.user
voyager:~ # chmod 600 /daten/aquota.user
```

Nun muß die Verwaltungsdatei eingerichtet werden. Dies übernimmt das Programm `quotacheck`.

```
voyager:~ # quotacheck -avR
quotacheck: WARNING -  Quotafile /daten/aquota.user was probably truncated.
Can't save quota settings...
quotacheck: Scanning /dev/hdb1 [/daten] done
quotacheck: Checked 3 directories and 1 files
voyager:~ # quotacheck -avR
quotacheck: Scanning /dev/hdb1 [/daten] done
quotacheck: Checked 3 directories and 1 files
```

Jetzt ist es an der Zeit für eine Benutzer Quoten anzulegen. Dies erfolgt über den Befehl `edquota`, der für die Eingabe der Werte den `vi` benutzt.

```
voyager:~ # edquota -u walter
```

Un so sollte der Eintrag lauten, wenn der Benutzer *walter* eine Beschränkung von 5 MB als weiches und hartes Limit besitzen soll.

```
Disk quotas for user walter (uid 510):
  Filesystem          blocks       soft       hard     inodes     soft     hard
  /dev/hdb1                0       5120       5120          0        0        0
```

Nach der Einrichtung sollten die Disk Quotas auch aktiviert werden. Danach liefert `repquota` den derzeitigen Stand der Dinge.

```
voyager:~ # quotaon /daten
voyager:/daten # repquota -av
*** Report for user quotas on device /dev/hdb1
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      --      32       0       0              4     0     0
walter    --       0    5120    5120              0     0     0

Statistics:
Total blocks: 7
Data blocks: 1
Entries: 2
Used average: 2,000000
```

Zum Testen legen wir uns eine ungefähr 2 MB große Datei an. Um die Datei zu füllen benutzen wir den Befehl `dd` (4.5.4), der aus dem Gerät `/dev/zero` Daten liest und sie in die Datei `testfile` schreibt. Wenn nun doe Datei dreimal rüberkopiert wird in den Verzeichnisbaumabschnitt `/daten`, sollten die Disk Quotas greifen.

```
walter@voyager:~> dd if=/dev/zero of=testfile count=2000000 bs=1
2000000+0 Records ein
2000000+0 Records aus
walter@voyager:~> cp testfile /daten/test/testfile1
walter@voyager:~> cp testfile /daten/test/testfile2
walter@voyager:~> cp testfile /daten/test/testfile3
ide0(3,65): write failed, user block limit reached.
cp: Schreiben von »/daten/test/testfile3«: Der zugewiesene Plattenplatz (Quota)
ist überschritten
```

Mit den Quotas klappt das ja schon ganz gut. Allerdings sind die Quotas nach dem Neustart des Rechners nicht mehr aktiv und müssen per Hand gestartet werden. Abhilfe schafft hier eines der vielen `init`-Skripte. Das Skript `quota` startet die Quoten und `boot.quota` sorgt dafür, daß nach einem unsauberen Beenden des Dateisytems die Quoten überprüft werden. Der Befehl `insserv` sorgt dafür, daß das angegebene Skript nun auch in den im Skript angegebenen Runleveln startet.

```
yoyager:~ # cd /etc/init.d
voyager:/etc/init.d # insserv quota
voyager:/etc/init.d # insserv boot.quota
```

Mit dem Befehl `quota` können Sie sich nun die Quoten der Benutzer anschauen.

```
voyager:~ # quota
Disk quotas for user root (uid 0): none
voyager:~ # quota walter
Disk quotas for user walter (uid 501):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
      /dev/hdb1    5120*   5120    5120               3       0       0
```

Manchmal kann es vorkommen, daß durch ungünstige Umstände die Anzahl der benutzten Ressourcen in der Verwaltungsdatei mit der Wirklichkeit nicht mehr übereinstimmt. Dies passiert z. B. wenn Dateien verändert wurden, während das Quota-System aus war. Der Befehl `quotacheck` ändert diese Information. Allerdings nur unter Verwendung einiger Schalter.

```
voyager:~ # repquota -av
*** Report for user quotas on device /dev/hdb1
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      --      32       0       0              4     0     0
walter    --    5120    5120    5120              3     0     0

Statistics:
Total blocks: 7
Data blocks: 1
Entries: 2
Used average: 2,000000

voyager:~ # quotacheck -avR
quotacheck: Quota for users is enabled on mountpoint /daten so quotacheck might
damage the file.
Please turn quotas off or use -f to force checking.
voyager:~ # quotacheck -afvR
quotacheck: Cannot remount filesystem mounted on /daten read-only so counted
values might not be right.
Please stop all programs writing to filesystem or use -m flag to force checking.
voyager:~ # quotacheck -amfvR
quotacheck: Scanning /dev/hdb1 [/daten] done
quotacheck: Checked 4 directories and 1 files
voyager:~ # repquota -av
*** Report for user quotas on device /dev/hdb1
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      --      32       0       0              4     0     0
walter    --       0    5120    5120              0     0     0

Statistics:
Total blocks: 7
Data blocks: 1
Entries: 2
Used average: 2,000000
```

XIV | Arbeitsblatt
Dateisystem und Disk Quotas | 14.1

Zu Beginn der Aufgabe sollten Sie sich im Heimatverzeichnis befinden! Wechseln Sie das Verzeichnis nur, falls Sie dazu ausdrücklich aufgefordert werden. Einige Aufgaben lassen sich nicht ausführen! Versuchen Sie in diesen Fällen die Fehlermeldung zu interpretieren.

1 Loggen Sie sich als Benutzer *walter* auf der Konsole 1 ein.

2 Kopieren Sie die Datei `passwd` aus `/etc` ins aktuelle Verzeichnis!

3 Wem gehört die Datei `passwd` a) in `/etc`, b) im aktuellen Verzeichnis?

4 Geben Sie der Datei `passwd` den zusätzlichen Hardlink `meingottwalter`!

5 Kopieren Sie die Datei `/bin/mv` ins aktuelle Verzeichnis !

6 Legen Sie ein Verzeichnis `test` an !

7 Kopieren Sie alle Dateien des aktuellen Verzeichnisses ins Verzeichnis `test` und wechseln Sie nach `test`!

8 Benennen Sie `passwd` um in `halleluja`!

9 Stellen Sie fest, wie viele Hardlinks die Datei `passwd` in `/etc` hat.

10 Suchen Sie die Datei `passwd` im gesamten Dateisystem !

11 Suchen Sie nach allen Dateien mit Namen `utmp` im gesamten Dateisystem!

12 Erstellen Sie in Ihrem Heimatverzeichnis das Verzeichnis `teest2`!

13 Geben Sie der Datei `mv` in Ihrem Heimatverzeichnis den zusätzlichen Namen (Softlink) `move`!

14 Lassen Sie sich alle Dateien im Dateisystem anzeigen, die Ihnen gehören!

15 Lassen Sie sich ausführliche Informationen zu allen Dateien anzeigen, die im Verzeichnis `/usr` liegen und größer als 1 kB sind!

16 Wieviele Dateien gibt es im Verzeichnis `/dev` ?

17 Wechseln Sie zur Konsole 2 und loggen Sie sich dort als *root* ein.

18 Konfigurieren Sie das Heimatverzeichnis so, daß der Benutzer *Willi* dort nur 20 MB Plattenplatz belegen darf. Welchen Dienst nutzen Sie dafür. Beschreiben Sie die Vorgehensweise.

Notizen:

Kapitel 12

Bootvorgang und Prozeßverwaltung

12.1 Linux startet

Linux zu starten ist ganz einfach. Sie müssen dazu nur den Rechner einschalten und einen Bootmanager (bzw. *boot loader*) verwenden oder von einer Diskette booten.

Wenn Sie Ihren Rechner starten, werden einige Schritte während des Startvorgangs durchgeführt. Diesen Zeitraum nennt man POST (*Power On Self Test*). Danach kann das Betriebssystem geladen werden. Ein Bootmanager wie LILO übernimmt die Aufgabe den Kernel (Betriebssystemkern) zu laden.

Die Art und Weise wie das Betriebssystem geladen wird, hängt davon ab, wie Ihr System eingerichtet ist. Das macht aber nichts, da die grundlegenden Schritte, wie Sie das Betriebssystem starten und zum Laufen bringen, gleich sind. Es ist sehr wichtig den Vorgang und die Schritte des Bootens zu verstehen, um Probleme beim Startvorgang beseitigen zu können.

Der Kernel befindet sich entweder auf einer Diskette oder auf der Festplatte im Verzeichnis `/boot`[1]. Der gepackte Kernel ist normalerweise in der Datei `vmlinuz-versionsnummer`. Bei SuSE lautet der Dateiname nur `vmlinuz`. Wie auch immer, der Name und der Dateipfad sind eigentlich unwichtig. Wichtig ist nur, daß der Kernel während des Bootvorgangs lokalisiert werden kann.

Als erstes wird der Kernel in den Speicher geladen. Normalerweise ist der Kernel gepackt. Nur der Code für das Entpacken ist natürlich in ungepackter Form enthalten. Sie können das Programm `gunzip` zum Entpacken des Kernels verwenden. Der Kernel enthält wichtige Informationen wie z. B. den Ort des Wurzeldateisystems.

Während des ganzen Ladevorgangs gibt der Kernel Meldungen aus und speichert sie in der Datei `/var/log/messages` (Abschnitt 13.2.3). Um sich die Meldungen später anzusehen, können Sie den Befehl `dmesg` verwenden. Er benötigt keine Argumente und gibt die letzten Meldungen aus dem Meldungspuffer des Kernels aus.

Diese Nachrichten enthalten

- den Konsolentyp und -zeichensatz,
- die Erkennung des PCI-Bus und der eingebauten PCI-Karten,
- eine Angabe über die Prozessorgeschwindigkeit,
- den verfügbaren Arbeitsspeicher,
- den CPU-Typ,
- die Versionsnummer des Kernels,
- die geladenenen Gerätetreiber,
- den zur Verfügung stehen Swap-Speicher und

[1] Bei älternen Linux-Distributionen kann er sich auch im Wurzelverzeichnis (`/`) befinden.

- die Netzwerkadapter mit ihren Einstellungen.

Nachdem der Kernel gestartet wurde, mounted er das Wurzel-Dateisystem, das normalerweise auf der Festplatte liegt. Nach dem Mounten wird die Kontrolle an die Programme auf der Festplatte übergeben, während der Kernel im Speicher bleibt.

12.1.1 dmesg

Der Befehl `dmesg` gibt letzten Meldungen des Kernels aus. In den meisten Fällen wird er dazu benutzt um sie die Bootmeldungen nach dem Starten des Systems noch einmal in Ruhe anzuschauen.

```
dmesg [OPTIONEN]
```

Hier noch ein Beispiel für die vom Kernel gelieferten Meldungen über den Startvorgang.

```
root@defiant:/ > dmesg
Linux version 2.2.14 (root@defiant) (gcc version 2.95.2 19991024) #2 Wed Jul 26 20:49:06 MEST 2000
Detected 365818579 Hz processor.
Console: colour VGA+ 80x25
Calibrating delay loop... 730.73 BogoMIPS
Memory: 62800k/65472k available (1280k kernel code, 412k reserved, 936k data, 44k init, 0k bigmem)
Dentry hash table entries: 8192 (order 4, 64k)
Buffer cache hash table entries: 65536 (order 6, 256k)
Page cache hash table entries: 16384 (order 4, 64k)
CPU: Intel Celeron (Mendocino) stepping 0a
Checking 386/387 coupling... OK, FPU using exception 16 error reporting.
Checking 'hlt' instruction... OK.
POSIX conformance testing by UNIFIX
PCI: PCI BIOS revision 2.10 entry at 0xf0200
PCI: Using configuration type 1
PCI: Probing PCI hardware
Linux NET4.0 for Linux 2.2
Based upon Swansea University Computer Society NET3.039
NET4: Unix domain sockets 1.0 for Linux NET4.0.
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP, IGMP
TCP: Hash tables configured (ehash 65536 bhash 65536)
Initializing RT netlink socket
Starting kswapd v 1.5
Detected PS/2 Mouse Port.
pty: 256 Unix98 ptys configured
apm: BIOS version 1.2 Flags 0x0f (Driver version 1.12)
Real Time Clock Driver v1.09
loop: registered device at major 7
Uniform Multi-Platform E-IDE driver Revision: 6.30
ide: Assuming 40MHz system bus speed for PIO modes; override with idebus=xx
ALI15X3: IDE controller on PCI bus 00 dev 78
ALI15X3: not 100% native mode: will probe irqs later
ALI15X3: simplex device:  DMA disabled
ide0: ALI15X3 Bus-Master DMA disabled (BIOS)
ALI15X3: simplex device:  DMA disabled
ide1: ALI15X3 Bus-Master DMA disabled (BIOS)
hda: IBM-DBCA-204860, ATA DISK drive
hdc: CD-224E, ATAPI CDROM drive
ide0 at 0x1f0-0x1f7,0x3f6 on irq 14
ide1 at 0x170-0x177,0x376 on irq 15
hda: IBM-DBCA-204860, 4645MB w/420kB Cache, CHS=592/255/63
hdc: ATAPI 24X CD-ROM drive, 128kB Cache
Uniform CDROM driver Revision: 2.56
Floppy drive(s): fd0 is 1.44M
```

```
FDC 0 is a National Semiconductor PC87306
scsi0 : SCSI host adapter emulation for IDE ATAPI devices
scsi : 1 host.
scsi : detected total.
Partition check:
 hda: hda1 hda2 < hda5 > hda3 hda4
VFS: Mounted root (ext2 filesystem) readonly.
Freeing unused kernel memory: 44k freed
Adding Swap: 128516k swap-space (priority -1)
Serial driver version 4.27 with HUB-6 MANY_PORTS MULTIPORT SHARE_IRQ enabled
ttyS00 at 0x03f8 (irq = 4) is a 16550A
```

12.2 Der Init-Daemon

Das Letzte, was der Kernel macht, ist den Init-Daemon (`/sbin/initd`) zu starten. Dieser bleibt so lange aktiv, bis das System heruntergefahren wird. Damit ist `init` der erste laufende Prozeß. Alle weiteren Prozesse werden entweder vom Init-Daemon direkt oder indirekt durch Subprozesse von `init` gestartet. Dieser Daemon ist also für die Erstellung der Prozesse verantwortlich, die der Rest des Systems nutzt. Als Beispiel ist hier die Login-Shell zu nennen. Daneben sorgt er dafür, das bestimmte Prozesse nach ihrem Ende wieder neu gestartet werden. Er sorgt z. B. dafür, daß nach dem Logout die Konsole neu gestartet wird, um eine erneutes Einloggen zu ermöglichen. Bei einem Shutdown ist `init` der letzte noch laufende Prozeß, der sich um das korrekte Beenden aller anderen Prozesse kümmert.

Der Init-Daemon und seine Aktionen werden durch die Konfigurationsdatei `/etc/inittab` gesteuert.

12.2.1 /etc/inittab

Jede Anweisungszeile der Datei `/etc/inittab` besteht aus vier Feldern. Von links nach rechts gesehen sind dies die ID, der Runlevel, die Aktion und der Prozess.

Feld: ID

Die ID besteht aus ein oder zwei Zeichen, die den Einstiegspunkt bestimmen. Sie muß eindeutig sein und ist oft der Gerätename.

Feld: Runlevel

Dies Feld gibt den Runlevel an, für den diese Zeile gültig ist. Hier können mehrere Runlevel angegeben werden. Wird das Feld leer gelassen, dann gilt die Anweisung für alle Runlevels.

Feld: Aktion

Mit der Aktion wird angegeben, wie mit dem Eintrag verfahren werden soll. So kann z. B. festgelegt werden, daß der Prozess automatisch neu gestartet wird, wenn er beendet wurde. Eine Übersicht über die möglichen Einträge bietet die Tabelle 12.1. Weitere Informationen erhalten Sie mit `man inittab`.

Feld: Prozeß

Im letzten Feld wird das Linux-Kommando oder das Programm aufgeführt, daß in der Situation gestartet werden soll.

Das folgende Listing gibt eine gekürzte Fassung der `/etc/inittab` der SuSE-Linux 6.4 Distribution wieder.

```
# /etc/inittab

# default runlevel
```

Eintrag	Aktion
`boot`	Startet, wenn die `/etc/inittab` zum ersten Mal gelesen wird
`bootwait`	Startet nach den Einträgen für `boot`
`ctrlaltdel`	Startet nach der Tasteneingabe <Strg>+<Alt>+<Entf>
`initdefault`	Definiert den Standard-Runlevel
`once`	Startet beim Wechsel eines Runlevels
`ondemand`	Hält Prozesse in bestimmten Runlevels am Laufen
`powerfail`	Startet, wenn ein Stromausfallsignal empfangen wurde
`sysinit`	Startet, bevor die Einträge `boot` und `bootwait` ausgeführt werden.
`respawn`	Startet den Prozeß nach seinem Ende neu
`wait`	Startet den Prozeß einmal beim Erreichen des Runlevels und `init` wartet auf sein Ende

Tabelle 12.1: Einträge für das Feld Aktion in der `/etc/inittab`

```
id:2:initdefault:

# check system on startup
# first script to be executed if not booting in emergency (-b) mode
si:I:bootwait:/sbin/init.d/boot

# /sbin/init.d/rc takes care of runlevel handling
#
# runlevel 0 is halt
# runlevel S is single-user
# runlevel 1 is multi-user without network
# runlevel 2 is multi-user with network
# runlevel 3 is multi-user with network and xdm
# runlevel 6 is reboot
l0:0:wait:/sbin/init.d/rc 0
l1:1:wait:/sbin/init.d/rc 1
l2:2:wait:/sbin/init.d/rc 2
l3:3:wait:/sbin/init.d/rc 3
l6:6:wait:/sbin/init.d/rc 6

# what to do in single-user mode
ls:S:wait:/sbin/init.d/rc S
~~:S:respawn:/sbin/sulogin

# what to do when CTRL-ALT-DEL is pressed
ca::ctrlaltdel:/sbin/shutdown -r -t 4 now

# what to do when power fails/returns
pf::powerwait:/sbin/init.d/powerfail    start
pn::powerfailnow:/sbin/init.d/powerfail now
po::powerokwait:/sbin/init.d/powerfail  stop

# getty-programs for the normal runlevels
# <id>:<runlevels>:<action>:<process>
# The "id" field  MUST be the same as the last
# characters of the device (after "tty").
1:123:respawn:/sbin/mingetty --noclear tty1
2:123:respawn:/sbin/mingetty tty2
3:123:respawn:/sbin/mingetty tty3
4:123:respawn:/sbin/mingetty tty4
5:123:respawn:/sbin/mingetty tty5
6:123:respawn:/sbin/mingetty tty6
```

```
# end of /etc/inittab
```

12.2.2 Die Runlevel

Ein Runlevel definiert einen Satz von Prozessen, der als Teil der Bootsequenz gestartet wird. Das kann von einem minimalen Satz von Prozessen zur Systemadministration bis zu einer Vielzahl von Prozessen für alle eingerichteten Geräte reichen.

Die Nummer, die mit dem Runlevels verknüpft sind, variieren mit den verschiedenen Distributionen. Sie werden in der `/etc/inittab` aufgeführt. So sieht z. B. der Eintrag für SuSE-Linux 6.4 so aus:

```
# runlevel 0 is halt
# runlevel S is single-user
# runlevel 1 is multi-user without network
# runlevel 2 is multi-user with network
# runlevel 3 is multi-user with network and xdm
# runlevel 6 is reboot
```

und der für SuSE-Linux 7.3 so aus:

```
# runlevel 0  is  System halt   (Do not use this for initdefault!)
# runlevel 1  is  Single user mode
# runlevel 2  is  Local multiuser without remote network (e.g. NFS)
# runlevel 3  is  Full multiuser with network
# runlevel 4  is  Not used
# runlevel 5  is  Full multiuser with network and xdm
# runlevel 6  is  System reboot (Do not use this for initdefault!)
```

Die Einteilung der Runlevel von 7.3 ist auch aktuell (8.1) noch gültig.

Um zu bestimmen, in welchem Runlevel Sie sich gerade befinden, können Sie den Befehl `runlevel` verwenden.

Die Runlevels 0 und 6 sind normalerweise reserviert für die Zustände `halt` und `reboot`. Aber egal wie die Nummernvergabe auch läuft, der Init-Daemon arbeitet nur die Zeilen der `/etc/inittab` ab, die mit dem Runlevel verknüpft sind.

Der Runlevel *Single-User* unterscheidet sich etwas von den anderen Runlevels. Er wird gerne zur Administration und Konfiguration des Systems verwendet, wie z. B. die Reparatur eines beschädigten Dateisystems. Niemand kann sich in das System einloggen, wenn es sich in diesem Modus befindet. Trotzdem können mehrere Prozesse gleichzeitig ausgeführt werden. Dieser Modus ist auch der einzige, in dem der Init-Daemon nicht die Datei `/etc/inittab` ausliest und abarbeitet. Normalerweise starten Sie das Programm `/bin/su` um dann als Superuser zu arbeiten.

Wenn Sie einen höheren Runlevel einstellen als den Single-User-Modus, dann startet das System immer im Multiuser-Betrieb. Dann werden die Zeilen mit den Aktionen `sysinit`, `boot` und `bootwait` ausgeführt. Oft werden in diesen Zeilen auch die Dateisysteme gemountet.

Danach führt der Init-Daemon die anderen Zeilen aus, die mit dem Runlevel verknüpft sind. Der angestrebte Runlevel wird in der Zeile
`id:2:initdefault:`
definiert. Denken Sie daran, daß diese Zeile keinen Prozeß startet, sondern den Runlevel festlegt, mit dem gestartet werden soll. Ein Wechsel des Runlevels erfolgt mit den Befehlen `init` und `telinit`.

Im nächsten Abschnitt wird definiert, welche Startskripte ausgeführt werden sollen. Diese Zeile könnte lauten
`si::sysinit:/etc/rc.d/rc.sysinit`
oder z. B. bei SuSE
`si:I:bootwait:/sbin/init.d/boot`

Die ID `si` am Anfang der Zeile wird intern vom Init-Daemon verwendet. Diese Zeile startet das Skript `rc.sysinit`, welches Aufgaben wie die Aktivierung der Swap-Partition, Starten von `fsck` und Mounten des Dateisystems übernimmt.

Der folgende Abschnitt definiert die Skripte, die für den jeweiligen Runlevel ausgeführt werden sollen. Sie stehen im Verzeichnis `/etc/rc.d` bzw. bei SuSE im Verzeichnis `/etc/init.d`.

```
l0:0:wait:/sbin/init.d/rc 0
l1:1:wait:/sbin/init.d/rc 1
l2:2:wait:/sbin/init.d/rc 2
l3:3:wait:/sbin/init.d/rc 3
l6:6:wait:/sbin/init.d/rc 6
```

Das `rc`-Skript wird mit einem Parameter, der den Runlevel angibt, aufgerufen. Dabei führt es Skripte aus dem Verzeichnis `rc`*n*`.d` aus, wobei *n* für die Nummer des Runlevels steht.

Daneben kann `rc` auch Kernel-Module aufrufen. Dies sind Teile des Kernels, die nicht fest einkompiliert wurden. Sie können bei Bedarf nachgeladen und wieder entladen werden. Meistens handelt es sich hierbei um Gerätetreiber, die nicht oft benötig werden.

Die Datei `/etc/modules.conf` enthält die Liste der Module mit ihren Parametern. Oft handelt es sich bei den Parametern um I/O-Adressen und Interrupts. Ein Ausschnitt aus der `/etc/modules.conf` könnte so aussehen.

```
# Aliases - specify your hardware

alias parport_lowlevel    parport_pc
options parport_pc io=0x378 irq=none,none

isp16_cdrom_type=Sanyo

options ne      io=0x300

alias block-major-1       rd
alias block-major-2       floppy

alias char-major-4        serial
alias char-major-5        serial
alias char-major-6        lp
alias char-major-9        st
```

Als nächstes werden die Prozesse abgearbeitet, die bei jedem Runlevel ausgeführt werden. Dies ist u. a. der `getty`-Prozeß, der die virtuellen Terminals anlegt.

```
# getty-programs for the normal runlevels
1:123:respawn:/sbin/mingetty --noclear tty1
2:123:respawn:/sbin/mingetty tty2
3:123:respawn:/sbin/mingetty tty3
4:123:respawn:/sbin/mingetty tty4
5:123:respawn:/sbin/mingetty tty5
6:123:respawn:/sbin/mingetty tty6
```

Nach dem Lesen und abarbeiten der `/etc/inittab` bleibt der Init-Daemon aktiv. Er überwacht das System auf Anweisungen den Runlevel zu ändern oder ob ein Prozeß gestoppt wurde. Wenn einer der Kindsprozesse von `init` gestoppt ist, macht der Daemon zwei Sachen.

1. Er liest die `/etc/inittab` wieder. Wenn dort der erneute Start als Aktion eingetragen ist, wird der Prozeß wieder neu gestartet.

2. Dann trägt er in die Dateien `/var/log/wtmp` und `/var/run/utmp` die Tatsache ein, daß der Prozeß beendet wurde und aus welchem Grund.

Genauso liest der Daemon die `/etc/inittab` ein, wenn der Runlevel geändert wurde oder er das Signal über einen Stromausfall bekommt.

Änderungen in der `/etc/inittab` können Sie mit einem einfachen Editor machen während das System läuft. Bevor die Änderungen aber wirksam werden, muß erst wieder die `/etc/inittab` gelesen werden. Dies passiert

- wenn der Runlevel geändert wird,
- wenn ein Stromausfall signalisiert wird und
- wenn Sie das Kommando `init q` eingeben, das direkt das erneute Lesen der Datei bewirkt.

Seien Sie sehr vorsichtig und gründlich, wenn Sie die Datei `/etc/inittab` bearbeiten wollen. Ein Fehler in der Datei kann dazu führen, daß Sie sich nicht mehr am System einloggen können. Legen Sie deshalb vorher eine Kopie der Datei an und erstellen Sie eine Bootdiskette für den Notfall. Es besteht die Möglichkeit, daß Sie noch im Single-User-Modus starten können, darauf sollten Sie aber nicht vertrauen. Auch ist es möglich, daß Sie eine Endlosschleife programmieren, so daß ein Prozeß immer wieder gestartet wird. Für diesen Fall überprüft der Init-Daemon, ob ein Prozeß innerhalb von 2 Minuten 10 mal gestartet wurde. Er gibt dann eine Warnung aus und wartet 5 Minuten, bevor er den Prozeß wiederum startet.

12.2.3 runlevel

Der Befehl `runlevel` ermittelt den aktuellen und den vorherigen Runlevel.

```
runlevel [UTMP]
```

Der Befehl liest dafür die Datei `/var/run/utmp` aus um den Runleveleintrag zu finden und gibt dann den alten und neuen Runlevel getrennt durch ein Leerzeichen aus. Falls es keinen vorherigen Runlevel gegeben hat, wie es nach einem Start der Fall ist, dann wird der Buchstabe N angezeigt.

Ist der Runlevel nicht ermittelbar, dann zeigt `runlevel` das Wort "unknown" und beendet sich mit einem Fehlersignal.

Optional kann der genaue Name der Datei `utmp` mit Pfad angegeben werden, wenn sich sich z. B. nicht in `/var/run` befindet.

Beispiel

Der Befehl `runlevel` zeigt den aktuellen und vorherigen Runlevel an.

```
ole@enterprise:~> /sbin/runlevel
N 5
```

Die Informationen dazu bezieht er dabei aus der Datei `/var/run/utmp`.[2]

```
ole@enterprise:~> od -c /var/run/utmp | grep ^00006
0000600 001  \0  \0  \0   5   N  \0  \0   ~  \0  \0  \0  \0  \0  \0  \0
0000620  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0
0000640  \0  \0  \0  \0  \0  \0  \0  \0   ~   ~  \0  \0   r   u   n   l
0000660   e   v   e   l  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0
```

12.2.4 init

Der Befehl `init` ermöglich das Wechseln des Runlevels.

```
init [OPTIONEN] [RUNLEVEL]
```

Beim Aufruf des Befehl sendet `init` ein Stopsignal an alle Prozesse, die nicht für den neuen Runlevel definiert sind. Danach werden die Prozesse "getötet" und die Prozesse für den Runlevel, die noch nicht laufen, werden gestartet.

Die Eingabe des Runlevels '`s`' oder '`S`' bewirkt einen Wechsel in den Single-User-Modus. Die Angabe '`q`' führt zu einem erneuten Einlesen und abarbeiten der `/etc/inittab`.

[2] Beispiel aus SuSE 8.0 und 8.1.

Beispiel
Um das X-Window-System mit SAX zu konfigurieren, sollte es möglichst beendet werden. Ist die graphische Anmeldung aktiv, so startet sich das X-Window-System nach einer Beendigung automatisch neu für den Anmeldeschirm. Wenn nun der Windowmanager keine Option zum Wechseln in den Konsolenmodus anbietet, so kann man dies durch folgende Schritte erreichen.

1. Wechseln Sie zur Textkonsole tty1 (oder auch einer anderen ;-)).
2. Loggen Sie sich als *root* ein.
3. Wechseln Sie mit dem Befehl `init 3` in den Runlevel für die Konsole.
4. Das X-Window-System ist jetzt noch immer aktiv. Wenn Sie es aber jetzt beenden, wird die graphische Anwendung nicht mehr automatisch gestartet.

12.2.5 telinit

Das Kommando `telinit` ist im Prinzip nur ein Link auf das Programm `init`.

```
telinit [OPTIONEN] [RUNLEVEL]
```

Der Befehl sorgt dafür, daß `init` erst mit 5 Sekunden Verzögerung ausgeführt wird. Mit dem Schalter `-t SEKUNDEN` kann eine andere Verzögerung definiert werden.

12.2.6 Automatisches Starten von Diensten

Soll ein Dienst beim Booten des Systems automatisch gestartet werden, so kann das nur durch den Daemon `initd` erfolgen. Dafür müßte jedes Programm in der `/etc/inittab` zusammen mit den passenden Runleveln eingetragen werden. Dies hat man durch die Einführung des Skriptes `/etc/init.d/rc` erleichtert.

```
enterprise:/etc/init.d # head -15 rc
#! /bin/bash
#
# Copyright (c) 1996-2002 SuSE Linux AG, Nuernberg, Germany.
# All rights reserved.
#
# Author: Florian La Roche <feedback@suse.de> 1996
#         Werner Fink <werner@suse.de> 1994-99,2000-2001
#
# /etc/init.d/rc   --    The Master Resource Control Script
#
# This file is responsible  for  starting/stopping  services
# when the runlevel changes.  If the action for a particular
# feature in the new run-level is the same as  the action in
# the previous run-level, this script will neither start nor
# stop that feature.
```

In dem Verzeichnis `/etc/init.d` stehen die Skripte zum Starten der Programme bzw. Dienste. Für jeden Runlevel existiert ein Verzeichnis in dem sich nun Links auf die Skripte befinden, die in diesem Runlevel gestartet werden sollen.

```
enterprise:/etc/init.d # ls -ld rc*
-rwxr-xr-x    1 root     root         6082 Mär  6 17:56 rc
drwxr-xr-x    2 root     root         4096 Mär 21 01:14 rc0.d
drwxr-xr-x    2 root     root         4096 Mai  2 22:07 rc1.d
drwxr-xr-x    2 root     root         4096 Mai  2 22:07 rc2.d
drwxr-xr-x    2 root     root         4096 Mai  6 20:19 rc3.d
```

```
drwxr-xr-x    2 root     root        4096 Mär 21 01:14 rc4.d
drwxr-xr-x    2 root     root        4096 Mai  6 20:19 rc5.d
drwxr-xr-x    2 root     root        4096 Mär 21 01:14 rc6.d
drwxr-xr-x    2 root     root        4096 Mai  2 22:07 rcS.d
enterprise:/etc/init.d # ls -lG rc1.d
insgesamt 8
lrwxrwxrwx    1 root            9 Mai  2 22:07 K03single -> ../single
lrwxrwxrwx    1 root            9 Mai  2 22:07 K12splash -> ../splash
lrwxrwxrwx    1 root            8 Mai  2 22:07 K13fbset -> ../fbset
lrwxrwxrwx    1 root           10 Mai  2 22:07 K16hotplug -> ../hotplug
lrwxrwxrwx    1 root           10 Mai  2 20:25 S07hotplug -> ../hotplug
lrwxrwxrwx    1 root            8 Mai  2 20:33 S10fbset -> ../fbset
lrwxrwxrwx    1 root            6 Mai  2 20:33 S11kbd -> ../kbd
lrwxrwxrwx    1 root            9 Mai  2 20:33 S11splash -> ../splash
lrwxrwxrwx    1 root            9 Mai  2 20:23 S20single -> ../single
```

Die Links, die mit "K" (kill) beginnen, stehen für die Dienste, die beim Verlassen des Runlevels beendet werden sollen und die Links, die mit "S" (start) beginnen, beziehen sich auf Dienste, die in diesem Runlevel zu starten sind. Im dem Bootskript selber steht oft drinnen, in welchen Runleveln dieser Dienst gestartet werden soll, wenn er überhaupt gestartet werden soll.

Um nun einen Dienst beim Booten starten zu können Sie nun entweder per Hand die Skripte mit Links versehen oder sich komfortabler mit dem Programm `insserv` behelfen. Unter YaST2 steht ihnen der Runlevel-Editor für diese Aufgabe zur Verfügung.

12.2.7 insserv

Das Programm `insserv` aktiviert installierte Init-Skripte, in dem es die passenden Links in den Runlevel-Verzeichnissen (`/etc/init.d/rc`*n*`.d`) setzt.

```
insserv [OPTIONEN] [SKRIPTE]
```

Dazu wertet das Programm den Kommentarblock des Skriptes aus, wie er z. B. in diesem Skript vorkommt.

```
enterprise:/etc/init.d # head -18 splash
#! /bin/bash
# Copyright (c) 1995-2000 SuSE GmbH Nuernberg, Germany.
#
# Author: Michael Schroeder <feedback@suse.de>
#
# /etc/init.d/splash
# /usr/sbin/rcsplash
#
# System startup script for console splash screens
#
### BEGIN INIT INFO
# Provides: splash
# Required-Start: $remote_fs fbset
# Required-Stop:
# Default-Start:  1 2 3 5 S
# Default-Stop:
# Description:    Splash screen setup
### END INIT INFO
```

Optionen

`-r`	Entfernt das Skript aus allen Runlevels
`-d`	Installiert das Skript in den vorgegebenen Runleveln

Konfiguriert wird `insserv` über die Konfigurationsdatei `/etc/insserv.conf`.

12.3 LILO

Der von Werner Almesberger geschriebene LILO (*LInux LOader*) ist der wohl am weitesten verbreitete Bootmanager (*boot loader*) unter Linux. Er kann sich auf der Festplatte oder auf einer Bootdiskette befinden, und wird ausgeführt, wenn das System geladen wird. Einige der Eigenschaften von LILO sind:

- LILO arbeitet mit DOS, LINUX, UNIX, OS/2, Windows 95/98 und Windows NT zusammen.
- LILO ersetzt den Masterbootrecord der Festplatte oder Diskette.
- LILO kann bis zu 16 verschiedene Bootimages auf verschiedenen Partitionen verwalten, wobei jedes einzelne durch ein Kennwort geschützt werden kann.
- LILO erlaubt es, daß die Boot-Sektor-Datei, die Map-Datei und das Bootimage sich auf verschiedenen Partitionen befinden kann.

LILO besteht aus zwei Teilen. Der erste Teil befindet sich normalerweise im Master Boot Record (MBR) der Festplatte. Dieser Programmteil wird während des Bootvorgangs vom BIOS gestartet. Seine Aufgabe besteht darin, den größeren zweiten Teil zu starten, der sich irgendwo anders auf einem Medium befinden kann. Der zweite Teil nun bietet dem Benutzer eine Möglichkeit zur Auswahl des gewünschten Betriebssystems, findet den Kernel, lädt ihn in den Speicher und startet ihn. Die Zweiteilung ist notwendig, da der MBR nur sehr wenig Platz (meistens nur 1 Block) bietet. Daher ist der erste Teil sehr kompakt und dient nur als Starter für den größeren zweiten Teil.

Da LILO auch andere Betriebssysteme laden kann, ist er bestens dafür geeignet als Wahlschalter für Systeme mit mehreren installierten Betriebssystemen zu agieren.

Wurde keine Wahlmöglichkeit geschaffen, dann wird das standardmäßig eingestellte Betriebssystem gebootet. Durch das Drücken einer Taste (Alt-, Shift- oder Strg-Taste) oder einem Eintrag in der `/etc/lilo.conf` kann dieser Vorgang abgebrochen und ein Boot-Prompt angezeigt werden.
`BOOT:`
oder auch
`LILO:`

Am Boot-Prompt können Sie den Namen des zu ladenden Betriebssystems angeben. Mit der Tab-Taste oder der Eingabe eines '`?`' wird Ihnen eine Liste der zu bootenden Betriebssysteme angezeigt. Sollte Sie keine Entscheidung treffen, so bootet LILO nach einer voreingestellten Zeit automatisch das erste Betriebssystem in der Liste.

```
LILO: <TAB>
linux*     windoof     oldlinux     betalinux
```

Beim Eintippen des Namens des Betriebssystems können Kernelparameter angehängt werden. Dies wird oft dazu benutzt das System im Single-User-Modus zu starten.

Die Einstellungen von LILO werden in seiner Standardkonfigurationsdatei `/etc/lilo.conf` festgelegt. Ein Bootloader wird normalerweise bei der Installation angelegt. Soll der Bootloader später verändert werden, so müssen die Änderungen in der Konfigurationsdatei noch in das System übertragen werden. Diese Aufgabe übernimmt das Programm `lilo`.

12.3.1 lilo

Der Befehl `lilo` entnimmt aus der Konfigurationsdatei `/etc/lilo.conf` Informationen über die zu startenden Kernel, Bildschirmeinstellungen, Bootplatte etc. und konfiguriert damit den Bootloader.

```
lilo [OPTIONEN]
```

`lilo` liest die Konfigurationsdatei und codiert die Informationen auf Basis der physikalischen Festplattenkonfiguration. (Was befindet sich in welchem physikalischem Block.) Dies ist notwendig, da es zum Startzeitpunkt noch kein Dateisystem bzw. keine Dateiverwaltung gibt. Diese Informationen werden in einer Map-Datei (z. B. `/boot/map`) gespeichert.

Option	Aktion
`-b BOOTDEV`	Gibt die Bootplatte an
`-C DATEINAME`	Andere Konfigurationsdatei verwenden
`-c`	Beschleunigtes Booten durch zusammenfassendes Lesen von benachbarten Sektoren
`-D KERNEL`	Benutze den Kernel mit der angegebenen Bezeichnung
`-d ZENTELSEK`	Wartezeit bis zum automatischen Booten
`-f DATEINAME`	Datei für die Plattenparameter (Standard: `/etc/disktab`)
`-I`	Fragt nach dem Pfad für den Kernel
`-i DATEINAME`	Name der Bootsectordatei (Standard: `/boot/boot.b`)
`-l`	Erzeugt lineare Sektoradressen
`-m DATEINAME`	Name für Map-Datei vorgeben
`-P [fix|ignore]`	Berichtigt oder ignoriert beschädigte Partitionstabellen
`-q`	Listet die Namen und Orte der zu bootenden Kernel aus der Map-Datei
`-R KOMMANDO`	Setzt den Standard für den nächsten Bootvorgang
`-r VERZEICHNIS`	Ändert die Wurzel des Betriebssystem vor allem Anderen; wird für die Fehlerbehebung verwendet
`-S DATEINAME`	Legt fest, wo der alte Bootsektor gespeichert werden soll; alte Sicherungen werden überschrieben
`-s DATEINAME`	Legt fest, wo der alte Bootsektor gespeichert werden soll; alte Sicherungen werden nicht überschrieben
`-t`	Testen der Konfiguration ohne Installation
`-u` oder `-U`	Deinstallation von LILO
`-v`	LILO liefert ausführliche Informationen über seine Arbeit

Tabelle 12.2: Optionen des Befehls `lilo`

Um LILO ganz normal zu installieren reicht ein Aufruf ohne Parameter. Es können aber verschiedenste Optionen mit übergeben werden. Eine Übersicht über diese Schalter liefert Tabelle 12.2.

Meistens werden Sie aber eine Konfigurationsdatei bei Ihren Aufrufen von LILO verwenden. Bei der Verwendung einer Konfigurationsdatei müssen Sie Schlüsselworte anstatt Optionen verwenden. Die Tabelle 12.3 zeigt die Optionen mit ihren dazugehörigen Schlüsselworten.

12.3.2 /etc/lilo.conf

Die Standardkonfigurationsdatei für `lilo` ist `/etc/lilo.conf`. Hier ein Beispiel für die Konfigurationsdatei.

```
# LILO Konfigurations-Datei
boot=/dev/hda
vga=normal
read-only
prompt
timeout=100
# End LILO global Section
#
other = /dev/hda1
  label = win
  table = /dev/hda
#
image = /boot/vmlinuz
  root = /dev/hda3
  label = linux
#
image = /boot/vmlinuz.old
  root = /dev/hda3
  label = oldlinux
```

Option	Schlüsselwort
`-b BOOTDEV`	`boot=BOOTDEV`
`-c`	`compact`
`-D KERNEL`	`default=KERNEL`
`-d ZENTELSEK`	`delay=ZEHNTELSEK`
`-f DATEINAME`	`disktab=DATEINAME`
`-i DATEINAME`	`install=DATEINAME`
`-l`	`linear`
`-m DATEINAME`	`map=DATEINAME`
`-P fix`	`fix-table`
`-P ignore`	`ignore-table`
`-S DATEINAME`	`force-backup=DATEINAME`
`-s DATEINAME`	`backup=DATEINAME`
`-v`	`verbose=LEVEL`

Tabelle 12.3: Optionen und Schlüsselworte für LILO

```
password=hamster
```

Es können Parameter wie `append`, `ramdisk`, `read-only`, `read-write`, `root` und `vga` an den Kernel weitergegeben werden. Dies kann global oder auch in den einzelnen Image-Sektionen passieren. So kann z. B. für die Fehlerbehebung ein Kernel in eine RAM-Disk geladen werden. Genauso können Sie Ihr System für den Bootvorgang als Nur-Lesend definieren.

Wenn Sie die Konfigurationsdatei geändert haben, ist es nötig das Kommando `/sbin/lilo` auszuführen, damit die Änderungen auch in den LILO übernommen werden. Führen Sie also das Kommando nach jeder Änderung der Konfigurationsdatei aus. Auch bei Änderungen der Systemkonfiguration, wie z. B. Partitionswechsel und Kernelkompilierung muß der Befehl ausgeführt werden, damit LILO sauber weiter arbeiten kann.

Parameter

boot Der Parameter `boot` legt fest, auf welchem Datenmedium sich der Bootsektor befindet. Für PCs mit IDE-Festplatten können das Geräte wie z. B. `/dev/hda`, `/dev/hdb` etc. sein.

read-only Durch das Setzen dieses Parameters wird das Wurzeldateisystem als "nur lesbar" gemountet. Normalerweise wird es dann durch die weitere Startprozedur als "schreibbar" wiedergemountet.

prompt Wird dieser Parameter gesetzt, so wird dem Benutzer beim Start ein Bootmenü angeboten. Fehlt der Parameter, so kann der Benutzer durch Drücken der <Shift>-, <Strg>- oder <Alt>-Taste trotzdem ins Bootmenü gelangen.

timeout Der Parameter `timeout` setzt die Zeit in Zehntelsekunden fest, in der der Benutzer sein System auswählen kann, bevor das System mit der Standardeinstellung gestartet wird. Ist der Parameter `prompt` gesetzt, muß `timeout` festgelegt werden um ein automatisches Starten zu ermöglichen.

map Der Parameter muß nur gesetzt werden, wenn eine andere Map-Datei als die vom System voreingestellte `/boot/map` verwendet werden soll.

install Dieser Parameter legt die Datei fest, die als Bootsektor installiert werden soll. Voreingestellt ist `/boot/boot.b`.

image Dieser Parameter legt zusammen mit `label` und `root` die Kernel fest, die im Bootmenü angeboten werden sollen. Für jeden Kernel bzw. Betriebssystem wird ein Eintrag benötigt.

label Dieser optionale Parameter legt den Text fest, unter dem der Kernel im Auswahlmenü erscheint.

root Mit diesem Parameter, der stets hinter `image` kommt, wird das Wurzelverzeichnis für das Betriebssystem bzw. die Kernelvariante festgelegt.

other Dieser Parameter entspricht `image`, wird aber für andere Betriebssystem wie z. B. Windows verwendet.

password Es ist manchmal sinnvoll den Zugriff auf einzelne Images zu beschränken. Dies geschieht über die Option `password`. Hiermit können Sie für jedes Image ein individuelles Passwort festlegen. Falls Sie das machen, achten Sie auf jeden Fall darauf, daß die Datei `/etc/lilo.conf` nur für den Benutzer *root* lesbar und schreibbar ist.

mandatory Wird diese Option gesetzt, gilt das Passwort für dieses Image auch für alle anderen Images.

restricted Das Passwort wird nur abgefragt, wenn der Kernel mit zusätzlichen Parametern gestartet wird.

12.3.3 Sicherheit: Zugriff aufs System ohne Passwort

Was macht man, wenn man das Passwort für *root* vergessen hat. Wer schlau war hat sich inzwischen einen zweiten Administrator angelegt (8.3.6). Wenn nicht, kann man über Disketten bzw. CD ein Rettungssystem starten. Dieses ist ein eigenständiges Linux-System. Sie brauchen dann nur die Wurzelpartition (/) mounten und können nun entweder das Passwort in der Datei `/etc/shadow` ändern oder in der Datei `/etc/passwd` einem anderen Benutzer, von dem Sie dann hoffentlich noch das Passwort kennen, die UID 0 zuweisen.

Es geht auch ohne Rettungssystem. Wenn Sie beim Starten des Image den Parameter `init=/bin/bash` an den Imagenamen anhängen, startet der Kernel gleich eine Bash.

```
LILO: linux init=/bin/bash
```

Das Wurzelsystem ist aber nur als Read-Only eingebunden. Es muß daher noch als lesend und schreibend wiedergemountet werden.

```
(none):/# mount -t ext3 -o rw,remount /dev/hda2 /
EXT3 FS 2.4-0.9.17, 10 Jan 2002 on ide0(3,2), internal journal
```

Und schon können die Benutzerdateien von Hand verändert werden. Es gibt nur zwei Nachteile am System. Der Pfad ist etwas sehr dürftig und die englische Tastatur ist eingeschaltet.

Da es so einfach ist das Passwort von *root* zu ändern oder sich einen anderen Benutzer zu *root* zu machen, muß ich mein System schützen. Denken Sie daran, jedes System, an das jemand pysikalisch herankommt, ist knackbar. Im einfachsten Fall baut er nur die Festplatte aus.

Das Booten eines anderen Systems über externe Medien (Floppy oder CD) verhindern Sie dadurch, daß in der Bootreihenfolge die Festplatte an erster Stelle kommt. Damit nicht jeder diese Reihenfolge umstellen kann, schützen Sie dann das BIOS noch mit einem Passwort.

Die Möglichkeit über LILO und `init` können Sie verhindern, in dem Sie alle Linux-Images mit einem Passwort schützen. Zusätzlich geben Sie noch den Parameter `restricted` dazu. Dann wird das Passwort nur abgefragt, wenn manuell Parameter beim Booten mit angegeben werden. Ohne Parameter startet das System wie sonst auch.

12.3.4 Fehlererkennung

Wenn LILO geladen wird, zeigt er Buchstabe für Buchstaben seinen Namen an. Dabei steht jeder Buchstabe für einen bestimmten Bereich, der abgearbeitet wird. Stürzt LILO beim Laufen ab, so kann man anhand der dargestellten Zeichen erkennen, an welcher Stelle ein Problem auftrat.

nichts - LILO wurde nicht geladen. Er ist entweder nicht installiert oder die Partition ist nicht aktiv.

L FEHLERCODE - Wahrscheinlich ein Medienfehler.

LI - Der erste Teil wurde erfolgreich beendet, aber der zweite Teil wurde nicht ausgeführt. Das kann einem Plattengeometriefehler liegen oder die Datei `/boot/boot.b` wurde nicht gefunden.

LIL - Der erste und zweite Teil wurde erfolgreich beendet. Die Map-Datei kann nicht geladen werden. Wahrscheinlich ein Medienfehler.

LIL? - Der Loader für den zweiten Teil wurde an eine falsche Adresse geladen. Wahrscheinlich Geometriefehler oder die Datei `/boot/boot.b` konnte nicht gefunden werden.

LIL- - Beschädigte Beschreibungstabelle. Wahrscheinlich ein Geometriefehler oder die Map-Datei konnte nicht gefunden werden.

LILO - LILO wurde erfolgreich geladen.

Es kann sein, daß Sie hexadezimale Zahle nach dem ersten 'L' sehen. Diese werden zufällig erzeugt und weisen indirekt auf eine temporäres Plattenproblem hin. Wenn LILO lädt, ist das Problem nicht behoben. Sie sollten so bald wie möglich ein `fsck` durchführen.

12.4 Linux beenden

Sie sollten **niemals** Ihr Linux-System einfach abschalten. Es ist üblich, daß der Kernel wichtige Daten wie z. B. die Inode-Tabelle oder frisch geschriebene Daten im RAM vorhält. Diese Daten werden erst dann syncronisiert, wenn der Kernel dafür Zeit hat oder Platz benötigt wird. Wenn Sie nun einfach den Rechner ausschalten, hat der Kernel keine Zeit mehr die Festplatte mit den im RAM gespeicherten Daten zu synchronisieren. Es kann daher zu Datenverlusten kommen bis zu erheblichen Fehlern im Dateisystem. Vor allem kann es zu schwerwiegenden Fehlern kommen, wenn Prozesse gerade aktiv schreibend auf die Festplatten zugreifen.

12.4.1 shutdown

Um das laufende System zu beenden wird der Befehl `shutdown` eingesetzt.

```
shutdown [OPTIONEN] WANN [NACHRICHT]
```

Der `shutdown`-Befehl beendet alle Prozesse in dem er das SIGTERM-Signal an sie sendet. Als nächstes wird `init` aufgerufen um den Runlevel zu ändern und die Dateisystem zu unmounten.

Der Shutdown-Vorgang kann zu einem bestimmten Zeitpunkt ausgelöst werden. Mit dem Stichwort `now` wird der Vorgang sofort eingeleitet. Die Angabe der Minuten als eine positiven Zahl wie z. B. `+15` löst den Vorgang nach der angegebenen Zeit aus. Es kann aber auch in der Form `hh:mm:ss` ein fester Zeitpunkt vorgegeben werden.

Wahlweise kann eine spezielle Nachricht an alle eingeloggten Benutzer gesendet werden. Wird keine Nachricht angegeben, meldet sich das System bei den Benutzern mit einer Standardnachricht.

Optionen	
`-c`	Beendet einen laufenden Shutdown-Vorgang
`-f`	`fsck` wird nach einem Neustart nicht ausgelöst
`-F`	`fsck` wird nach einem Neustart ausgelöst
`-h`	Anhalten des Systems nach dem Herunterfahren; Runlevel 0
`-k`	Sendet nur die Warnung, fährt das System aber nicht herunter
`-n`	Fährt herunter ohne `init` aufzurufen
`-r`	Führt einen Neustart nach dem Herunterfahren aus; Runlevel 6
`-t SEKUNDEN`	Zeit zwischen dem "Töten" der Prozesse und dem Aufruf von `init`

Nur *root* hat das Recht den Rechner mit dem Befehl `shutdown` herunterzufahren. Wenn die Datei `/etc/shutdown.allow` existiert und der Benutzer befindet sich in der darin enthaltenen Liste, dann kann auch er den Befehl (zusammen mit dem Schalter `-a`) nutzen.

Denken Sie daran: Schalten Sie nie den Rechner aus, wenn der Shutdown-Vorgang noch nicht beendet worden ist. Der Vorgang ist beendet, wenn der Text
`The system is halted`
oder
`Runlevel 0 is reached`
erschien ist.

12.4.2 halt und reboot

Neben `shutdown` gibt es noch zwei weitere Befehle um den Rechner herunterzufahren. Dies sind die Befehle `halt` und `reboot`. Ihre Funktion bedarf wohl keiner weiteren Erklärung.

```
halt [OPTIONEN]
reboot [OPTIONEN]
```

Wenn Sie die Befehle `halt` oder `reboot` aufrufen, wird erst der aktuelle Runlevel überprüft. Ist das System im Runlevel 0 oder 6, dann wird der Befehl ausgeführt. Ansonsten wird das Kommando `shutdown -nf` ausgeführt.

Weder `halt` noch `reboot` senden einen Warnhinweis an die eingeloggten Benutzer noch lassen Sie eine Verzögerung bis zur Ausführung zu.

Optionen	
`-d`	Es wird nicht in die Datei `/var/log/wtmp` geschrieben
`-f`	Der Befehl `shutdown` wird nicht aufgerufen aber ein halt oder reboot versucht
`-i`	Das Netzwerkinterface wird vor dem Herunterfahren deaktiviert
`-n`	Der Befehl `sync` wird nicht vor halt oder reboot ausgeführt
`-p`	Schaltet die Energie nach dem Herunterfahren aus
`-w`	Fährt das System nicht herunter, sondern schreibt nur in die Datei `var/log/wtmp`

12.5 Prozeßverwaltung

Jede Anfrage, die an die Shell gesendet wird, wird als Prozeß bezeichnet. Da nun Linux ein Multiuser- und Multitasking-Betriebssystem ist, laufen ständig eine ganze Reihe von Prozessen, z. B. für jeden eingeloggten Benutzer die Login-Shell (meist die `bash`) oder Prozesse, die die seriellen Schnittstellen und das Netzwerk dahingehend überprüfen, ob ein Benutzer einloggen will.

Die Verwaltung all dieser Prozesse ist eine der Hauptaufgaben des Systems. Zur Verwaltung werden die sogenannten Prozeßkenndaten verwandt. Die wichtigsten sind:

- **PID** - die Prozeßnummer, über die eine Interprozeßkommunikation erfolgen kann. Das bekanntest Beispiel hierfür ist das Versenden von Signalen. Das Signal 9 etwa tötet einen Prozeß!
- **PPID** - Prozeßnummer des Elternprozesses (Parent PID). Jeder Prozeß, bis auf den ersten (besser sollte man sagen den 0.), stammt von einem anderen ab.
- **Prozeßzustand** - kann z. B. sein *Running* (wird gerade von der CPU bedient), *Waiting* (wartet auf ein Ereignis, z. B. eine Eingabe), *Suspended* (ist bereit, hat aber nicht die CPU zur Verfügung) und *Zombie* (ist eigentlich beendet).
- **Priorität** - Die Nice-Priorität ist eine Zahl, die dem Prozeß beim Start mitgegeben wird. Diese, sein Zustand, die Zeit, die er auf Bedienung durch die CPU gewartet hat gehen ein in die Berechnung der aktuellen Priorität, die letztlich bestimmt, wann der Prozeß wieder CPU-Zeit erhält. Diese Berechnung erfolgen periodisch.
- **UID, GID** - Benutzer- und Gruppennummer, um den Benutzer zu identifizieren. UID und GID gehen normalerweise ein in die:
- **effektive UID, GID** - über die die Rechte des Benutzers kalkuliert werden. Falls die Programmdatei mit dem s-Recht versehen ist, wird die effektive UID oder GID die UID oder GID des Besitzers der Programmdatei. Z. B. ist der Besitzer der Programmdatei `/etc/passwd` der Benutzer *root*. Startet ein gewöhnlicher Benutzer dieses Programm erhält dieser Prozeß die effektive UID 0, er wird für den Programmlauf zum Superuser!
- **kontrollierendes Terminal** - Terminal von dem aus der Prozeß gestartet wurde. Diese Angabe ist insofern wichtig, da der Prozeß seine Ausgaben/Eingaben über diesen Terminal abwickelt! Eine Reihe von Prozessen wird gewöhnlich nicht von einem Terminal aus gestartet (sondern häufig automatisch

beim Systemstart). Solche Prozesse ohne Terminal nennt man Dämonprozesse. Sie verrichten in der Regel wichtige Systemdienste. Bekannte Dämonen sind lpd (Druckerdämon), crond (Zeitdienstedämon) oder inetd (für das Netzwerk zuständig).

12.6 Prozeßüberwachung

Um sich eine Übersicht über die Prozesse zu verschaffen und übelgelaunte Prozesse, die das System belasten, zu identifizieren, können eine Reihe von Tools verwendet werden.

12.6.1 ps

Um sich eine Liste der Prozesse und deren PIDs anzeigen zu lassen, kann man den Befehl `ps` verwenden.

`ps [OPTIONEN]`

Die Optionen bei `ps` werden ohne führenden Bindestrich geschrieben, weil der Befehl keine Parameter kennt.

Der Befehl `ps` zeigt normalerweise die unter der eigenen UID laufenden Prozesse an. Die Option `-e` bewirkt eine Anzeige aller laufenden Prozesse.

Optionen	
`a`	Zeigt alle Prozesse auf einem Terminal an
`e`	Zeigt die dazugehörigen Umgebungsvariablen
`f`	Zeigt die Ausgabe als Baum (Eltern- und Kindsprozesse)
`l`	Ausführliche Ansicht
`u`	Zeigt Benutzer und Startzeit an
`x`	Zeigt die Prozesse mit einem zugeordneten Terminal
`-e`	Zeigt alle Prozesse an

Beispiel

```
tapico@defiant:~ > ps
  PID TTY          TIME CMD
  189 tty1     00:00:00 bash
  196 tty1     00:00:00 startx
  197 tty1     00:00:00 tee
  206 tty1     00:00:00 xinit
  210 tty1     00:00:00 kwm
  280 tty1     00:00:00 kde <defunct>
  300 tty1     00:00:01 kfm
  301 tty1     00:00:00 krootwm
  305 tty1     00:00:00 kbgndwm
  308 tty1     00:00:01 kpanel
  316 tty1     00:00:05 nedit
  569 tty1     00:00:00 kvt
  583 pts/0    00:00:00 bash
  631 pts/0    00:00:00 ps
tapico@defiant:~ > ps f
  PID TTY      STAT   TIME COMMAND
  189 tty1     S      0:00 -bash
  196 tty1     S      0:00 sh /usr/X11R6/bin/startx
  206 tty1     S      0:00  \_ xinit /home/tapico/.xinitrc --
  210 tty1     S      0:00      \_ kwm
  280 tty1     Z      0:00          \_ [kde <defunct>]
  197 tty1     S      0:00 tee /home/tapico/.X.err
  308 tty1     S      0:01 kpanel
  305 tty1     S      0:00 kbgndwm
```

```
  301 tty1     S      0:00 krootwm
  300 tty1     S      0:01 kfm
  316 tty1     S      0:06 /usr/X11R6/bin/nedit
  569 tty1     S      0:00 kvt -T ption Terminal -icon kvt.xpm -miniicon kvt.xpm
  583 pts/0    S      0:00  \_ bash
  680 pts/0    R      0:00      \_ ps f
```

12.6.2 pstree

Das Kommando `pstree` gibt einen Baum von allen Prozessen aus. Dadurch wird deutlich, welcher Prozeß von welchem anderen Prozeß gestartet wurde.

```
pstree [OPTIONEN] [PID]
```

Optionen	
`-a`	Zeigt die Kommandozeilenargumente an
`-c`	Verhindert die kompakte Darstellung von identischen Zweigen
`-l`	Zeigt lange Zeilen an, sonst werden die Zeilen für die Bildschirmbreite umgebrochen
`-n`	Sortiert die Prozesse nach PID und nicht nach Name
`-p`	Zeigt die PIDs zu den Prozessen an
`-u`	Zeigt an, wenn die UID von Kinds- und Elternprozeß sich unterscheidet

Ohne weitere Parameter gibt `pstree` den Baum ab dem Prozeß `init` aus. Durch Eingabe einer PID kann die Ausgabe auf einen Teilast beschränkt werden.

```
tapico@defiant:~> pstree -p 1079
kdeinit(1079)-+-kdeinit(1088)
              |-kdeinit(1102)-+-bash(1103)-+-kdvi(1254)---kviewshell(1255)
              |               |            `-pstree(2415)
              |               `-bash(1186)---su(1193)---bash(1194)
              `-nedit(1207)
```

12.6.3 top

Ein anderer Weg die laufenden Prozesse zu betrachten ist der Befehl `top`. Er erlaubt daneben auch gleichzeitig Statistiken über den Speicher und die Swap-Datei. Systemlaufzeit, CPU-Status und Prozeßgröße sind weitere Angaben, die das Tool liefert. Im Gegensatz zu `ps` gibt `top` nicht nur einen Schnappschuß der Prozesse wieder, sondern erlaubt eine kontinuierliche Beobachtung.

Hier ein Beispiel für eine Ausgabe des Befehls `top`.

```
  8:38pm  up 29 min,  3 users,  load average: 0.11, 0.08, 0.09
59 processes: 51 sleeping, 2 running, 6 zombie, 0 stopped
CPU states:  0.7% user,  0.7% system,  0.0% nice, 98.4% idle
Mem:    62844K av,  61308K used,    1536K free,   46324K shrd,   15880K buff
Swap:  128516K av,     380K used,  128136K free                  17252K cached

  PID USER     PRI  NI  SIZE  RSS SHARE STAT  LIB %CPU %MEM   TIME COMMAND
  724 tapico     9   0  1060 1060   876 R       0  0.9  1.6   0:00 top
  207 root      10   0 11160  10M  2008 R       0  0.3 17.7   0:16 X
  569 tapico     2   0  4272 4272  3100 S       0  0.1  6.7   0:01 kvt
    1 root       0   0   200  200   172 S       0  0.0  0.3   0:05 init
    2 root       0   0     0    0     0 SW      0  0.0  0.0   0:00 kflushd
    3 root       0   0     0    0     0 SW      0  0.0  0.0   0:00 kupdate
    4 root       0   0     0    0     0 SW      0  0.0  0.0   0:00 kpiod
    5 root       0   0     0    0     0 SW      0  0.0  0.0   0:00 kswapd
   63 bin        0   0   396  396   320 S       0  0.0  0.6   0:00 portmap
```

```
   67 root       0   0   480  480   400 S       0  0.0  0.7   0:00 rpc.ugidd
   75 root       0   0   556  556   468 S       0  0.0  0.8   0:00 syslogd
   79 root       0   0   784  784   328 S       0  0.0  1.2   0:00 klogd
  105 root       0   0   508  492   396 S       0  0.0  0.7   0:00 rpc.mountd
  108 root       0   0   500  484   396 S       0  0.0  0.7   0:00 rpc.nfsd
  115 root       0   0  1536 1516  1400 S       0  0.0  2.4   0:00 httpd
  116 wwwrun     0   0  1056  912   800 S       0  0.0  1.4   0:00 httpd
  122 at         0   0   540  540   460 S       0  0.0  0.8   0:00 atd
```

Top besitzt eine Reihe von interaktiven Kommandos. Mit der Taste <h> oder <?> wird ein Hilfe-Bildschirm angezeigt.

```
Proc-Top Revision 1.2
Secure mode off; cumulative mode off; noidle mode off

Interactive commands are:

space   Update display
^L      Redraw the screen
fF      add and remove fields
oO      Change order of displayed fields
h or ?  Print this list
S       Toggle cumulative mode
i       Toggle display of idle proceses
c       Toggle display of command name/line
l       Toggle display of load average
m       Toggle display of memory information
t       Toggle display of summary information
k       Kill a task (with any signal)
r       Renice a task
N       Sort by pid (Numerically)
A       Sort by age
P       Sort by CPU usage
M       Sort by resident memory usage
T       Sort by time / cumulative time
u       Show only a specific user
n or #  Set the number of process to show
s       Set the delay in seconds between updates
W       Write configuration file ~/.toprc
q       Quit

Press any key to continue
```

Weil der Befehl `top` kontinuierlich seine Informationen auf den neuesten Stand bringt und damit den Bildschirm füllt, ist es ratsam den Befehl auf einer seperaten Konsole oder X-Terminal laufen zu lassen. Interessant ist `top` vor allen Dingen, weil er das interaktive "töten" von Prozessen erlaubt, sowie das Ändern von Prozeßprioritäten.

12.6.4 free

Der Befehl `free` zeigt eine Übersicht über die Auslastung des Arbeitsspeichers.

```
free [OPTIONEN]
```

Optionen

`-b`	Angabe in Byte (*byte*)
`-k`	Angabe in Kilobyte(*kB*)
`-m`	Angabe in Megabyte(*MB*)
`-o`	Kein Anzeigen der Puffer/Cache-Korrektur(*ohne*)
`-s ZEIT`	Kontinuierliche Anzeige im Abstand von ZEIT Sekunden(*switchtime*)
`-t`	Zeigt eine Zeile mit den Werten aller Speicher an(*total*)
`-V`	Versionsnummer()

Beispiel
In diesem Beispiel zeigt sich deutlich, daß die Programme nur die Hälfte des Speichers (Angaben in MByte) für sich beanspruchen und daß die andere Hälfte durch Puffer und Caches benutzt wird. Die Momentaufnahme entstand während der Kompilierung eines Kernels.

```
tapico@enterprise:~> free -mt
             total       used       free     shared    buffers     cached
Mem:           501        492          9          0         55        215
-/+ buffers/cache:        220        280
Swap:         1004         28        975
Total:        1505        521        984
```

12.7 Prozeßadministration

Prozesse können auf verschiedenste Arten und Weisen gesteuert werden. So können Prozesse mit unterschiedlicher Priorität laufen. Sie können auf der Shell direkt ausgeführt werden oder zur Stillarbeit im Hintergrund verdonnert werden. Laufenden Prozesse können mit Signalen gesteuert oder gekillt werden. Wobei es beim Killen noch deutliche Nuancen gibt.

Eine Reihe von Shell-Kommandos können zur Administration verwendet werden.

12.7.1 nice

Wie oft und schnell ein Programm CPU-Zeit bekommt hängt u. a. von seiner Prozeßpriorität ab. Dies Prozeßpriorität kann mit dem Befehl `nice` verringert werden. Nur der Superuser ist in der Lage einem Prozeß eine höhere Priorität zuzuordnen.

```
nice [OPTION] KOMMANDO
```

Das Kommando ist ein Befehl oder eine Kette von Befehlen, die mit der angegebenen Priorität ausgeführt werden können. Im Normalfall ist die Prozeßpriorität bei Verwendung von `nice` auf den Wert 10 festgelegt. Mit dem Schalter `-n ZAHL` kann eine Priorität zwischen -20 und 19 angegeben werden. Dabei bedeutet eine kleinere Zahl eine höhere Priorität und eine große Zahl eine niedrige Priorität. Nur der Superuser ist in der Lage einen negativen Wert, und damit eine höhere Priorität als normal, einzustellen.

Beispiel
`nice -n 19 find / -name urmel* -print > urmelliste.txt`
Dieser langwierige Prozeß bekommt eine sehr niedrige Ausführungspriorität zugewiesen. Er wird praktisch nur ausgeführt, wenn das System genug Zeit hat.

12.7.2 renice

Der Befehl `renice` erlaubt eine Änderung der Prozeßpriorität im Gegensatz zu `nice` (12.7.1) für laufende Programme.

```
renice PRIORITÄT PROZESS
```

Dabei kann der Zielprozeß durch Angabe der PID bestimmt werden. Mit den Schaltern `-u` und `-g` kann sich auf alle Prozesse einer Benutzers oder einer Gruppe bezogen werden und mit `-p` weitere PIDs angegeben werden.

Beispiel
Das Kommando
`renice +5 4711 -u wwwrun kingkong -p 42`
erniedrigt die Prozeßpriorität der Prozesse 42 und 4711, sowie aller Prozesse von den Benutzern *wwwrun* und *kingkong*. Normale Benutzer können nur auf die Priorität Ihrer eigenen Prozesse einwirken und dabei die Priorität, wie schon aus `nice` bekannt, verringern.

12.7.3 Vorder- und Hintergrundprozesse

Die Shell ist Kommandozeileninterpreter und Programmiersprache zugleich. Die Kommandos können einzeln (synchron) oder mehrere parallel nebeneinander (asynchron) ausgeführt werden. Wenn ein synchrones Kommando ausgeführt wird, wartet die Shell bis der Befehl abgearbeitet worden ist, bevor sie weitere Eingaben akzeptiert. Dies bezeichnet man auch als einen Prozess im *Vordergrund* laufen zu lassen. Asynchrone Kommandos laufen ab, während die Shell weitere Kommandos ausführt. Dieses Kommando läuft dann im *Hintergrund.*

Also immer wenn Sie einen Befehl eingeben, läuft dieser Befehl im Vordergrund. Die Shell wartet so lange, bis der Befehl abgearbeitet worden ist. Das kann z. B. beim `find`-Befehl etwas länger dauern. Um das Terminal dabei nicht zu blockieren, kann der Prozeß in den Hintergrund gelegt werden. Dabei sollte natürlich die Ausgabe des Befehls abgefangen und in eine Datei umgeleitet werden. Ein an die Zeile angefügtes kaufmännisches Und (`&`) sorgt für das Verschieben in den Hintergrund.
`find / -user tapico -exec rm -f {}; 2> /dev/null &`

Ein laufendes Programm können Sie unter der `bash` mit der Tastenkombination <Strg>+<z> stoppen und in den Hintergrund verschieben.

Ein Job ist eine Folge von einem oder mehreren Kommandos. Also immer wenn Sie Linux einen einzelnen oder mehrere miteinander verknüpfte Befehle geben, erstellen Sie einen Job. Die Shell ist in der Lage diese Jobs zu kontrollieren und weißt diesen Jobs daher Identifikationsnummern zu.

12.7.4 jobs

Um sich die Jobs anzeigen zu lassen, die sich im Moment im Hintergrund befinden, können Sie den Befehl `jobs` verwenden.

```
jobs [OPTIONEN] [JOB]
```

Optionen

`-r`	Zeigt laufende Jobs an
`-s`	Zeigt gestoppte Jobs an

12.7.5 fg

Um einen im Hintergrund laufenden Prozeß wieder in den Vordergrund zu holen, kann der Befehl `fg` verwendet werden.

```
fg [JOB]
```

Außerdem geht auch `%` oder `fg %`. Wenn mehrere Hintergrundjobs vorhanden sind, muß die Jobnummer oder der Name angegeben werden.

Beispiele
`fg`
holt den einzigen im Hintergrund arbeitenden Job in den Vordergrund.

`fg %4`
holt den Prozeß mit der Nummer 4 aus dem Hintergrund.

`fg %joe`
holt den Job “`joe`” aus dem Hintergrund hervor.

```
ole@defiant:~ > tail -f ~/.X.err &
[1] 976
ole@defiant:~ > Creating Harddisk icons...
Starting kcontrol -init...Done.
No sound device available. kaudioserver not started.
Starting krootwm...Starting kfm...Done.
Done.
No sound device available. kwmsound not started.
Starting kbgndwm...Done.
Starting kpanel...Done.
startkde: program khotkeys not found!
Max Entries = 23
fg %1
tail -f ~/.X.err

ole@defiant:~ >
```

Der Befehl `tail` wird erst in den Hintergrund geschoben, dann wieder in den Vordergrund gebracht und mit <Strg>+<c> abgebrochen.

12.7.6 bg

Wenn ein Prozeß mit <STRG>+<Z> gestoppt und in den Hintergrund geschoben worden ist, kann er mit dem Befehl `bg` im Hintergrund wieder gestartet werden.

`bg [JOB]`

Außerdem geht auch `bg %N` oder `fg %NAME`. Wenn mehrere Hintergrundjobs vorhanden sind, muß die Jobnummer oder der Name angegeben werden.

12.7.7 kill

Es ist möglich mit den Prozessen über Signale zu kommunizieren. Das Signal kann eine Unterbrechung, eine illegale Anweisung oder andere Bedingungen anweisen. Das Kommando `kill` kann zur Sendung solcher Signale verwendet werden. Meistens wird es zum “töten” eines Prozesses benutzt.

`kill [OPTIONS] ID`

Das Standardsignal ist das Beenden des Prozesses mit *SIGTERM*. Man spricht in diesem Fall auch von Terminieren. Nur der Superuser oder der Besitzer des Prozesses dürfen Signale senden. Die ID kann die PID, `%` (Das wäre der aktuelle Job), `%N` oder `%JOBNAME` sein.

Eine Übersicht über die Signal liefert Tabelle 12.4.

Beispiel
`kill -9 %joe`
fordert den Job “`joe`” zum Selbstmord auf.

`kill -HUP $(cat /var/run/httpd.pid)`
führt zum Neustart des Apache-Webservers.

Nr.	Langname	Kurzname	Bedeutung
1	SIGHUP	HUP	Hangup: Reinitialisierung des Prozesses
2	SIGINT	INT	Interrupt (wie <STRG>+<C>)
3	SIGQUIT	QUIT	Quit: Beenden
9	SIGKILL	KILL	Sofortiges Beenden des Prozesses (wird nicht ignoriert)
15	SIGTERM	TERM	Sofortiges Beenden des Prozesses (kann ignoriert werden)

Tabelle 12.4: Signale des Befehls `kill`

12.7.8 killall

Genau wie der Befehl `kill` (12.7.7) leitet der Befehl `killall` Signale an Prozesse weiter. Allerdings wird bei diesem Befehl nicht die PID sondern der Name des laufenden Programm angegeben. Alle Prozesse, die diesem Namen zugeordnet sind, werden dann beendet.

```
killall [OPTIONS] ID
```

Die sonstige Arbeitsweise ist identisch mit `kill`.

Notizen:

XV	Arbeitsblatt Bootvorgang und Prozesse	15.1

1 Beenden Sie das X-Window-System und arbeiten Sie nur mit den Terminals.

2 Ermitteln Sie die folgenden Daten aus den Bootmeldungen des Kernels.

1. Konsolentyp:
2. Prozessortyp:
3. Prozessorgeschwindigkeit:
4. PCI-Bios:
5. Swap-Partition:
6. Kernelversion:
7. Kernelerstellungsdatum:
8. Geräte:

3 Für welchen Zweck wird das Programm `fsck` verwendet?

4 Loggen Sie sich auf Terminal 1 als *root* ein.

5 Lassen Sie sich die Informationen über die Auslastung des Systems (CPU und Speicher) anzeigen.

6 Beenden Sie die Anzeige und löschen Sie den Bildschirm!

7 Suchen Sie im ganzen Dateisystem nach Dateien, die den Rechnernamen enthalten. Starten Sie den Befehl als Hintergrundprozeß und achten Sie auf Datenumleitung!

8 Lassen Sie sich die Prozeßnummer Ihrer Prozesse anzeigen! Finden Sie speziell die Prozeßnummer des Hintergrundprozesses heraus!

9 Killen Sie den Hintergrundprozeß!

10 Prüfen Sie, ob `grep` trotzdem Informationen in Dateien abgelegt hat!

11 Lassen Sie sich die Prozeßnummern aller Prozesse anzeigen! (Welche Prozeßnummer hat der init-Prozeß?)

12 Lassen Sie sich ausführliche Informationen zu allen laufenden Prozessen anzeigen! Welcher Terminal kontrolliert den init-Prozeß, von welchem Prozeß stammt der init-Prozeß und gibt es sonst irgendeinen Prozeß der nicht von init abstammt?

13 Starten Sie nun den Editor `vi`.

14 Stoppen Sie `vi` ohne das Programm zu beenden.

15 Starten Sie den Befehl aus Aufgabe 7 als Hintergrundprozeß. Merken Sie sich die Job-ID.

16 Starten Sie das X-Window-System (`startx`) ohne das die Konsole blockiert wird. Wechseln Sie, falls nötig, zur Konsole zurück (<Strg>+<Alt>+<F1>).

17 Beenden Sie den in Aufgabe 15 gestarteten Hintergrundprozeß.

18 Kehren Sie zu `vi` zurück und schreiben Sie die Job-ID aus Aufgabe 15 in den Text.

19 Stoppen Sie `vi` wieder.

20 Starten Sie nun den Editor `joe`.

21 Stoppen Sie `joe` ohne das Programm zu beenden. Welches Problem tritt auf?

XV	Arbeitsblatt Bootvorgang und Prozesse	15.2

22 Lassen Sie sich alle Jobs anzeigen.

23 Lassen Sie sich alle Prozesse in der Baumansicht anzeigen.

24 Kehren Sie zu `joe` zurück, holen Sie den Prozeß in den Vordergrund und beenden Sie das Programm.

25 Kehren Sie zu `vi` zurück und schreiben Sie den Satz: "Mit den Aufgaben ist jetzt Schluß." Speichern Sie den Text als `/root/blubber.txt` ab und beenden `vi`.

Notizen:

Kapitel 13

Administrative Aufgaben und Datensicherung

13.1 Arbeiten nach Fahrplan

Ein großer Teil der Arbeiten des Systemadministrators werden regelmäßig wiederholt oder müssen zu bestimmten Zeiten gestartet werden. So ist es sinnvoll, die tägliche Datensicherung erst in der Nacht zu starten um die normale Tagesarbeit nicht zu behindern. Oder die Daten der letzten Inventur sollen an die Zentrale geschickt werden. Da es sich um einige Megabyte handelt, wäre es sinnvoll die freien Leitungen in der Nacht zu nutzen. Aber warum sollte der Administrator dafür so lange in der Firma bleiben.

Für die automatische Verwaltung täglicher Aufgaben und einmaliger Aufgaben besitzt Linux die entsprechenden Ressourcen.

13.1.1 at

Wenn Sie einen Job einmal zu einer bestimmten Zeit starten wollen, benutzen Sie das Kommando `at`.

```
at [OPTIONEN] ZEIT
```

Verantwortlich für die Ausführung des Befehls zur richtigen Zeit ist der Dämon `atd`, der im Hintergrund läuft und wie ein Wecker zum voreingestellten Zeitpunkt die entsprechenden Maßnahmen einleitet.

Optionen

`-b`	Startet den Job, wenn das System nicht ausgelastet ist (*batch*)
`-d`	Löscht einen Job (*delete*)
`-f DATEI`	Liest den Job aus der angegebenen Datei (*file*)
`-l`	Zeigt alle Jobs des Benutzers an (*list*)
`-m`	Sendet nach dem Abschluß des Jobs eine eMail (*mail*)

Die Benutzung des `at`-Kommandos ist ganz einfach.

1. Geben Sie `at` zusammen mit dem Ausführungszeitpunkt an.
2. Geben Sie die Kommandos ein, die ausgeführt werden sollen.
3. Drücken Sie <Strg>+<d> um den Job abzuschicken.

```
tapico@defiant:~ > at +1 minutes
warning: commands will be executed using /bin/sh
at> who > ~/who.txt
at> <EOT>
job 9 at 2001-02-10 18:25
```

Bei der Zeitangabe stehen Ihnen mehrere Möglichkeiten zur Verfügung.

- Ein Zeitpunkt als Tag, Stunde und Minute.
- Ein Zeitpunkt von jetzt an gemessen.
- Der Wochentag als Zahl oder als Name.
- Die Verwendung von Worten wie `midnight`, `noon`, `teatime`, `today`, `tommorrow` und `now`.

Beispiele

```
at 15:30
at 18.03.01 15:30
at 03/18/01 15:30
at 031801 15:30
at +3 hours
at +5 minutes
at now
```

13.1.2 atq

Der Befehl `atq` zeigt die `at`-Jobs des aktuellen Benutzers an.

```
atq
```

Die Funktion des Befehls ist identisch zu `at -l`.

13.1.3 atrm

Der Befehl `atrm` löscht einen wartenden `at`-Job.

```
atrm JOBNUMMERN
```

Die Funktion des Befehls ist identisch zu `at -d JOBNUMMER`.

13.1.4 Jobverwaltung

Jobs benötigen wie jeder Prozess Systemressourcen. Wenn nun viele Benutzer komplexe Jobs laufen lassen, kann dies signifikante Einschränkungen der Systemperformance zur Folge haben.

Mit zwei Dateien ist es möglich festzulegen, wer überhaupt Jobs starten darf. Dies sind die Dateien `/etc/at.allow` und `/etc/at.deny` . Wenn also ein Benutzer einen Job anlegen will, dann prüft das System zuerst, ob der Benutzer dazu berechtigt ist.

Existiert die Datei `/etc/at.allow` nicht, dann wird in der Datei `/etc/at.deny` nachgeschaut, ob der Zugriff für den Benutzer verboten ist.

Existiert die Datei `/etc/at.allow` und der Benutzer ist dort aufgeführt, dann darf der Benutzer den Job anlegen. Die Datei `/etc/at.deny` wird nicht überprüft. Eine leere `/etc/at.deny` erlaubt jedem Benutzer das Anlegen eines Jobs. Existiert hingegen die Datei gar nicht, dann hindert das alle Benutzer am Anlegen von Jobs. Wenn beide Dateien nicht existieren, dann darf nur der Superuser Jobs anlegen.

Auf jeden Fall sollten Sie als Systemadministrator darauf achten, daß das System nicht durch zu viele Job verstopft wird. Benutzen Sie dazu die Kommandos `atq` und `atrm`.

Feld	Werte
`Minute`	0 - 59
`Stunde`	0 - 23
`Tag des Monats`	0 - 31
`Monat`	0 - 11 oder Namen
`Wochentag`	0 - 7 oder Namen (0 oder 7 ist Sonntag)

Tabelle 13.1: Erlaubte Werte für einen Crontab-Job

13.1.5 batch

Das Kommando `batch` bewirkt das Gleiche wie `at` mit dem Schalter `-b` und legt einen Job an.

`batch [OPTIONEN] [ZEIT]`

Im Gegensatz zu `at` hat der Job eine sehr niedrige Priorität. Der Befehl untersucht die virtuelle Datei `/proc/loadavg`. Wenn die durchschnittliche Belastung (*average load*) unter 1.5 sinkt, dann wird der Job ausgeführt. Sie haben außerdem eine geringere Priorität als Hintergrundjobs. Im Gegensatz aber zu diesen, werden Sie nicht beim Ausloggen beendet, sondern laufen weiter. Vom Ende des Jobs werden Sie dann per eMail benachrichtigt.

13.1.6 crontab

Das Kommando `at` erlaubt nur die zeitlich gesteuerte einmalige Ausführung eines Jobs. Allerdings ist es manchmal angebracht ein Job regelmäßig auszuführen. Dazu gehört z. B. die Datensicherung oder das Abgleichen von zwei Datenbanken. Für diese Aufgaben steht der Befehl `crontab` zur Verfügung.

`crontab [OPTIONEN] [DATEI]`

Um `crontab` zu nutzen, müssen Sie zuerst eine Textdatei mit irgendeinem Namen außer `crontab` anlegen. Dort tragen Sie eine Zeile mit sechs Feldern ein. Diese sechs Felder sind von links nach rechts: `Minute`, `Stunde`, `Tag des Monats`, `Monat`, `Wochentag` und `Kommando`. Die Werte für die Felder entnehme Sie bitte der Tabelle 13.1.

Ein Asterisk '`*`' in einem Feld repräsentiert jeden möglichen Wert für das Feld.

Optionen

`-e`	Erstellt oder bearbeitet die `crontab`-Datei des Benutzers
`-l`	Zeigt die `crontab`-Datei des Benutzers an
`-r`	Löscht die `crontab`-Datei des Benutzers
`-u BENUTZER`	Legt fest, mit welcher `crontab`-Datei gearbeitet werden soll (nur Superuser)

Das `crontab`-Kommando erstellt eine `crontab`-Datei an und speichert eine Kopie unter dem Benutzernamen in dem Verzeichnis `/usr/lib/crontab` ab.

Beispiele
Um täglich festzuhalten, wer um 10 Uhr morgens eingeloggt ist, kann folgender Cronjob angelegt werden.

`00 10 * * * who >> /var/log/benutzerliste.log`

Zeitbereich können mit einem Bindestrich '`-`' angegeben werden. Bei einer Liste von Werten werden die Einzelwerte durch Kommata voneinander getrennt. Ein Job, der jeden Dienstag und Donnerstag um 12 Uhr ausgeführt werden soll, kann dann so angegeben werden.

`00 12 * * 2,4 myjob`

13.1.7 Der Daemon crond

Die Aufgabe des Ausführens der persönlichen `crontab`-Dateien und der regelmäßigen Systemjobs wird vom Daemon `crond` erledigt. Für die Systemjobs konsultiert der Daemon die Datei `/etc/crontab`, die bei S.u.S.E.-

Linux so aussehen könnte.

```
SHELL=/bin/sh
PATH=/usr/bin:/usr/sbin:/sbin:/bin:/usr/lib/news/bin
MAILTO=root

#-* * * * *  root  test -x /usr/sbin/atrun && /usr/sbin/atrun
0 21 * * *   root  test -x /usr/sbin/faxqclean && /usr/sbin/faxqclean
5 22 * * *   root  test -x /usr/sbin/texpire && /usr/sbin/texpire

# check scripts in cron.hourly, cron.daily, cron.weekly and cron.monthly
#
-*/15 * * * *   root  test -x /usr/lib/cron/run-crons && /usr/lib/cron/run-crons
0 0  * * *     root  rm -f /var/cron/lastrun/cron.daily
0 0  * * 6     root  rm -f /var/cron/lastrun/cron.weekly
0 0  1 * *     root  rm -f /var/cron/lastrun/cron.monthly
```

Bei S.u.S.E. sorgt das Skript `/usr/lib/cron/run-crons` dafür, daß die Skripte in den Verzeichnissen `/etc/cron.daily`, `/etc/cron.weekly` und `/etc/cron.monthly` regelmäßig ausgeführt werden. Bei anderen Distributionen erfolgt der Aufruf der Skripte durch das Programm `run-parts`, das durch den Daemon aufgerufen wird.

Nach dieser Konfigurationsdatei schaut der Daemon `crond`, ob es persönliche Aufgaben gibt. Diese stehen unter dem Benutzernamen im Verzeichnis `/var/spool/cron` .

Damit ist also der Daemon `crond` für die Ausführung der regelmäßigem System- und Benutzerjobs zuständig. Er wird gestartet, wenn Linux bootet und wacht jede Minute auf um zu überprüfen ob ein Job gestartet werden soll.

Es ist natürlich klar, daß Benutzer nur ihre eigenen Jobs und der Superuser alle Jobs bearbeiten kann.

13.1.8 Verwalten von `cron`-Jobs

Wie auch beim Befehl `at` können zu viele komplexe `cron`-Jobs das System erheblich beeinträchtigen. Deshalb existiert auch für diese Jobart Konfigurationsdateien. Dies sind die Dateien `/etc/cron.allow` und `/etc/cron.deny` . Ihre Funktionsweise ist identisch zu den Zugriffsdateien für `at`.

13.2 Logdateien

Eine wichtige Infomationsquelle bei Fehlfunktionen des Betriebssystems sind die Logdateien. Systeminformationen werden in verschiedenen Logdateien mitgeschrieben. Diese Logdateien sind entweder speziell für eine Anwendung gedacht oder mehrere Anwendungen schreiben ihre Nachrichten in eine Datei.

13.2.1 Der Daemon `syslogd`

Der Daemon `syslogd` übernimmt die Aufgabe die Systemaktivitäten mitzuloggen. Er wird normalerweise durch einen Teil der `rc`-Konfigurationsdateien gestartet, wenn das System bootet.[1] Wenn der Daemon startet wird die Datei `/etc/syslog.conf` ausgelesen, in der die Optionen für die Ausführung des Daemons `syslogd` stehen.

```
syslogd
```

[1] Siehe dazu auch Kapitel 12 Bootvorgang.

Signal	Aktion
`SIGHUP`	Reinitializiert `syslogd` durch Stoppen des Daemons, Neulesen der Datei `/etc/syslog.conf` und dem erneuten Starten des Daemons.
`SIGTERM`	Beendet `syslogd`.
`SIGQUIT`	Beendet `syslogd`.
`SIGINT`	Beendet `syslogd`.
`SIGUSR1`	Wechselt zum Debugging-Modus, wenn mit `-d` gestartet.
`SIGALARM`	Setzt eine Markierungslinie

Tabelle 13.2: Signale und ihre Auswirkungen auf `syslogd`

Optionen

`-f DATEI`	Gibt alternative Konfigurationsdatei an (*file*)
`-h`	Veranlaßt `syslogd` dazu, Nachrichten von verbundenen Hosts weiterzuleiten (*hosts*)
`-l HOST`	Die angegebenen HOSTs (Liste mit Doppelpunkt getrennt) werden nur mit einfachem und nicht mit vollem Hostnamen mitgeloggt (*list of hosts*)
`-m INTERVALL`	Zeit zwischen zwei markierten Linien (Standard: 20 Minuten) (*markintervall*)
`-r`	Erlaubt den Empfang von Netzwerknachrichten (*remotenews*)

Der Daemon `syslogd` kann durch Signale gesteuert werden. Der Befehl `kill` (Abschnitt 12.7.7) kann mit den Signalen dazu verwendet werden, `syslogd` zu starten, stoppen, die Datei `/etc/syslog.conf` auf den neuesten Stand zu bringen u. s. w.

```
kill -SIGNAL $(cat /var/run/syslogd.pid)
```

Die Datei `/var/run/syslogd.pid` enthält die aktuelle PID des `syslogd`-Prozesses. Ein Übersicht über die Signal zeigt Tabelle 13.2.

13.2.2 /etc/syslog.conf

Die Datei `/etc/syslog.conf` ist die Konfigurationsdatei für den Daemon `syslogd`. Sie gibt an welche Informationen wohin geschrieben werden. Das folgende Beispiel ist der S.u.S.E.-Linux-Distribution entnommen.

```
# print most on tty10 and on the xconsole pipe
#
kern.warn;*.err;authpriv.none  /dev/tty10
kern.warn;*.err;authpriv.none |/dev/xconsole
*.emerg  *

# enable this, if you want that root is informed
# immediately, e.g. of logins
#*.alert  root

# all email-messages in one file
#
mail.* -/var/log/mail

# all news-messages
# these files are rotated and examined by "news.daily"
news.crit -/var/log/news/news.crit
news.err -/var/log/news/news.err
news.notice -/var/log/news/news.notice

# Warnings in one file
#
*.=warn;*.=err -/var/log/warn
*.crit  /var/log/warn
```

```
# save the rest in one file
#
*.*;mail.none;news.none -/var/log/messages
```

Dabei besteht jede Linie aus drei Angaben. Der **Nachrichtenquelle**, dem **Nachrichtentyp** und der **Logdatei**[2].

Die Nachrichtenquelle ist eines der folgenden Schlüsselworte: `auth`, `authpriv`, `cron`, `daemon`, `kern`, `lpr`, `mail`, `mark`, `news`, `security`, `syslog`, `user`, `uucp` und `local0` bis `local7`.

Der Nachrichtentyp bzw. die Nachrichtenpriorität wird in aufsteigender Reihenfolge durch die Schlüsselworte `debug`, `info`, `notice`, `warning`, `err`, `crit`, `alert` und `emerg` ausgedrückt.

Ein Minuszeichen vor dem Namen der Logdatei sorgt dafür, daß die Nachrichten nicht sofort auf Platten geschrieben werden, sondern, wie sonst auch üblich, im RAM gespeichert und erst bei der nächsten routinemäßigen Synchronisation von RAM und Platte geschrieben werden.

Weitere Informationen finden Sie unter `man syslog.conf`.

Beispiele
Der Stern steht für alle Nachrichtenquellen bzw. Nachrichtentypen. So loggt die folgende Zeile alle Nachrichten mit der Priorität `emerg` mit.

```
*.emerg                  root
```

Das Gleichheitszeichen sorgt dafür, daß die genannte Nachrichtenpriorität exklusiv in einer Datei mitgeloggt wird.

```
*.=crit                 /var/log/critical
```

Das Ausrufungszeichen '`!`' wird als Negationsoperator verwendet.

```
mail.*;mail.!=info      /var/log/maillog
```

Damit zum Beispiel überhaupt keine Mail-Nachrichten mitgeloggt werden, kann `mail.!*` oder `mail.none` gesetzt werden.

13.2.3 Verwaltung der Logdateien

Jede Nachricht wird als eine Zeile in eine Logdatei geschrieben. Datum, Quelle und natürlich die Nachricht selber werden in den Logdateien geschrieben. Welche Logdateien Sie verwenden, hängt von der Konfiguration Ihres Systems ab. Es gibt aber ein paar wichtige Systemlogdateien, die überall eigentlich vorhanden sein sollten.

/var/log/messages

Dies ist die Hauptlogdatei des Systems. Hier laufen die meisten Meldung auf. Dies hängt aber stark davon ab, wieviele weitere Logdateien eingerichtet worden sind.

/var/log/wtmp

Hier werden die Login-Zeiten und die Login-Dauer der Benutzer festgehalt. Das Kommando `last` (Abschnitt 8.7.6) greift auf diese Logdatei zurück.

/var/run/utmp

Diese Logdatei enthält die Informationen über die aktuell eingeloggten Benutzer. Auf diese Datei greifen die Befehle `finger`, `w` und `who` zurück.

[2]Denken Sie daran: Auch Terminals sind im Prinzip für Linux nur Dateien.

Kommando	Aktion
`compress`	Benutzt `gzip` zur Kompression alter Dateien
`copytruncate`	Kopiert das Log und kürzt dann das alte Log
`create`	Benutzt die angegebenen Rechte fürs neue Log; sind keine Rechte angegeben, werden die Rechte des alten Logs übernommen.
`daily`	Logs rotieren täglich
`delaycompress`	Kompression bei der nächsten Rotation
`errors MAILTO`	Sendet Fehler an die angegebene Mailadresse
`ifempty`	Rotiert auch leere Logs
`include DATEI`	Fügt die angegebene Datei in die Konfigurationsdatei ein
`mail MAILTO`	Sendet Logs zur angegebenen Mailadresse, wenn sie gelöscht werden.
`monthly`	Logs rotieren monatlich
`nocompress`	Alte Logs werden nicht komprimiert
`nocopytruncate`	Logs werden nicht kopiert und gekürzt
`nocreate`	Gegenteil von `create`
`nodelaycompress`	Alte Logs werden sofort komprimiert
`noolddir`	Alte Logs werden nicht in ein anderes Verzeichnis verschoben
`notifempty`	Leere Logs werden nicht komprimiert
`olddir VERZEICHNIS`	Alte Logs werden in das angegebene Verzeichnis verschoben.
`postrotate`	Startet Skript nach der Rotation
`prerotate`	Startet Skript vor der Rotation
`rotate N`	Gibt die Anzahl der alten Logs an, die archivert werden.
`size N`	Rotation, wenn das Log die angegebene Größe erreicht hat. (k für kB und M für MB)

Tabelle 13.3: Kommandos für die Konfigurationsdateien von `logrotate`.

/var/log/lastlog

In dieser Datei werden die Loginzeitpunkte der Benutzer gespeichert. Der Befehl `lastlog` verwendet diese Datei und ermöglicht so den Zeitpunkt des letzten Einloggens für jeden Benutzer zu sehen.

Vorsicht: Bei den Dateien `/var/log/wtmp`, `/var/run/utmp` und `/var/log/lastlog` handelt es sich um Binärdateien.

13.2.4 Rotation von Logdateien: `logrotate`

Die Natur von Logdateien ist es stetig zu wachsen und das über einen langen Zeitraum. Deswegen müssen diese Dateien wie Hecken regelmäßig zurückgeschnitten werden oder sie werden irgendwann das gesamte System überwuchern.

Für diesen Zweck gibt es den Befehl `logrotate`. Es sorgt dafür, daß ältere Daten aus den Logdateien entnommen und archiviert werden, alte Logdateien gelöscht und neue leere Logdateien erstellt werden.

```
logrotate [OPTIONEN] KONFIG_DATEI
```

Normalerweise ist die Konfigurationsdatei `/etc/logrotate.conf` , es können aber auch andere Dateien verwendet werden. Die Statusinformationen werden in der Datei `/var/lib/logrotate.status` gespeichert.

Die möglichen Kommandos entnehmen Sie bitte der Tabelle 13.3.

Eine Konfigurationsdatei kann globale und locale Optionen enthalten. *Globale* Optionen gelten für alle Logs, während *lokale* Optionen für ein bestimmtes Log gedacht sind.

Der Eintrag

```
# Globale Optionen
weekly
```

```
# Lokale Optionen
# für wtmp
/var/log/wtmp {
  monthly
}
```

bewirkt, daß alle Logs wöchentlich rotieren. Dies ist eine *globale* Option. Die explizite Angabe des Lognamens beim zweiten Eintrag bewirkt, daß die globale Option durch die *lokale* Option überschrieben wird und die Logdatei `wtmp` nur monatlich rotiert.

13.3 Datensicherung

Die Datensicherung eines Systems ist eine der wichtigsten Aufgaben eines Administrators. In vielen Fällen kann es sogar die einzige Aufgabe sein.

Eine Datensicherung macht man nur aus einem Grund. Man will in der Lage sein ein zerstörtes System in kürzester Zeit wieder zum Laufen zu bringen. Eine Datensicherung zu besitzen kann einen davor bewahren Tage oder sogar Wochen an der Wiederherstellung des Systems und der Daten zu sitzen.

13.3.1 Vorüberlegungen

Mehrere Faktoren spielen bei der Planung von Datensicherungsstrategien eine Rolle.

Kosten für die Ausfallzeit Jedes System, sogar Linux, steht zu manchen Zeiten nicht den Benutzer zur Verfügung. Dies kann daran liegen, daß eine neue Programmversion aufgespielt wird, Fehler gepatcht werden, Hardware ausgetauscht wird oder die Systemleistung verbessert wird.

Steht das System dem Benutzer nicht zur Verfügung, so müssen auch die Kosten für die Ausfallzeit der Mitarbeiter mit in die Kosten für das gesamte Projekt einkalkuliert werden. Da dies sehr teuer werden kann, sollte man diesen Posten bei der Abwägung für die Durchführung der Aufgabe genau unter die Lupe nehmen.

Kosten für die Sicherung Es gibt eine große Bandbreite von Datensicherungsstrategien. Das reicht von der Speicherung der Daten auf einer Diskette bis zur Bildung eines riesigen Clusters für kritische Dienste die 24 Stunden am Tag und 7 Tage die Woche angeboten werden müssen. Wie Sie leicht sehen sind mit den verschiedenen Strategien auch ganz unterschiedliche Kosten verbunden. Hier muß abgewägt werden, was der Ausfall kosten würde und wieviel die Datensicherung kostet.

Arbeitslast des System Die Rolle des Systems ist ein wichtiger Faktor für die benötigte Verfügbarkeit. Während ein Webserver 24 Stunden am Tag aktiv sein muß, reicht es bei den meisten Bürosystem aus, daß sie während der Bürozeiten laufen. Mehrkosten für aufwendigere Datensicherungsstrategien machen sich meist dann bezahlt, wenn das System eine wichtige Rolle einnimmt bzw. einen hohe Verfügbarkeitsstufe besitzt.

13.3.2 Backup-Strategien

Es gibt eine Vielzahl von Backup-Strategien aus denen Sie die Richtige für sich auswählen können. Das kann z. B. sein:

- Cluster
- Standby- oder Ausfallserver
- Duplizierter oder Backup-Server
- Sicherung und Wiederherstellung

Cluster Ein Cluster besteht aus mindestens zwei Rechnern, die gemeinsam den Zugriff auf einen zentralen Datenpool ermöglichen. Jeder der Server kann verschiedene Applikationen ausführen. Wenn nun ein Mitglied des Clusters ausfällt, können die anderen Mitglieder seine Aufgaben übernehmen. Von dieser Maßnahme bemerkt der Benutzer nichts. Wenn Sie eine Verfügbarkeiten von 24 Stunden am Tag benötigen, ist dies eine vernünftige Wahl. Neben den Hardwarekosten fallen vor allem hohe Kosten für das Management und Setup an.

Standby- oder Ausfall-Server Bei dieser Lösung ist neben dem eigentlichen Server ein identischer Server im Betrieb. Dieser steht im Gegensatz zum Cluster den Anwendern nicht zur Verfügung. Er ist allerdings ständig aktiv und gleicht seine Daten mit dem Hauptserver kontinuierlich ab. Fällt der Hauptserver nun aus, dann springt der Standby-Server ein um die Aufgaben zu erfüllen. Auch dieser Vorgang ist, da der Server die gleiche IP-Adresse und den gleichen Namen hat, für den Anwender nicht nachvollziehbar. Diese Lösung ist dank des einfacheren Managements günstiger als der Cluster, obwohl ein Teil der Hardware brach liegt. Trotzdem sind die Kosten noch so hoch, daß es sich nur für Server mit einer Verfügbarkeit von 24 Stunden am Tag lohnt.

Duplizierter oder Backup-Server Auch bei dieser Lösung existiert ein zweiter identischer Server. Dieser ist allerdings nicht im Betrieb. Fällt nun der Hauptserver aus, so muß der Administrator per Hand den Rechner auswechslen und die aktuellen Daten dem neuen Server zur Verfügung stellen. Das kann zum Beispiel durch den Tausch der Datenfestplatte oder durch den Anschluß an das Raid-Archiv geschehen. Bei Medienfehlern ist allerdings das Aufspielen des Backups nötig. Diese Lösung ist wiederum etwas günstiger, da kein kompliziertes Management des Systems nötig ist. Allerdings bedeutet dieses Konzept eine Ausfallzeit, falls der Hauptserver seinen Geist aufgibt.

Sicherung und Wiederherstellung Dies ist die häufigste und günstigste Methode der Sicherung. Selbst wenn Sie eine drei oberen Lösungen gewählt haben, müssen Sie dieses Verfahren zusätzlich anwenden. Es basiert auf dem einfachen Prinzip, daß die Daten einfach an einem zusätzlichen Ort gespeichert werden. Bei diesem Konzept können sich die Kosten in einem weiten Bereich bewegen. Von der Diskette bis zu einem riesigen Storagetower ragt die Bandbreite.

13.3.3 Planung

Als ersten Schritt bei der Planung der Datensicherung sollten Sie sich über vier Fragen klar werden.

1. Was soll gesichert werden?
2. Wie oft soll gesichert werden?
3. Wieviel Zeit steht für die Sicherung zur Verfügung?
4. Wohin soll gesichert werden?

Was soll gesichert werden?

Ob eine Datei gesichert werden soll oder nicht läßt sich ganz einfach feststellen. Vergleichen Sie einfach die Zeit, die Sie für das Rekonstruieren der Daten benötigen, und die Zeit, die Sie für die Datensicherung benötigen. In den meisten Fällen brauchen Sie für das Rekonstruieren länger.

Als erstes sollten Sie Ihre Daten unterteilen in jene, die sich ändern, und jene, die sich nicht ändern. Als Regel kann man davon ausgehen, daß die System- und Applikationsdateien sich nicht so häufig ändern. Dies passiert ja nur, wenn das System und die Applikationen auf den neuesten Stand gebracht werden. Daher ist es eine Verschwendung von Ressourcen, wenn man diese Daten täglich sichert. Es ist aber eine gute Idee, nach jedem Update ein Backup vom aktuellen System zu fahren. So ist eine schnelle Wiederherstellung des laufenden Systems in kürzester Zeit. Ein Installation mit womöglich noch nachträglich zu installierenden Patches und Treiber dauert wesentlich länger, als eine Sicherung wieder auf das System zu spielen.

Für die Datendateien, die sich naturgemäß häufiger ändern, sollten Sie einen regelmäßigen Turnus für die Datensicherung einführen. Da die wichtigsten Dateien sich täglich ändern, sollte auch das Backup täglich durchgeführt werden.

Haben Sie sich entschieden verschiedene Datengruppen in verschiedenen Intervallen zu sicher, so sollten Sie diese Dateien auch an unterschiedlichen Orten, am besten in unterschiedlichen Partitionen, speichern. Dies erleichtert die Datensicherung wesentlich.

So sollte sich das Heimatverzeichnis auf einer seperaten Partition befinden um die Sicherung der persönlichen Dateien der Benutzer wesentlich zu vereinfachen. Außerdem würde ein Schaden an der Systempartition keinen Schaden an den Daten verursachen und die Wiederherstellung des Systems wesentlich vereinfachen.

Wie oft soll gesichert werden?

Nachdem Sie sich nun entschieden haben, was Sie sichern wollen, und das kann das gesamte System sein, müssen Sie noch festlegen, wie oft die Daten gesichert werden. Es ist nicht notwendig jede Datei jeden Tag zu sichern. Sie können es aber tun.

Warum sollten Sie Systemressourcen, Platz und Arbeitskraft für ein tägliches Update Ihrer Systemdateien verschwenden, wenn sich diese alle paar Wochen nur ändern. Eigentlich brauchen Sie nur eine Sicherung, wenn Sie Veränderung am System, wie Patchen oder Aufspielen neuer Versionen, durchgeführt haben. Ein regelmäßiges Backup in größeren Abständen (z. B. monatlich) sollte aber durchgeführt werden, um sicher zu gehen, daß Sie eine funktionierende Kopie Ihres Systems besitzen.

Wenn es um die Häufigkeit eines Backups geht, dann fragen Sie sich doch einfach, welcher Schaden Ihrem Unternehmen entsteht, wenn die Daten verlorengehen, bevor eine Sicherung gefahren werden konnte. In den meisten Fällen reicht ein tägliches Backup aus. Bei sehr wichtigen sich häufigen ändernden Daten kann es sogar sinnvoll sein eine stündliche Sicherung zu fahren.

Wieviel Zeit steht für die Sicherung zur Verfügung?

Auf jeden Fall sollten Sie berücksichtigen wieviel Zeit für ein Backup zur Verfügung steht. Da ein Backup eine nicht geringe Menge an Systemressourcen für sich beansprucht, sollte man die Sicherung in eine Zeit niedriger Auslastung verlagern. So sollte eine Sicherung für eine Verwaltung am späten Abend beginnen und vor dem nächsten Morgen abgeschlossen sein. Der schlimmste Fall tritt ein, wenn die Dauer der Sicherung länger dauert als das Sicherungsintervall lang ist. Hier muß man dann die Sicherungsstrategie wechseln.

Bei Systemen, die rund um die Uhr ausgelastet sind, muß man Fenster der Inaktivität nutzen um immer wieder einzelne Dateien zu sichern.

Wohin soll gesichert werden?

Die Menge der Daten, die Zeit, die zum Sichern zur Verfügung steht, und die Dauer der Wiederherstellung bestimmen letztendlich die Auswahl des Speichermediums. Bandgeräte sind seit langer Zeit und auch heute noch die erste Wahl der Sicherung. Sie können große Mengen an Daten sicher speichern und sind dabei sehr kostengünstig. Gegen die ausgereiften Bandgeräte spricht die lange Dauer der Sicherung und die Schwierigkeit einzelne Dateien aus einem solchen Backup wieder extrahieren. Ein Bandgerät enthält schließlich nicht ein Dateisystem wie eine Festplatte. Für die Sicherung auf Band existieren viele Applikationen, die die Sicherung managen wie auch beim Wiederherstellen der Daten helfen. Dies ist dann fast so einfach wie ein Festplattezugriff, aber durch die sequentielle Speicherung der Daten kann es sehr lange dauern.

In der heutigen Zeit der billigen Festplatten, CD-Rs und optischen Laufwerken haben Sie ein große Zahl weiterer Medien für ihr Sicherung zur Verfügung.

Eine zweite Festplatte ist ein einfaches und schnelles Sicherungsmedium. Sollen aber mehrere Sicherungen über längere Zeit aufbewahrt werden, wird dieses Prinzip bald unerschwinglich teuer.

Die CD-R ist durch ihren günstigen Preis und ihre lange Haltbarkeit ein weitere gute Lösung. Allerdings macht ihr beschränkter Datenplatz bei größeren Backups Probleme. Daß sie nur einmal verwendbar ist, ist bei dem geringen Medienpreis zu verschmerzen.

Wollen Sie aber dennoch ein mehrfach beschreibbares Medium verwenden, dann sollten Sie sich mit den Magneto-Optischen-Medien beschäftigen. Sie stellen mehr Platz als CD-Rs zur Verfügung, der Zugriff auf die Daten ist flexibel wie bei einer Festplatte und sie sind haltbarer als Bänder.

13.3.4 Backuptypen

Beim Backup werden vier Typen unterschieden, die sich darin unterscheiden was gesichert und wie es wiederhergestellt wird.

- Kopie
- Volles Backup
- Partielles Backup
- Inkrementelles oder differenzielles Backup

Kopie

Die einfachste und am häufigsten angewandte Methode der Datensicherung ist das Kopieren von Dateien. Obwohl es die am häufigsten angewandte Methode ist, ist es eigentlich kein richtiges Backup.

Wenn man eine Datei an einen anderen Ort kopiert, ist man in der Lage die Datei wiederherzustellen, falls sie beschädigt worden ist. Meisten benutzt man ein wiederbeschreibbares Medium wie Floppy oder Zip-Disk für diese Aufgabe. Hauptanwendungsgebiet ist die Sicherung von Konfigurationsdateien. So kann man bei Fehlern beim Konfigurieren durch einfaches Zurückkopieren der Dateien den alten Zustand des Systems wieder herstellen.

Volles Backup

Ein volles Backup umfaßt, wie der Name schon sagt, jede einzelne Datei auf dem System. Man hat also ein Abbild des jetzigen Zustands des Systems. Da alle Dateien gesichert sind der administrative Aufwand für Sicherung und Wiederherstellung minimal. Natürlich hat dieses Backup auch seine Nachteil. Da das gesamte System gesichert werden muß, fällt natürlich eine Menge an Daten an. Daher dauert die Sicherung sehr lange und es ist eventuell sogar nötig während der Sicherung die Medien tauschen zu müssen. Das verhindert natürlich eine automatische Ausführung.

Partielles Backup

Beim partiellen Backup hingegen werden nur Teile der Daten gespeichert. Dadurch kann die Häufigkeit der Sicherung auf die verschiedenen Dateiarten angepaßt werden. Systemdateien werden nur nach Änderungen im System gespeichert, während die Datenpartition z. B. täglich gesichert wird. Es existiert also ein aktuelles Abbild des Systems, was aber zu verschiedenen Zeiten erstellt wurde. Daher werden dann bei der Sicherung weniger Daten übertragen. Die Sicherung ist damit schneller und man kommt mit weniger Medien aus, was dazu führt, daß der Vorgang auch automatisiert ausgeführt werden kann.

Inkrementelles oder differenzielles Backup

Das inkrementelle oder differentielle Backup reduziert noch weiter die zu sichernden Daten. Man beginnt mit einem vollen Backup (oder partiellem Backup) aller zu sichernden Daten. In den nächsten Sicherungen werden nur die Dateien gesichert, die sich seitdem geändert haben.

Beispiel Sie machen Freitag in der Nacht ein volles Backup. Damit sind alle Daten auf Ihrem Medium. Am Montag bis zum Donnerstag sichern Sie jetzt nur die Daten, die seit der letzten Sicherung sich geändert haben.

Kommt es nun an einem Mittwoch zu einem Systemausfall, dann müssen Sie zuerst das Backup vom Freitag – das ist das volle Backup – einspielen. Danach müssen Sie dann die Backups vom Montag und Dienstag einspielen um die Änderungen vom letzten vollen Backup wiederherzustellen.

13.3.5 Lagerung der Backups

Als Administrator wird man häufig gebeten eine Datei wiederherzustellen, die durch einen Fehler gelöscht worden ist. Allerding merken die meisten Leute diesen Fehler nicht gleich sondern erst nach ein paar Tagen. Es ist daher wichtig die Backups auch einige Zeit aufzubewahren. In Abhängigkeit von den Informationen und wie schnell sie sich ändern, sollten Sie die Lagerhaltung planen.

Beispiel Sie wollen die Sicherung acht Wochen lang aufbewahren. Sie benötigen daher für die acht Wochen auch die Bänder. Nach den acht Wochen können Sie die ältesten Bänder wieder benutzen. Daneben wollen Sie das monatliche Backup für ein Jahr aufheben.

1. Jeden Freitag führen Sie ein volles Backup durch. Das Medium können Sie nach acht Wochen wieder benutzen. Sie benötigen dafür diese Sicherung also acht Bänder.
2. Von Montag bis Donnerstag führen Sie ein differentielles Backup durch. Auch diese Bänder müssen acht Wochen aufbewahrt werden, da eine Datei ja am Montag erstellt werden und dann am Donnerstag ausversehen gelöscht kann. Sie brauchen hier vier mal acht Bänder also 32 Stück.
3. An jedem ersten Tag im Monat machen sie ein volles Backup und bewahren die Bandkassette ein Jahr lang auf. Das macht also noch einmal zwölf Bänder.

Summa sumarum brauchen Sie also 52 Bänder um Ihre Backupstrategie durchführen zu können. Vergessen Sie aber auf keinen Fall die Bänder auch sorgfältig zu beschriften, sonst nützt Ihnen auch das vollständigste Backup nichts.

Lagerplatz und Schutz

Was nützt Ihnen das schönste Backup, wenn Sie Ihre Bänder direkt neben den Server legen. Ein kleines Feuer im Serverraum und nicht nur Ihr Server ist hinüber, sondern auch Ihre Sicherung. Also aufgepaßt. Ihre Sicherung nützt Ihnen nur etwas, wenn Sie die Bänder sicher aufbewahren. Am besten schaffen Sie sich einen feuerfesten Tresor an, wenn Sie die Bänder in der Nähe des Servers aufbewahren. Sinnvoll ist es, einen Teil der Bänder auch außerhalb der Firma aufzubewahren. Denken Sie aber auch daran, daß jeder, der an die Bänder kommt, auch Ihre Daten lesen kann. Ein Tresor zu Hause oder ein Schließfach in Ihrer Bank sind sehr sinnvoll.

Protokoll

Auch nützt Ihnen eine Sicherung nicht, wenn Sie vergessen ein Protokoll über Ihre Sicherung zu führen. Zumindest die Kassetten sollte durch eine Beschriftung Auskunft geben, was auf Ihnen gespeichert sind. Sonst müssen Sie Band für Band durchgehen, was auf ihm drauf ist. Oder Sie haben die Sicherung auf CD-R gebrannt. Inzwischen ist der Stapel an CD-Rs schon fünf mal umgekippt und alles ist durcheinander.

Machen Sie sich einfach einen Zettel und notieren Sie bei jedem Backup darauf:

- Wann wurde das Backup gefahren?
- Welcher Medientyp wurde verwendet?
- Welcher Bezeichnung hat das Medium?
- Was wurde gesichert?
- Wo wird das Medium aufbewahrt?

13.4 Werkzeuge für die Sicherung

Es gibt eine große Anzahl von Programmen für die Datensicherung. Einige davon sind große mächtige kommerzielle Lösungen, andere sind klein und gehören zur Linux-Grundausstattung. Zwei davon sind `tar` und `cpio`.

13.4.1 tar

Das Programm `tar` (*tape archive*) wird dazu benutzt um mehrere Dateien zu einer Archivdatei zusammenzupacken. Dabei wird die Verzeichnisstruktur beibehalten. Obwohl `tar` entwickelt wurde um Daten auf Magnetbänder zu schreiben, kann ein solches `tar`-Archiv auf jedem beliebigem Medium gespeichert werden. Daneben kann `tar` die Archive bei der Erstellung auch gleichzeitig mit `gzip` (13.5.1) komprimieren.

```
tar [OPTIONEN] [TARARCHIV] [DATEILISTE]
```

In den Optionen bilden die Schalter eine besondere Gruppe. Es kann immer nur ein Schalter zur Zeit verwendet werden, während die anderen Optionen kombinierbar sind.

Schalter	
`A`	Hängt das ARCHIV2 and das Ende von ARCHIV1
`c`	Legt ein neues Archiv an
`d`	Vergleicht den Inhalt des Archivs mit anderen Dateien
`r`	Fügt die neuen Dateien an das Ende eines bestehenden Archivs an
`t`	Zeigt eine Liste aller Dateien im Archiv
`u`	Fügt nur neue oder veränderte Dateien zum Archiv hinzu
`x`	Extrahiert Dateien aus dem Archiv
Optionen	
`b`	Definiert die Blockgröße
`e`	Verhindert das Aufsplittern von Dateien über Archiv-Volumes
`f`	Name des Archivs mit Pfad oder Gerätename
`m`	Übernimmt nicht die Änderungszeit der Datei aus dem Archiv
`n`	Das Gerät ist kein Bandgerät
`p`	Die Originalrechte werden übernommen
`v`	Zeigt die Liste der Dateien an, die hinzugefügt werden
`w`	`tar` arbeitet interaktiv
`z`	Benutzt `gzip` zur Kompression
`F`	Am Ende jedes Mediums das angegeben Skript ausführen
`L`	Länges des Bandes in kByte
`M`	Das Archiv ist in mehrere Teile (Volumes) aufgeteilt.
`W`	Überprüft die Dateien, nachdem sie zum Archiv hinzugefügt worden sind.

Anlegen eines `tar`-Archivs

Erstellung eines Archivs (`archiv.tar`) über alle Dateien im aktuellen Arbeitsverzeichnis und dessen Unterverzeichnissen.

```
tar cf archiv.tar .
```

Als Ziel kann neben einer Datei auch ein Gerät angegeben werden. Der folgenden Befehl speichert das komplette Dateisystem auf dem Gerät `/dev/tape`.

```
tar cf /dev/tape /
```

Dabei wird der Inhalt des Magnetbandes überschrieben. Meistens müssen Sie bei einem Magnetband die Blockgröße mit angeben. Die Blockgröße definiert dabei die Menge an Daten (in Einheiten zu 512 Byte), die zur gleichen Zeit geschrieben werden können. Die Angabe erfolgt durch die Option `b`.

```
tar cfb /dev/tape 20 /
```

Wie Sie sehen, wird der Wert für `b` nicht wie üblich direkt hinter die Option geschrieben, sondern erst werden die Optionen aufgeführt und dann in der Reihenfolge die Werte.

Wenn Sie das Archiv auf mehrere Geräte (z. B. Disketten) aufteilen wollen, müssen Sie die Größe der Archiv-Teile (Volumes) angeben. Dies erfolgt durch den Schalter `M`. So speichert der folgende Befehl den Inhalt des Verzeichnis `/home` auf das Diskettengerät `/dev/fd0` mit einer Größe von 1440 kB.

```
tar cfML /dev/fd0 1440 /home
```

Wenn Sie ein bestehendes Archiv nur auf den neuesten Stand bringen wollen, ohne alle Dateien erneut hineinzupacken, dann benutzen Sie den Schalter `u`.

```
tar uf archiv.tar .
```

Sie können Dateien auch an ein bestehendes Archiv anhängen.

```
tar rf archiv.tar neueDatei
```

Solche Archive können sehr groß werden, wenn Sie z. B. Logdateien speichern. Diese können sehr gut gepackt werden. Mit der Option `z` wird `tar` angewiesen, das entstandene Archiv auch gleich mit `gzip` (Abschnitt 13.5.1) zu packen.

Um sich die Dateiliste des Archivs anzeigen zu lassen, kann man den folgenden Befehl verwenden.

```
tar tf archiv.tar
```

Entpacken eines `tar`-Archivs

Um eine Archiv zu entpacken (Schalter `x`) und dabei auch die Dateinamen zu sehen (Option `v`) können Sie den folgenden Befehl benutzen.

```
var xvf archiv.tar
```

Vor dem Entpacken, sollten Sie feststellen welche Verzeichnispfad gespeichert wurden. Dies können Sie mit dem Befehl

```
tar tvf archiv.tar
```

feststellen. Die Option `v` erzeugt eine ausführlichere Anzeige als `t` alleine.

Schauen wir uns doch mal folgendes Beispiel an:

Aus dem Wurzelverzeichnis haben Sie die Daten im Verzeichnis `/home` mit folgendem Befehl gespeichert.

```
tar cf home.tar home/*
```

Daher wurde das Verzeichnis `home` in jedem Pfad mit eingefügt. Sind Sie allerdings ins Verzeichnis `/home` gewechselt und haben den Befehl als

```
tar cf home.tar .
```

ausgeführt, so ist das Verzeichnis `home` nicht mehr Bestandteil des Dateipfads.

Bei einem Backup sollten Sie die Dateien immer in dem gleichen Verzeichnis extrahieren in dem Sie auch das Archiv erzeugt hatten.

Um einzelne Dateien aus einem Archiv zu entpacken, können Sie zum einen den interaktiven Modus (Option `w`) von `tar` wählen. Bei dem folgenden Befehl werden Sie bei jeder Datei gefragt, ob Sie die Datei entpacken wollen.

```
tar xvwf archiv.tar
```

Allerdings hat diese Methode einen schwerwiegenden Nachteil. Auf diesem Wege eine Datei aus einem 10.000 Dateien-Archiv zu entpacken ist doch etwas mühselig. Hier bietet sich einfach an, die gewünschte Datei einfach anzugeben.

```
tar xf archiv.tar meineDatei.txt
```

13.4.2 Was ist ein Tarball?

Diese Frage stellen sich viele Benutzer, die zum ersten Mal ein Programm aus dem Netz laden und nicht von der Distribution installieren wollen. Auf den meisten Seiten findet sich neben Bezeichnungen wie RPM- und GNU Debian-Paketen auch die Möglichkeit das Programm als "*tarball*" herunterzuladen.

Neben den schon fertig kompilierten Binärdateien erhalten Sie in der Open Source Gemeinde auch den Quellcode um ihr Programm selbst kompilieren und installieren zu können. Dabei wird der Quellcode nicht nur alleine vertrieben, sondern meist zusammen mit Konfigurationsdateien für die Installation und Dokumentationen zum Programm. Diese Dateien befinden sich meistens in einer Verzeichnisstruktur.

Solche Strukturen werden am besten in einem tar-Archiv gesichert. Damit das entstandene Archiv auch schnell heruntergeladen werden kann, wird es mit `gzip` (13.5.1) komprimiert.

```
barclay@enterprise:~> tar cf superduper_2.1.tar superduper/2.1
barclay@enterprise:~> gzip superduper_2.1.tar
```

Es ist dann eine Datei mit der Endung `tar.gz` entstanden. Ein solches komprimiertes Archiv wird als **Tarball** bezeichnet.

Beim Auspacken geht man den umgekehrten Weg. Erst entpacken und dann die Dateien aus dem Archiv extrahieren.

```
tapico@defiant:~> gunzip superduper_2.1.tar.gz
tapico@defiant:~> tar xf superduper_2.1.tar
```

Nun ist das Verzeichnis `/superduper/2.1` entstanden, daß alle Dateien enthält. Heute können Sie das Entpacken und Extrahieren in einem Schritt durchführen. Der Schalter `x` beim Befehl `tar` berücksichtigt die Kompression.

```
tapico@defiant:~> tar zxf superduper_2.1.tar.gz
```

13.4.3 cpio

Ein weiteres Werkzeug für Arbeit mit Archiven ist `cpio` (*copy in and out*). Sie können damit nicht nur aus mehreren Dateien ein Archiv machen und die Dateien aus einem solchen Archiv wieder extrahieren, sondern auch komplette Verzeichnisstrukturen an einen anderen Ort kopieren.

```
cpio [OPTIONEN]
```

Wie auch bei `tar` gibt es auch hier Schalter und Optionen.

Schalter	
`-o`	Legt ein neues Archiv an
`-i`	Extrahiert Dateien aus einem Archiv
`-p`	Kopiert komplette Verzeichnisstrukturen

Optionen

`-a`	Setzt die Zugriffszeit der Dateien nach dem Kopieren zurück.
`-d`	Erzeugt Verzeichnisse, wenn Sie benötigt werden (`-i` und `-p`)
`-E DATEI`	Name einer Datei mit zusätzlichen Mustern für die zu entpackenden Dateien
`-F ARCHIV`	Name des Archivs, was entpackt werden soll (`-i`)
`-m`	Erhält die Änderungszeit der Dateien (`-i`)
`-r`	Fragt nach einem neuen Namen für die Datei vorm Kopieren, wird kein Namen angegeben, wird die Datei nicht kopiert.
`-t`	Zeigt mit `-i` die Dateiliste des Archivs an
`-u`	Überschreibt existierende Dateien
`-v`	Zeigt die Dateinamen bei der Bearbeitung an

Im Gegensatz zu `tar` arbeitet `cpio` nicht direkt mit den Dateien sondern übernimmt nur die Aufgabe der Archivierung. Um alle Dateien und Verzeichnisse im Verzeichns `/home` zu sichern, müssen die Dateinamen mit `find` übergeben werden und die Ausgabe in eine Datei umgeleitet werden.

```
find /home | cpio -o > home.cpio
```

Um nun zu prüfen welche Dateien in dem Verzeichnis sind, können Sie folgenden Befehl benutzen.

```
cpio -itF home.cpio
```

Um das Archiv zu entpacken, benutzen Sie den Befehl

```
cpio -iF home.cpio
```

Eine einzelne Datei läßt sich genau wie bei `tar` aus einem Archiv extrahieren.

```
cpio -iF home.cpio meineDatei.txt
```

13.5 Kompression

Selbst im Zeitalter der heutigen riesigen Festplatten[3] ist nie genug Platz vorhanden. Wichtig wird die Menge der Daten, wenn es um die Übertragung durchs Internet geht. Hier sind die Datenraten noch immer viel zu klein. Da aber die meisten Dateien viel weniger Informationen erhalten, als Daten in ihnen steckt, wurden Algorithmen entwickelt um die Daten auf diese Information zu komprimieren. Dieser Abschnitt behandelt nun die unter Linux gängigen Programme zur Kompression.

13.5.1 gzip

Das Tool `gzip` komprimiert den Inhalt einer Datei und erzeugt daraus eine neue Datei mit dem gleichen Namen und einem angehängten `.gz`. Im Gegensatz zu dem Programm `zip`, daß es auch unter verschiedenen Namen unter Windows gibt, kann `gzip` nur eine Datei packen und löscht standardmäßig auch die Originaldatei.

```
gzip [OPTIONEN] [DATEI]
```

Optionen

`-c`	Zeigt den Inhalt an ohne die komprimierte Datei zu löschen; zusammen mit `-d` (*content*)
`-d`	Dekomprimiert die Datei (*decompress*)
`-n`	Speichert nicht die originalen Zeitstempel und Dateinamen(*no name*)
`-N`	Speichert die originalen Zeitstempel (Standard) (*!(no name)*)
`-q`	Unterdrückt Warnmeldungen (*quiet*)
`-r`	Verarbeitet auch die Dateien in Unterverzeichnissen (*recursive*)
`-t`	Test der Datenintegrität(*test*)
`-v`	Name und Kompressiongrad werden ausgegeben(*verbose*)
`-ZAHL`	Gibt den Kompressionsgrad mit ZAHL an; Wert zwischen 1 (niedrig) und 9 (hoch).

[3] Meine Plattenkapazität hat sich von 1990 bis 2001 von 40 MB auf 40 GB, also den Faktor 1000, erhöht.

Beispiel
Um die alte Version des Manuskripts zu komprimieren, deren Dateien im Verzeichnis `lk-0.3` liegen, kann man wie folgt vorgehen. Beachten Sie, daß jede Datei einzeln komprimiert wurde.

```
tapico@defiant:~ > ls-l lk-0.3
insgesamt 428
-rw-r--r--    1 tapico   users       27494 Okt 17 21:32 lk-backup.tex
-rw-r--r--    1 tapico   users       48607 Okt 17 21:32 lk-bootvorgang.tex
-rw-r--r--    1 tapico   users       42321 Okt 17 21:32 lk-dateisystem.tex
-rw-r--r--    1 tapico   users       40262 Okt 17 21:32 lk-grundbefehle.tex
-rw-r--r--    1 tapico   users       25652 Okt 17 21:32 lk-hilfe.tex
-rw-r--r--    1 tapico   users       13915 Okt 17 21:32 lk-installation.tex
-rw-r--r--    1 tapico   users       13394 Okt 17 21:32 lk-listen.tex
-rw-r--r--    1 tapico   users       30716 Okt 17 21:32 lk-shell.tex
-rw-r--r--    1 tapico   users       39591 Okt 17 21:32 lk-textfilter.tex
-rw-r--r--    1 tapico   users        3427 Okt 17 21:32 lk.tex
tapico@defiant:~ > gzip -v lk-0.3/*
lk-0.3/lk-backup.tex:    65.0% -- replaced with lk-0.3/lk-backup.tex.gz
lk-0.3/lk-bootvorgang.tex:       64.8% -- replaced with lk-0.3/lk-bootvorgang.tex.gz
lk-0.3/lk-dateisystem.tex:       65.2% -- replaced with lk-0.3/lk-dateisystem.tex.gz
lk-0.3/lk-grundbefehle.tex:      67.1% -- replaced with lk-0.3/lk-grundbefehle.tex.gz
lk-0.3/lk-hilfe.tex:     63.7% -- replaced with lk-0.3/lk-hilfe.tex.gz
lk-0.3/lk-installation.tex:      59.6% -- replaced with lk-0.3/lk-installation.tex.gz
lk-0.3/lk-listen.tex:    69.7% -- replaced with lk-0.3/lk-listen.tex.gz
lk-0.3/lk-shell.tex:     65.7% -- replaced with lk-0.3/lk-shell.tex.gz
lk-0.3/lk-textfilter.tex:        69.2% -- replaced with lk-0.3/lk-textfilter.tex.gz
lk-0.3/lk.tex:           58.7% -- replaced with lk-0.3/lk.tex.gz
tapico@defiant:~ > ls-l lk-0.3
insgesamt 164
-rw-r--r--    1 tapico   users        9642 Okt 17 21:32 lk-backup.tex.gz
-rw-r--r--    1 tapico   users       17105 Okt 17 21:32 lk-bootvorgang.tex.gz
-rw-r--r--    1 tapico   users       14729 Okt 17 21:32 lk-dateisystem.tex.gz
-rw-r--r--    1 tapico   users       13261 Okt 17 21:32 lk-grundbefehle.tex.gz
-rw-r--r--    1 tapico   users        9331 Okt 17 21:32 lk-hilfe.tex.gz
-rw-r--r--    1 tapico   users        5648 Okt 17 21:32 lk-installation.tex.gz
-rw-r--r--    1 tapico   users        4087 Okt 17 21:32 lk-listen.tex.gz
-rw-r--r--    1 tapico   users       10556 Okt 17 21:32 lk-shell.tex.gz
-rw-r--r--    1 tapico   users       12230 Okt 17 21:32 lk-textfilter.tex.gz
-rw-r--r--    1 tapico   users        1438 Okt 17 21:32 lk.tex.gz
tapico@defiant:~ >
```

13.5.2 gunzip

Der Befehl `gunzip` sorgt dafür, daß eine mit `gzip` gepackte Datei wieder entpackt wird. Im Prinzip ist `gunzip` nichts anderes als `gzip -d`.

```
gunzip [OPTIONEN] [DATEILISTE]
```

Daher gelten auch die gleichen Optionen wie bei `gzip`.

13.5.3 compress

Das Tool `compress` ist eines der ältesten Kompressionstools. Die komprimierte Datei endet auf `.Z`.

```
compress [DATEILISTE]
```

Da die Kompressionsraten von `compress` allerdings schlechter sind, als die der heutigen Tools, wird es kaum noch verwendet.

13.5.4 uncompress

Um mit `compress` komprimierte Dateien wieder zu entpacken, benutzt man das Toool `uncompress`.

```
uncompress [DATEILISTE]
```

13.5.5 zcat

Das Kommando `zcat` arbeitet wie der Befehl `cat` (Abschnitt 4.5.2). Im Gegensatz zu diesem gibt es den Inhalt mit `gzip` und `compress` gepackter Dateien auf den Bildschirm aus.

```
zcat [DATEILISTE]
```

Die gepackte Datei wird durch den Befehl nicht verändert.

Notizen:

XVI	Arbeitsblatt Administrative Aufgaben	16.1

Stellen Sie sicher, daß Sie auf keiner Konsole eingeloggt sind und auch keine Terminalemulationen auf der graphischen Oberfläche laufen. Führen Sie diese Aufgaben **nur** auf der Textkonsole aus und **nicht** in den Terminalemulationen des X-Window-Systems.

1 Loggen Sie sich an der Konsole 1 als *root* ein.

2 Legen Sie die Benutzer *ryker* und *troi* mit ihren Heimatverzeichnissen an. Geben Sie *ryker* das Kennwort `terra` und *troi* das Kennwort `betazed`.

3 Loggen Sie sich in Konsole 2 als *ryker* ein.

4 Loggen Sie sich in Konsole 3 als *troi* ein.

5 Auf welchem Terminal arbeiten Sie gerade?

6 Loggen Sie sich in Konsole 4 als `ryker` ein.

7 Stellen Sie fest, wer eingeloggt ist.

8 Stellen Sie fest, an welchem Terminal Sie gerade arbeiten.

9 Wechseln Sie zur Konsole 2.

10 Schreiben Sie mit dem `echo`-Befehl den Satz "Scott me up, Beamy". Leiten Sie die Ausgabe auf die Konsole 4 um.

11 Wechseln Sie zur Konsole 3 und wiederholen die Aufgabe 10. Schauen Sie auf der Konsole 4 nach dem Ergebnis.

12 Wechseln Sie zur Konsole 1 und wiederholen die Aufgabe 10. Schauen Sie auf der Konsole 4 nach dem Ergebnis.

13 Vergleichen Sie Ihr Ergebnis mit den Rechten auf die Konsole 4.

14 Wechseln Sie in die Konsole 3.

15 Erstellen Sie die Datei `.plan` mit dem Inhalt: "Heute Kaffekränzchen"

16 Lassen Sie sich ausführliche Informationen über den Benutzer *troi* anzeigen. Wurde der Inhalt der Datei `.plan` mit ausgegeben?

17 Wechseln Sie zur Konsole 2. Stellen Sie fest, wer Sie nun sind.

18 Lassen Sie sich ausführliche Informationen über den Benutzer *troi* anzeigen. Wurde der Inhalt der Datei `.plan` mit ausgegeben?

19 Sollte der Inhalt von `.plan` nicht ausgegeben worden sein, dann erläutern Sie das Grund und ändern das System so, daß der Fehler nicht mehr auftritt.

20 Sie haben gerade einen Anruf erhalten. Sie sollen den Gesprächspartner in 5 Minuten zurückrufen. Basteln Sie sich mit Bordmitteln einen Wecker, der Sie mit einem Signalton und einer Textnachricht daran erinnert.

21 Lassen Sie sich alle aktuellen at-Jobs anzeigen.

22 Wechseln Sie Ihre Benutzeridentität zu *root*.

23 Pünktlichkeit war noch nie Ihr Stärke. Sie haben nun zum wiederholten Male Ärger bekommen, weil Sie in der Mittagspause durchgearbeitet haben. Richten Sie nun einen Dienst ein, der alle Benutzer täglich um 12:15 Uhr daran erinnert, Mittagspause zu machen.

XVI	Arbeitsblatt Administrative Aufgaben	16.2

24 Unter welcher Benutzeridentät arbeiten Sie gerade?

25 Unter welchem Namen hatten Sie sich auf dieser Konsole eingeloggt?

26 Lassen Sie sich den letzten Einloggzeitpunkt der Benutzer anzeigen. Welche Benutzer haben sich noch nie eingeloggt?

27 Welche Benutzer sind zur Zeit eingeloggt?

28 Lassen Sie sich den Inhalt der Datei `/var/run/utmp` möglichst effektiv anzeigen und vergleichen Sie den Inhalt mit dem Ergebnis der vorherigen Aufgabe.

29 Verbieten Sie dem Benutzer *troi* das Anlegen von `at-Jobs`. Testen Sie den Erfolg der Maßnahme.

30 Kehren Sie zur Konsole 1 zurück.

31 Verständigen Sie sich mit Ihrem Partner am Nachbarrechner.

32 Loggen Sie sich mit SSH auf dem Nachbarrechner als `ryker` ein.

33 Auf welchem Terminal arbeiten Sie gerade?

34 Wer ist gerade eingeloggt?

35 Wie lautet ihr Loginname?

36 Loggen Sie sich wieder aus.

37 Wo befinden Sie sich nun im System?

Notizen:

Kapitel 14

Programminstallation und Kernelmanagement

Bei den heutigen Distributionen ist die Installation der mitgelieferten Programme oder besser gesagt Programmpakete durch die distributionsspezifischen Verwaltungswerkzeuge sehr einfach. In den meisten Fällen liegen die Programmpakete als RPM- (Redhat Package Manager) oder als Debian-Archiv vor. Wie kann der Benutzer nun Programme installieren, die nicht mit der Distribution mitgeliefert wurden?

Das hängt vor allem davon ab, wie das Programm vorliegt: Als ein Archiv mit dem Quellcode, als ein Archiv mit den Binärdateien mit eigenem Installationsprogramm oder als Binärdateien in Installationsarchiven vom Typ Debian oder RPM.

14.1 Kompilieren des Quellcodes

Eigentlich alle Open Source Programme sind als Quellcode verfügbar. Der große Vorteil dieser Methode ist die Flexibilität bei der Einpassung des Programms in bestehende Strukturen (Verzeichnisse, Bibliotheken), während die Weitergabe als Binärdateien oft bestimmte Bedingungen an die Form und Ausstattung des Systems stellt.

Die Installation auf diese Art und Weise ist komplizierter, da sie individuell vom Programmierer vorgegeben wird. Es hat sich aber eine bestimmte Vorgehensweise eingebürgert, nach der doch die meisten Programmierer vorgehen. Auf jeden Fall sollten Sie die Dateien `INSTALL` und `README` mit den Anweisungen des Programmierers lesen, **bevor** Sie mit der Installation beginnen.

14.1.1 Entpacken eines Tarballs

In den meisten Fällen werden die Dateien für die Installation in einem Verzeichnisbaum vertrieben. Dies schließt den Quellcode (in der Sprache C), die Make-Datei und zusätzliche Dokumentationen ein. Um diesen Baum einfach vertreiben zu können, werden die Dateien und die Struktur in einem `tar`-Archiv (13.4.1) zusammengefaßt und mit `gzip` (13.5.1) zu einem Tarball (13.4.2) gepackt. Entpackt werden kann das Archiv auf verschiedene Arten:

```
barclay@enterprise:~> gzip -d quellcode.tar.gz
barclay@enterprise:~> tar xvf quellcode.tar

barclay@enterprise:~> gunzip quellcode.tar.gz
barclay@enterprise:~> tar xvf quellcode.tar

barclay@enterprise:~> gzip -cd quellcode.tar.gz | tar xv

barclay@enterprise:~> tar zxvf quellcode.tar.gz
```

14.1.2 Aufbau von C-Programmen

Die meisten Programme für Linux sind in der Programmiersprache C bzw. C++[1] geschrieben. Bei größeren Projekten ist es üblich den Quellcode der Programme in mehreren einzelnen Dateien zu erstellen und zu bearbeiten. Diese Dateien mit der Endung `.c` müssen dann einzeln kompiliert werden. Es entstehen sogenannte Objekt-Dateien mit der Endung `.o`. In einem zweiten Arbeitsgang werden dann diese Dateien zu dem eigentlichen Programm zusammengelinkt.

Beispiel
An diesem Beispiel für ein zugegeben ziemlich primitives C-Programm kann der Vorgang nachvollzogen werden. Sie werden in den nächsten Abschnitten noch einmal darauf stoßen.

Als Erstes erstellen Sie die C-Datei `start.c`, die die Methode `main()` enthält.

Listing 14.1 *start.c*[2]

```
int main() {
  printHallo();
}
```

In der Methode `main()` wird die Methode `printHallo()` aufgerufen. Diese ist weder eine Methode aus den Standard-C-Bibliotheken noch in der gleichen Datei vorhanden. Sie ist nämlich in der Datei `hallo.c` definiert worden.

Listing 14.2 *hallo.c*

```
#include <stdio.h>

void printHallo() {
  printf("Hallo Fibelleser!\nEs hat geklappt\n");
}

```

Die beiden Quellcode-Dateien müssen nun zuerst kompiliert werden. Dazu verwenden Sie den C-Kompiler `gcc`. Der Schalter `-c` bewirkt, daß die angegebene Datei zu einer Objekt-Datei kompiliert wird. Dann werden die beiden Objektdateien durch `gcc` mit dem Schalter `-o` zu der Programmdatei `gruss` zusammengelinkt. Dann noch schnell die Rechte für das neue Programm geändert und fertig ist der Gruß.

```
barclay@enterprise:~/cprog> gcc -c start.c
barclay@enterprise:~/cprog> gcc -c hallo.c
barclay@enterprise:~/cprog> gcc -o gruss start.o hallo.o
barclay@enterprise:~/cprog> chmod 755 gruss
barclay@enterprise:~/cprog> gruss
Hallo Fibelleser!
Es hat geklappt
barclay@enterprise:~/cprog>
```

14.1.3 `configure`

Die meisten größeren Quellcodepakete enthalten ein "configure"-Skript. Dieses Skript muß vom Benutzer weder bearbeitet oder mit Schaltern beim Starten konfiguriert werden. Seine Aufgabe ist es, die Systemkonfiguration auf Kompiler, Bibliotheken und andere wichtige Elemente für die Kompilierung zu testen. Auf der Basis der ermittelten Informationen schreibt das Skript eine individuelle Installationskonfigurationsdatei (Make-Datei) passend für das System. Sollte `configure` Fehler wie fehlende Bibliotheken finden, so meldet das Skript das mit einer Fehlermeldung. Erfahrungsgemäß arbeitet `configure` in den meisten Fällen erfolgreich und Sie können mit dem eigentlichen Kompilieren beginnen.

[1] C++ ist die objektorientierte Weiterentwicklung von C.
[2] Die Zahlen am Anfang der Zeilen dienen nur der Orientierung und gehören nicht zum Programm.

14.1.4 `make`

Das Programm `make` ist ein Hilfsmittel für das Kompilieren. Wenn der Quellcode aus mehreren Dateien besteht, muß jede dieser Dateien einzeln kompiliert und dann alle Dateien zu einem Programm zusammgelinkt werden.

`make` erledigt das, in dem in einer Konfigurationsdatei Ziele und Abhängigkeiten definiert werden. Das Ziel einer solchen Datei ist die Kompilierung eines Programms. Dieses Programm hängt ab von den Objekt-Dateien, die wiederum abhängen von den jeweiligen Quellcode-Dateien. Der Vorteil bei der Verwendung von `make` ist unter anderem, daß die Änderungszeit von bereits bestehenden Objektdateien mit den jeweiligen Quellcode-Dateien verglichen werden. Normalerweise sind die Objektdateien immer jünger als die Quellcode-Dateien. In diesem Fall werden die Quellcode-Dateien nicht noch einmal kompiliert. Ist die Quellcode-Datei aber jünger als die Objektdatei, so wurde sie wahrscheinlich seit dem letzten Kompiliervorgang verändert und wird deshalb bei diesem Lauf erneut kompiliert. Dadurch sparen Sie enorm viel Zeit, da bei kleinen Änderungen keine komplette Neukompilierung notwendig ist.

Am Beispiel des Programms aus Abschnitt 14.1.2 möchte ich Ihnen den Aufbau einer solchen Make-Datei zeigen.

Das Ziel ist die Erstellung des Programms `gruss`. Dieses besteht aus den Objekt-Dateien `start.o` und `hallo.o`. Wir können also die Beziehung zwischen dem Programm und den Objekt-Dateien so bezeichnen:

```
gruss: start.o hallo.o
```

Die Objektdateien `start.o` und `hallo.o` hängen mit den Quellcode-Dateien `start.c` und `hallo.c` zusammen. In Kurzform ausgedrückt ergibt das:

```
start.o: start.c
hallo.o: start.c
```

Zu diesen Abhängigkeiten gehören die ausführenden Kommandos, die hinzugefügt werden müssen.

```
gcc -c start.c
gcc -c hallo.c
gcc -o gruss start.o hallo.o
```

Um die Make-Datei etwas unabhängiger vom Kompiler zu gestalten, kann sein Name durch eine Variable ersetzt werden.

```
comp=gcc
```

Wenn wir alles kombinieren, erhalten wir unsere Konfigurationsdatei `Makefile`. Achten Sie auf jeden Fall darauf, daß die Einrückung durch einen Tabulator erfolgt.

Listing 14.3 *Makefile*

```
comp = gcc

gruss: start.o hallo.o
	$(comp) -o gruss start.o hallo.o

start.o: start.c
	$(comp) -c start.c

hallo.o: hallo.c
	$(comp) -c hallo.c
```

Nach der Erstellung der Datei `Makefile`, nach der `make` im Normalfall sucht, führen wir den Befehl `make` aus. Dabei sollten eventuell vorhandene Dateien `gruss`, `start.o` und `hallo.o` vorher entfernt worden sein.

```
barclay@enterprise:~/cprog> vi Makefile
barclay@enterprise:~/cprog> make gruss
gcc -c start.c
gcc -c hallo.c
gcc -o gruss start.o hallo.o
barclay@enterprise:~/cprog> make gruss
make: ‘gruss’ is up to date.
barclay@enterprise:~/cprog>
```

Beim erneuten Ausführen von `make` erkennt das Programm, daß keine Änderungen vorgenommen worden und meldet dies dem Benutzer. Werden allerdings Quellcode-Dateien bearbeitet, so ist ihr Änderungsdatum jünger als das der Objekt-Datei. In diesem Fall wird neu kompiliert. Allerdings werden nur die Abhängigkeiten abgearbeitet, die von den Änderungen betroffen waren.

```
barclay@enterprise:~/cprog> vi hallo.c
barclay@enterprise:~/cprog> ls -l
insgesamt 48
-rw-r--r--    1 barclay      users         146 Jan 22 22:09 Makefile
-rw-r--r--    1 barclay      users       13719 Jan 22 22:09 gruss
-rw-r--r--    1 barclay      users          94 Jan 22 22:18 hallo.c
-rw-r--r--    1 barclay      users         980 Jan 22 22:09 hallo.o
-rw-r--r--    1 barclay      users          31 Jan 22 21:15 start.c
-rw-r--r--    1 barclay      users         812 Jan 22 22:09 start.o
barclay@enterprise:~/cprog> make gruss
gcc -c hallo.c
gcc -o gruss start.o hallo.o
barclay@enterprise:~/cprog>
```

Dieses `Makefile` ist nur ein schwaches Beispiel für die Möglichkeiten, die in dem Programm `make` stecken. Diese Möglichkeiten sprengen aber den Rahmen des Levels 1 der LPI-Prüfung und dieses Kapitels.

14.1.5 Installation

Für die meisten Quellcode-Programme wurde ein bestimmter Ort im Verzeichnisbaum für die ausführbaren Dateien festgelegt. In den meisten Fällen landen die Dateien im Verzeichnis `/usr/local/bin`. Um die Installation zu vereinfachen gibt es in manchen `Makefile`-Dateien ein Ziel namens `install`, daß den Installationsort definiert. Durch die Ausführung des Befehls `make install` werden die entstandenen Dateien kopiert und mit den erforderlichen Attributen und Rechten versehen.

Aber Vorsicht! Das eingestellte Installationsverzeichnis im `Makefile` kann inkompatibel mit dem Verzeichnisbaum Ihrer Linux-Distribution sein. Auch der Wechsel des Standardinstallationsverzeichnis bei einem Versionswechsel kann zu massiven Problemen führen.

14.2 Verwaltung von gemeinsam genutzten Bibliotheken

Ein Programm besteht in der Regel aus vielen immer wieder benutzten Grundfunktionen. Damit der Programmierer diese nicht immer wieder neu schreiben bzw. den Code der Funktionen in seinen Code integrieren muß, werden solche Funktionen in sogenannten Bibliotheken zusammengefaßt. Die meisten Kompiler sind mit solchen Grundbibliotheken ausgestattet. Wenn Sie sich unser Beispiel aus Abschnitt 14.1.2 ansehen, so enthält dies die Funktion `printf`. Diese Funktion ist Bestandteil der Bibliotheken des `gcc`-Kompilers und wird nun statisch in das Programm eingebunden. Dies ist u.a. ein Grund dafür, daß die Objektdatei deutlich größer ist als die Quellcodedatei.

`printf` ist natürlich eine sehr einfach Funktion. Es gibt auch viel kompliziertere Funktionen, die z. B. die Arbeit mit den Fenstern des X-Window-Systems ermöglichen. Da diese Funktionen natürlich von jedem X-Window-Programm benötigt werden, ist es von Vorteil, wenn sich die Programme diese Bibliotheken gemeinsam nutzen könnten. Der erste Schritt dahin ist, daß die Bibliotheksfunktion nicht statisch in das Programm eingebunden werden, sondern dynamisch verlinkt werden. D. h. das Programm verweist nur auf die Funktion in der Bibliothek. Wenn es nun eine Trennung von Programm und Bibliothek gibt, können jetzt mehrere Programme auf die gleichen Bibliotheken verweisen. Diese Bibliotheken werden als *Shared Libraries* bezeichnet. Für Windows-Jünger: Die Shared Libraries sind in ihrem Konzept mit den MS-Windows-DLLs (*Dynamic Link Libraries*) zu vergleichen.

Der Nachteil dieser dynamischen Bibliotheken liegt darin, daß diese natürlich installiert sein müssen, wenn ein Programm laufen soll. Um zu überprüfen welche Bibliotheken ein Programm braucht, wird das Tool `ldd` (14.2.2) verwendet. Die Übersicht über die installierten bzw. im System registrierten Dateien liefert der Befehl `ldconfig` (14.2.3), der auch für das Update der Liste verantwortlich ist.

Die dynamische Einbindung von Bibliotheken bietet sich zumeist bei großen Programmen an, die komplexe Standardfunktionen benötigen. Je kleiner und rudimentärer das Programm ist, desto besser ist eine statische Einbindung der Bibliotheksfunktionen.

Die Bibliotheken befinden sich meistens in den speziell dafür vorgesehenen Verzeichnissen. Typische Bibliotheksverzeichnisse sind `/lib`, `/usr/lib`, `/usr/local/lib`, `/usr/X11R6/lib` und `/opt/lib`. Die letzte Ziffer im Namen der Bibliothek ist die Hauptversionsnummer. Oft handelt es sich hierbei aber nur um einen Link, der auf die tatsächlich installierte Version zeigt.

```
tapico@defiant:~> ls -lG /lib/ld*
-rwxr-xr-x    1 root   94543 Sep 20 05:52 /lib/ld-2.2.4.so
lrwxrwxrwx    1 root      11 Nov 22 15:53 /lib/ld-linux.so.2 -> ld-2.2.4.so
```

Als Anwender werden sie mit den Bibliotheken nur dann konfrontiert, wenn Sie fehlen. In den meisten Fällen tritt ein solches Problem nur dann auf, wenn Sie ein Programm nachträglich installieren, das nicht von der Distribution stammt. So besitzen z. B. die Konfigurationsprogramme `yast` und `yast2` von SuSE die Möglichkeit andere benötigte Programmpakete wie z. B. Bibliotheken automatisch mit für die Installation auswählen zu lassen. Es kann vorkommen, daß neuere Programmversionen auch neuere Bibliotheken benötigen, wie es auch vorkommen kann, daß ältere Programme mit neueren Bibliotheken Probleme bekommen.

Als erste Hilfe bei Problemen bietet sich der Befehl `ldd` (14.2.2) an, um überhaupt festzustellen, welche Bibliotheken benötigt werden.

14.2.1 Runtime Linker: `ld.so`

Wie aber findet nun ein Programm die benötigten Bibliotheken. Für die Verbindung zwischen Programmen und Bibliotheken ist der Runtime Linker `ld.so` bzw. `ld-linux.so` zuständig. Damit er nicht ständig alle Verzeichnisse nach Bibliotheken durchsuchen muß, wertet er die Datei `/etc/ld.so.cache` aus. In dieser Datei befindet sich eine Liste der Bibliotheken mit allen relevanten Daten (Versionsnummer, Zugriffspfade etc.). Bei manchen Distributionen wird diese Datei automatisch bei jedem Start mit dem Programm `ldconfig` (14.2.3) aktualisiert. Werden neue Bibliotheken installiert, so muß der Befehl `ldconfig` manuell gestartet werden.

14.2.2 `ldd`

Der Befehl `ldd` zeigt für das angegebene Programm die benötigten Bibliotheken (*Shared Libraries*) an.

`ldd [OTPIONEN] PROGRAMM|BIBLIOTHEK`

Optionen	
`-v`	Ausführliche Informationen

Beispiele
Kleine Programme, wie z. B. `cp` brauchen nur wenige Bibliotheken, während andere Programme, wie z. B. der X-Editor `nedit` da wesentlich anspruchsvoller sind.

```
tapico@defiant:~> ldd /bin/cp
        libc.so.6 => /lib/libc.so.6 (0x40023000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
tapico@defiant:~> ldd /usr/X11R6/bin/nedit
        libm.so.6 => /lib/libm.so.6 (0x40023000)
        libXm.so.2 => /usr/X11R6/lib/libXm.so.2 (0x40045000)
        libXpm.so.4 => /usr/X11R6/lib/libXpm.so.4 (0x401f8000)
        libXext.so.6 => /usr/X11R6/lib/libXext.so.6 (0x40207000)
        libXt.so.6 => /usr/X11R6/lib/libXt.so.6 (0x40215000)
        libSM.so.6 => /usr/X11R6/lib/libSM.so.6 (0x40263000)
        libICE.so.6 => /usr/X11R6/lib/libICE.so.6 (0x4026e000)
        libX11.so.6 => /usr/X11R6/lib/libX11.so.6 (0x40286000)
        libXmu.so.6 => /usr/X11R6/lib/libXmu.so.6 (0x40369000)
        libXp.so.6 => /usr/X11R6/lib/libXp.so.6 (0x4037f000)
        libc.so.6 => /lib/libc.so.6 (0x40387000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

14.2.3 `ldconfig`

Um schneller seine Arbeit verrichten zu können, wertet der Runtime Linker `ld.so` die Binärdatei `ld.so.cache` aus, die die Liste der installierten Bibliotheken enthält. Um diese Datei zu aktualisieren wird das Programm `ldconfig` verwendet.

```
ldconfig [OPTIONEN] [VERZEICHNIS|BIBLIOTHEK]
```

Optionen

`-p`	Zeigt den Inhalt der aktuellen Cache-Datei an, anstatt sie zu erstellen (*print*)
`-v`	Ausführliche Informationen während der Ausführung (*verbose*)

Die Liste der zu durchsuchenden Verzeichnis steht in der Datei `/etc/ld.so.conf` . Die Verzeichnisse `/lib` und `/usr/lib` fehlen, da sie immer durchsucht werden. Hier ein Beispiel für die Datei aus SuSE 7.3:

```
tapico@defiant:~> cat /etc/ld.so.conf
/lib-aout
/usr/X11R6/lib/Xaw95
/usr/X11R6/lib/Xaw3d
/usr/X11R6/lib
/usr/i486-linux/lib
/usr/i486-linux-libc5/lib=libc5
/usr/i486-linux-libc6/lib=libc6
/usr/i486-linuxaout/lib
/usr/i386-suse-linux/lib
/usr/local/lib
/usr/openwin/lib
/opt/kde/lib
/opt/kde2/lib
/opt/gnome/lib
```

Beispiel
Eine neues Programm wurde installiert. Für dieses Programm wurden auch die passenden Bibliotheken installiert. Trotzdem kann das Programm nicht gestartet werden, da die Bibliotheken dem Runtime Linker nicht bekannt sind.

Zuerst wird geprüft, ob das Verzeichnis für die Bibliotheken auch in der Liste für `ldconfig` steht:

```
root@defiant:~> cat /etc/ld.so.conf
/lib-aout
```

```
/usr/X11R6/lib
/usr/i486-linux/lib
/usr/i486-linuxaout/lib
/usr/i386-suse-linux/lib
/usr/local/lib
```

Sollte das Verzeichnis nicht in der Liste stehen, so muß es hinzugefügt werden. Danach wird das Programm `ldconfig` gestartet und der Cache des Runtime Linkers aktualisiert. Nun steht einem erfolgreichem Programmstart nichts mehr im Weg.

```
root@defiant:~> vi ld.so.conf
root@defiant:~> ldconfig
```

14.3 Installation von RPM-Pakete

Der Vertrieb von Programmen in sogenannten Paketen erleichtert die Installation und Verwaltung der Programme erheblich. Dieser Abschnitt beschäftigt sich mit dem Red Hat Package Manager.

Die Installation von Programmen aus dem Quellcode hat einige deutliche Nachteile.

- Es ist schwer den Überlick über die installierte Programme und ihre Versionen zu behalten.
- Es ist praktisch unmöglich herauszufinden, welche Datei zu welchem Programmpaket gehört.
- Updates erfordern einen großen Aufwand, da bei einer Installation durch `make` (14.1.4) auch vorhandene Konfigurationsdateien überschrieben werden. Es existiert keine Beschreibung über die Funktion der Dateien.
- Daneben existieren auch keine Informationen, welche Pakete und Bibliotheken für die Ausführung des Programms benötigt werden. Es kann also Paket A installiert werden, ohne daß das benötigte Paket B auch installiert wurde.

RPM-Paket gibt es meistens im Doppelpack. In der einen Datei (Binärpaket) befinden sich die Binärdateien sowie die für die Installation erforderlichen Informationen und Konfigurationsdateien. Die zweite Datei (Quellpaket) enthält die für die Erstellung des Binärpakets benötigten Dateien mit dem Quellcode.

Der Dateiname eines solchen Pakets enthält wichtige Informationen: `jdk-1.3.1-1.i386.rpm` bezeichnet das Paket `jdk` mit der Versionsnummer `1.3.1`, rpm-Release 1. Die Release-Nummer ist meistens 1 oder wird ganz weggelassen. Nur wenn Fehler im RPM-Paket selber behoben, Änderungen durchgeführt oder Dokumentationsdateien hinzugefügt wurden, wird diese Nummer hochgezählt. Die Release-Nummer bezieht sich also auf das RPM-Paket selber, während sich die Versionsnummer auf das eigentliche Programm bezieht.

Die Kennung `i386` weißt darauf hin, daß dieses Paket Binärdateien für PC-Prozessoren enthält. Steht dagegen ein `src` dort, dann handelt es sich hier um das Paket mit den Quellcodedateien.

Neben den Binärdateien enthält das Paket auch wichtige Informationen:

- Eine kurze Paketbeschreibung
- Weitere Informationen über die Versionsnummer
- Einordnung in die Gruppenhierachie
- Abhängigkeit von anderen Paketen. Dies ist wichtig, falls das Paket bestimmte Programme (Interpreter, Bibliotheken) voraussetzt.

Um die Daten verwalten zu können, benutzt der Manager für die RPM-Pakete eine Datenbank für alle installierten Binärpakete. Diese Datenbank besteht aus mehreren `*.rpm`-Dateien im Verzeichnis `/var/lib/rpm`. Hier die Datenbank für SuSE 7.3:

```
tapico@defiant:~> ls -l /var/lib/rpm
insgesamt 32908
-rw-r--r--    1 root     root        16384 Nov 22 16:27 conflictsindex.rpm
-rw-r--r--    1 root     root      5230592 Nov 22 16:27 fileindex.rpm
-rw-r--r--    1 root     root        16384 Nov 22 16:27 groupindex.rpm
-rw-r--r--    1 root     root        28672 Nov 22 16:27 nameindex.rpm
-rw-r--r--    1 root     root     28339656 Nov 22 16:27 packages.rpm
-rw-r--r--    1 root     root       159744 Nov 22 16:27 providesindex.rpm
-rw-r--r--    1 root     root       172032 Nov 22 16:27 requiredby.rpm
-rw-r--r--    1 root     root        16384 Nov 22 16:27 triggerindex.rpm
```

Bei den `*.rpm`-Dateien handelt es sich nicht um RPM-Pakete, sondern um Binärdateien in einem rpm-spezifischen Format. Sie dürfen auch keinen Fall per Hand verändert werden. Auch die Deinstallaton von Paketen darf nicht durch einfaches Löschen erfolgen, sondern muß immer durch einen RPM-Manager erledigt werden.

14.3.1 Der Manager `rpm`

`rpm` ist ein sehr mächtiger und umfangreicher Paket-Manager für RPM-Pakete.

`rpm [OPTIONEN]`

Optionen

`-i`	Installation von Paketen (`--install`)
`-U`	Aktualisierung des Pakets auf eine höhere Versionsnummer (`--upgrade`)
`-e`	Entfernt installierte Pakete (`--erase`)
`-q`	Liefert Informationen über Pakete
`-h`	Zeigt einen Fortschrittsbalken von 50 Schweinegattern (`#`) während der Installaton an. (`--hash`)
`-v`	Ausführliche Informationen
`-vv`	Sehr ausführliche Informationen
`-a`	Zeigt eine Liste aller installierte Pakete (`--all`)(nur mit `-q`)
`-f DATEI`	Zeigt das Paket, das die angegebene Datei enthält (nur mit `-q`)
`-p PAKETDATEI`	Liefert Informationen über die angegebene Paketdatei (nur mit `-q`)
`-c`	Listet nur Konfigurationsdateien auf (`--configfiles`)(nur mit `-q`)
`-d`	Listet nur Dokumentationsdateien auf (`--docfiles`)(nur mit `-q`)
`-i PAKET`	Liefert Informationen über das angegebene Paket (nur mit `-q`)
`-l PAKET`	Listet die enthaltenen Dateien auf (`--list`)(nur mit `-q`)
`-R`	Zeigt die Abhängigkeiten an (`--requires`)(nur mit `-q`)
`-V`	Überprüfung von Dateien auf ihre Übereinstimmung mit den Angaben im Paket
`--force`	Erlaubt das Ersetzen von existierenden Paketen und Dateien
`--nodeps`	Führt keine Abhängigkeitsprüfung durch
`--quiet`	Beschränkt die Ausgabe auf Fehlermeldungen
`--test`	Testet nur und installiert nicht

Installation und Upgrade

Im Installationsmodus (`rpm -i`) werden Pakete installiert. Der Upgrade-Modus (`rpm -U`) wird dazu verwendet um installierte Pakete auf eine höhere Versionsnummer zu aktualisieren.

Um ein neues Paket zu installieren müssen Sie nur den Befehl `rpm -i` zusammen mit dem Paketnamen verwenden. Wenn das Paket auf auf anderen Paketen basiert, die nicht installiert sind, gibt `rpm` eine passende Fehlermeldung aus. Sie müssen dann die fehlende Pakete zuerst installieren.

```
root@defiant:~ # rpm -ivh jdk-1.3.1-fcs.i386.rpm
jdk                          ##################################################
```

Das Upgrade eines Pakets zu einer neuen Version wird durch die Option `-U` erledigt. Der Upgrade-Modus ist ein spezieller Installationsmodus. Existiert das zu aktualisierende Paket nicht, dann verhält sich `-U` wie der Schalter `-i`.

```
root@defiant:~ # rpm -U jdk-1.3.1-fcs.i386.rpm
```

Deinstallation

Dieser Modus wird für das Entfernen von installierten Paketen verwendet. Dabei wird natürlich ein Paket nur dann entfernt, wenn kein anderes Paket auf dieses Paket angewiesen ist. Für die Deinstallition wird der Schalter `-e` verwendet. Die Schalter `-u` und `--uninstall` sind veraltet.

```
root@defiant:~ # rpm -uv jdk-1.3.1-fcs
Fehler: -u und --uninstall sind veraltet und funktionieren nicht mehr.
Fehler: Benutzen Sie stattdessen -e oder --erase
root@defiant:~ # rpm -ev jdk-1.3.1-fcs
```

Informationen

Installierte und nicht installierte Pakete können im Informationsmodus (*Query-Mode*) mit dem Befehl `rpm -q` befragt werden.

Ausführliche Informationen über ein installiertes Paket bekommen Sie mit der Schalterkombination `-qi`.

```
root@defiant:~ # rpm -qi te_latex-1.0.7-285
Name        : te_latex                     Relocations: (not relocateable)
Version     : 1.0.7                             Vendor: SuSE GmbH, Nuernberg, Germany
Release     : 285                           Build Date: Son 23 Sep 2001 22:43:43 CEST
Install date: Don 22 Nov 2001 16:21:01 CET      Build Host: levi.suse.de
Group       : Applications/Publishing/TeX   Source RPM: tetex-1.0.7-285.src.rpm
Size        : 25835581                         License: 1999 - not specified
Packager    : feedback@suse.de
Summary     : All about LaTeX
Description :
This package provides LaTeX, to be exact LaTeX2e, and a huge amount
software around LaTeX.  This package is required by the most
(La)TeX documents.

Authors:
--------
    Leslie Lamport <lamport@pa.dec.com>
    Johannes Braams
    David Carlisle
    Alan Jeffrey
    Frank Mittelbach <frank.mittelbach@uni-mainz.de>
    Chris Rowley
    Rainer Schöpf
    Members of the LaTeX3 project

SuSE series: tex
```

Wenn die Informationen über eine RPM-Datei gefragt sind, kommt der Schalter `-p` ins Spiel.

```
root@defiant:~ # rpm -qpi jdk-1.3.1.i386.rpm
Name        : jdk                          Relocations: (not relocateable)
Version     : 1.3.1                             Vendor: Sun Microsystems
Release     : fcs                           Build Date: Son 06 Mai 2001 12:46:01 CEST
Install date: (not installed)               Build Host: lady-linux
Group       : Development/Tools             Source RPM: jdk-1.3.1-fcs.src.rpm
Size        : 59866441                         License: 1994-2001 Sun Microsystems, Inc.
Packager    : Java Software <j2se-comments@java.sun.com>
```

```
URL         : http://java.sun.com/linux
Summary     : Java(TM) 2 Software Development Kit, Standard Edition
Description :
The Java 2 SDK, Standard Edition includes the Java Virtual Machine,
core class libraries and tools used by programmers to develop Java software
applets and applications. The SDK also provides the foundation for IDE
(Integrated Development Environment) tools such as Sun's Forte for Java,
Community Edition, the Java(TM) 2 Platform, Enterprise Edition (J2EE),
Java-based application servers and more.
The Java 2 Software Development Kit, SDK, is a development
environment for building applications, applets, and components
that can be deployed on the Java platform.
The Java 2 SDK software includes tools useful for developing and
testing programs written in the Java programming language and
running on the Java platform.  These tools are designed to be used
from the command line.  Except for applet viewer, these tools do not
provide a graphical user interface.
```

Eine Liste der im Paket enthaltenen Dateien bekommen Sie durch den Schalter `-l`.

```
root@defiant:~ # rpm -qlp jdk-1.3.1.i386.rpm | less
```

Sie können sich aber auch aus der Liste die Dokumentationsdateien `-d` oder Konfigurationsdateien `-c` herauspicken

```
root@defiant:~ # rpm -qd cron
/usr/share/doc/packages/cron/CHANGES
/usr/share/doc/packages/cron/CONVERSION
/usr/share/doc/packages/cron/FEATURES
/usr/share/doc/packages/cron/MAIL
/usr/share/doc/packages/cron/MANIFEST
/usr/share/doc/packages/cron/README
/usr/share/doc/packages/cron/THANKS
/usr/share/man/man1/crontab.1.gz
/usr/share/man/man5/crontab.5.gz
/usr/share/man/man8/cron.8.gz
root@defiant:~ # rpm -qc cron
/etc/crontab
/etc/init.d/cron
/var/spool/cron/deny
```

Sie können sogar herausfinden, aus welchem Paket eine Datei stammt. Aber nicht jede Datei wurde aus einem Paket extrahiert.

```
root@defiant:~ # rpm -qf /bin/cp
fileutils-4.1-51
root@defiant:~ # rpm -qf /etc/fstab
die Datei »/etc/fstab« gehört zu keinem Paket
```

Und natürlich können Sie sich alle installierten Pakete anzeigen lassen. Da es sich um einige hundert Pakete handelt, empfielt es sich die Ausgabe zu filtern.

```
root@defiant:~ # rpm -qa | sort | less
root@defiant:~ # rpm -qa | wc -l
    581
root@defiant:~ # rpm -qa | grep netscape
netscape-plugins-4.78-14
netscape-4.78-34
```

Überprüfung

Dateien aus installierten Paketen können mit der im Paket vorgegebenen Konfiguration verglichen werden, wenn der Befehl `rpm -V` verwendet wird.

```
root@defiant:~ # rpm -V apache
S.5....T c /etc/httpd/httpd.conf
S.5....T c /etc/httpd/suse_addmodule.conf
S.5....T c /etc/httpd/suse_define.conf
S.5....T c /etc/httpd/suse_include.conf
S.5....T c /etc/httpd/suse_loadmodule.conf
S.5....T c /etc/httpd/suse_text.conf
missing    /var/log/httpd/ssl_engine_log
missing    /var/log/httpd/ssl_request_log
missing    /var/run/httpd.pid
```

14.4 Der Kernel

In Linux ist der *Kernel* der Kern der Software, die Ihr System verwaltet und kontrolliert. Er kontrolliert Hardware, Speicher und verwaltet die Prozesse. Daneben ist er eine Schnittstelle für den indirekten Zugriff der Anwenderprogramme auf die Hardware Ihres Systems.

Die Versionsnummer eines Kernels setzt sich aus mehreren Angaben zusammen: `2.4.16-4GB` bedeutet, es handelt sich um die Kernelversion 2.4 mit dem Patchlevel 16[3], der einen Arbeitspeicher von 4 GB verwalten kann. Die zweite Ziffer der Kernelversion hat eine besondere Bedeutung. Kernelversionen mit einer geraden Ziffer am Ende (z. B. 2.0, 2.2, 2.4) gelten als stabile Kernels. Ungerade Nummern wie 2.1, 2.3 und 2.5 bezeichnen experimentelle Kernels, in denen neue Funktionen getestet werden. Solche Kernels sollten Sie nicht auf Produktionssystemen einsetzten, da Sie in der Regel instabil sind.

Informationen über den eingesetzten Kernel erhalten Sie über den Befehl `uname`.

14.4.1 uname

Das Programm `uname` zeigt Informationen über das installierte System und den Kernel an.

```
uname [OPTIONEN]
```

Die Verwendung von `uname` ohne Schalter ist wie `uname -s`.

Optionen

-a	Gibt alle Informationen aus (`--all`)
-m	Gibt den Hardwaretyp aus (`--machine`)
-n	Gibt den Netzwerkname des Rechners aus (`--nodename`)
-r	Gibt die Release-Nummer des Systems aus (`--release`)
-s	Gibt den Systemnamen aus (`--sysname`)
-p	Gibt den Prozessortyp aus (`--processor`)
-v	Gibt die Versionsnummer aus (*version*)

Beispiele

Der Befehl ohne Schalter gibt einfach nur den Namen des Betriebssystems aus.

```
tapico@defiant:~> uname
Linux
```

Werden mehrere Schalter oder der Schalter `-a` verwendet, dann werden die Informationen in der Reihenfolge `-snrvmp` ausgegeben.

[3] D. h. die 16. Fehlerkorrektur wurde berücksichtigt.

```
tapico@defiant:~> uname -a
Linux defiant 2.4.10-4GB #1 Tue Sep 25 12:33:54 GMT 2001 i686 unknown
```

14.5 Verwaltung von Kernelmodulen

Die ersten Kernel waren monolithische Kernel, in denen alle Funktionen fest einkompiliert waren. Dies ergibt einen sehr schnellen und stabilen Kernel. Der Nachteil eines solchen Vorgehens zeigte sich aber, als der Kernel immer mehr Funktionen umfaßte und dadurch immer größer wurde. Außerdem mußte für jede neue Funktion ein neuer Kernel kompiliert werden. Um diesen Nachteil zu umgehen wurde der modularisierte Kernel entwickelt. Hier werden die meisten Hardwaretreiber als zusätzliche Objektdateien dem Kernel zur Verfügung gestellt und bei Bedarf im laufenden Betrieb hinzugeladen. Diese Module können im laufenden Betrieb ausgetauscht werden und ermöglichen so ein komfortables Aktualisieren des Systems.

Im Normalfall wird ein Modul dynamisch in den laufenden Kernel eingebunden, wenn es geladen wird. Die meisten Aktionen im Zusammenhang mit den Modulen laufen automatisch ab. Manchmal ist es aber auch nötig Modul manuell zu bearbeiten um z. B. einen neuen Treiber in das laufende System einzubinden. Die Liste der geladenen Module liefert der Befehl `lsmod` (14.5.2). Module werden mit `insmod` (14.5.3) oder `modprobe` (14.5.6) in den Kernel eingebunden und durch `rmmod` (14.5.4) wieder entfernt. Informationen über eingebundene Module bekommen Sie über den Befehl `modinfo` (14.5.5).

14.5.1 Die Moduldateien

Die Kernelmodule befinden sich in einem Unterverzeichnis von `/lib/modules` mit dem Namen der Kernelversion. Es können daher auch mehrere Modulverzeichnisse vorhanden sein, wenn mehrere Kernelversionen installiert wurden.

```
root@defiant:/ # uname -r
2.4.10-4GB
root@defiant:/ # ls -lG /lib/modules/2.4.10-4GB/
insgesamt 271
lrwxrwxrwx    1 root        26 Nov 22 16:49 build -> /usr/src/linux-2.4.10.SuSE
drwxr-xr-x    2 root       341 Nov 22 16:50 dvb
drwxr-xr-x    7 root       134 Nov 22 16:50 kernel
drwxr-xr-x    2 root      4735 Nov 22 16:50 misc
-rw-r--r--    1 root    127579 Jan 23 19:23 modules.dep
-rw-r--r--    1 root        31 Jan 23 19:23 modules.generic_string
-rw-r--r--    1 root      7965 Jan 23 19:23 modules.isapnpmap
-rw-r--r--    1 root        29 Jan 23 19:23 modules.parportmap
-rw-r--r--    1 root     51647 Jan 23 19:23 modules.pcimap
-rw-r--r--    1 root     65049 Jan 23 19:23 modules.usbmap
drwxr-xr-x    2 root        61 Nov 22 16:50 net
drwxr-xr-x    2 root      1018 Nov 22 16:50 pcmcia
drwxr-xr-x    3 root      1433 Nov 22 16:50 pcmcia-external
drwxr-xr-x    2 root       136 Nov 22 16:50 thinkpad
```

In diesem Verzeichnis existieren wiederum Unterverzeichnisse für die verschiedenen Typen von Modulen. Typische Unterverzeichnisse sind:

`block` Module für ein paar blockorientierte Geräte wie RAID-Kontroller oder IDE-Bandlaufwerke.

`cdrom` Treiber für nicht standartisierte CD-ROM-Laufwerke.

`fs` Module für Dateisysteme wie ReiserFS.

`ipv4` Enthält Module, die im Zusammenhang mit IP stehen.

`misc` Hier kommt alles rein, was nicht in die anderen Kategorien paßt.

`net` Treiber für die Netzwerkkarten.

`scsi` Enthält die Treiber für die SCSI-Kontroller.

`video` Für die speziellen Treiber der Grafikkarten.

14.5.2 lsmod

Dieser Befehl zeigt Ihnen für jedes geladene Modul den Namen, die Größe, die Anzahl der Benutzungen und eine Liste mit verknüpften Modulen.

```
lsmod
```

Beispiel

```
root@defiant:/ # lsmod
Module                  Size  Used by
parport_pc             19280   1  (autoclean)
lp                      5248   0  (autoclean)
parport                22240   1  (autoclean) [parport_pc lp]
nfsd                   64880   4  (autoclean)
ipv6                  124736  -1  (autoclean)
evdev                   4160   0  (unused)
input                   3072   0  [evdev]
usb-ohci               17680   0  (unused)
iptable_nat            12656   0  (autoclean) (unused)
ip_conntrack           12848   1  (autoclean) [iptable_nat]
iptable_filter          1728   0  (autoclean) (unused)
ip_tables              10496   4  [iptable_nat iptable_filter]
reiserfs              147920   2
usbcore                47264   1  [usb-ohci]
```

14.5.3 insmod

Das Kommando `insmod` bindet ein Modul in einen laufenden Kernel ein. Das Modul wird automatisch lokalisiert und eingebunden. Diese Funktion steht natürlich nur dem Superuser zur Verfügung.

```
insmod [OPTIONEN] MODUL
```

Optionen	
-s	Gibt die Resultate an `syslogd` weiter, anstatt sie auf dem Terminal auszugeben
-v	Liefert ausführliche Informationen

Beispiel
Um ein Modul zu laden muß man nur seinen Namen angeben.

```
root@defiant:/ # insmod umsdos
Using /lib/modules/2.4.10-4GB/kernel/fs/umsdos/umsdos.o
```

Basiert das Modul hingegen auf ein anderes Modul, so wird das Modul nicht geladen und es werden Fehlermeldungen ausgegeben. In diesem Fall müssen natürlich die benötigten Module vorher geladen werden.

14.5.4 rmmod

Um ein Modul wieder aus dem laufenden Kernel zu entfernen, wird der Befehl `rmmod` benutzt. Dies klappt allerdings nur, wenn das Modul unbenutzt ist und nicht von einem anderen Modul benötigt wird. Natürlich darf nur der Superuser Module aus dem laufenden Kernel entfernen.

```
rmmod [OPTIONEN] MODULE
```

Optionen

-a	Entfernt alle unbenutzten Module
-s	Gibt die Resultate an `syslogd` weiter, anstatt sie auf dem Terminal auszugeben
-v	Liefert ausführliche Informationen

Beispiele
Module werden einfach durch Angabe ihres Namens beim `rmmod`-Befehl aus dem laufenden Kernel entfernt.

```
root@defiant:/ # insmod ntfs
Using /lib/modules/2.4.10-4GB/kernel/fs/ntfs/ntfs.o
root@defiant:/ # lsmod | grep ntfs
ntfs                  47216   0  (unused)
root@defiant:/ # rmmod ntfs
root@defiant:/ # lsmod | grep ntfs
root@defiant:/ #
```

Sollte aber der Kernel das Modul benötigen, wird der Versuch mit einer Fehlermeldung quittiert.

```
root@defiant:/ # rmmod usbcore
usbcore: Device or resource busy
```

14.5.5 modinfo

Mit `modinfo` bekommen Sie Informationen über das Modul aus der Moduldatei. Allerdings enthalten nicht alle Module ausführliche Informationen. Manche begnügen sich mit einer kurzen beschreibenden Zeile.

```
modinfo [OPTIONEN] MODULDATEI
```

Optionen

-a	Zeigt den Autoren des Moduls an (*author*)
-d	Zeigt die Beschreibung des Moduls an (*description*)
-p	Zeigt die Parameter des Moduls an (*parameters*)

Beispiel
Hier sind zwei Beispiel für die Module `umsdos` und `sonypi`.

```
root@defiant:/ # modinfo  /lib/modules/2.4.10-4GB/kernel/fs/umsdos/umsdos.o
filename:    /lib/modules/2.4.10-4GB/kernel/fs/umsdos/umsdos.o
description: <none>
author:      <none>
root@defiant:/ # modinfo  /lib/modules/2.4.10-4GB/kernel/drivers/char/sonypi.o
filename:    /lib/modules/2.4.10-4GB/kernel/drivers/char/sonypi.o
description: "Sony Programmable I/O Control Device driver"
author:      "Stelian Pop <stelian.pop@fr.alcove.com>"
parm:        minor int, description "minor number of the misc device, default is -1 (automatic)"
parm:        verbose int, description "be verbose, default is 0 (no)"
parm:        fnkeyinit int, description "set this if your Fn keys do not generate any event"
parm:        camera int, description "set this if you have a MotionEye camera (PictureBook series)"
```

14.5.6 modprobe

Wie `insmod` (14.5.3) wird `modprobe` dazu benutzt dem Kernel Module hinzuzufügen. Eigentlich ist `modprobe` nur ein Wrapper für `insmod`, stellt aber dem Benutzer wesentlich mehr Funktionen zur Verfügung. So können automatisch neben dem eigentlichen Modul auch die dafür benötigten Module automatisch geladen werden. Auch das Laden aller Module eines Verzeichnis, z. B. um das richtige Modul für eine Hardwarekomponente zu finden. ist möglich.

```
modprobe [OPTIONEN] MODULE
```

Optionen

`-a`	Lädt alle Module (*all*)
`-c`	Anzeige einer kompletten Modulkonfiguration (*configuration*)
`-l`	Auflisten der vorhandenen Module (*list*)
`-r`	Entfernt Module wie `rmmod` (14.5.4) (*remove*)
`-s`	Leitet die Ergebnisse an den Protokolldaemon `syslog` weiter (*syslog*)
`-t TAG`	Bezieht die Aktionen auf die Module des Verzeichnis `TAG` (*tag*)
`-v`	Erweiterte Bildschirmausgabe(*verbose*)

Beispiele
Um das Modul `ntfs` zu installieren reicht der folgenden Befehl.

```
root@defiant:/ # modprobe -v ntfs
/sbin/insmod /lib/modules/2.4.10-4GB/kernel/fs/ntfs/ntfs.o
Using /lib/modules/2.4.10-4GB/kernel/fs/ntfs/ntfs.o
Symbol version prefix ''
root@defiant:/ # lsmod | grep ntfs
ntfs                   47216   0  (unused)
```

Sollte das zu ladende Modul auf anderen Modulen basieren, so kann `modprobe` dies aus der jeweiligen `modules.dep` entnehmen.

14.5.7 Modulabhängigkeiten: `modules.dep`

Die Informationen über die Abhängigkeiten eines Moduls von anderen Kernelmodulen findet das Programm `modprobe` in der Datei `/lib/modules/`*Kernel-Version*`/modules.dep`. Die Zeilen mit den Abhängigkeiten sehen wie folgt aus:

```
modulname.o: Abhängigkeit_1 Abhängigkeit_2 ...
```

Hier ein Beispiel für das ext3-Dateisystem aus der `modules.dep`

```
ole@enterprise:/lib/modules/2.4.10-4GB> grep ext3 modules.dep
/lib/modules/2.4.10-4GB/kernel/fs/ext3/ext3.o:
/lib/modules/2.4.10-4GB/kernel/fs/jbd/jbd.o
```

Alle für das System zur Verfügung stehenden Module sind hier aufgelistet mit den Modulen, auf denen sie basieren. Sollte das Modul keine anderen Module benötigen, wird es trotzdem in der Datei aufgeführt. Wenn jetzt mit `modprobe` ein Modul geladen werden soll, so konsultiert das Programm erst diese Datei und lädt die benötigten Module zuerst. Erst wenn alle benötigten Module eingebunden worden sind, dann wird auch das eigentlich angeforderte Modul geladen.

Die Datei `modules.dep` muß immer auf dem neuesten Stand sein. Sollte sich in den Abhängigkeiten etwas geändert haben ohne daß diese Änderung hier erfaßt wurde, dann kann `modprobe` eventuell Module nicht erfolgreich in den Kernel einbinden. Aus diesem Grund wird in den meisten Distributionen diese Datei bei jedem Start des Systems mittels des Befehls `depmod` neu erstellt. Bei SuSE 7.3 erfolgt dies in der Datei `/etc/rc.d/boot` wie im folgenden Ausschnitt zu sehen ist.

```
#
# initialize database for kerneld.  This should be done earlier, but
# could cause a lot of trouble with damaged file systems.
# restart of kerneld will be done by /etc/init.d/kerneld
#
MODULES_DIR=/lib/modules/`uname -r`
if test -x /sbin/depmod -a -d $MODULES_DIR ; then
    for i in $MODULES_DIR/* $MODULES_DIR/*/* /etc/modules.conf ; do
        test -e $i || continue
        if test $i -nt $MODULES_DIR/modules.dep ; then
            rm -f $MODULES_DIR/modules.dep
            break
        fi
    done
    if test ! -e $MODULES_DIR/modules.dep ; then
        echo -n Setting up $MODULES_DIR
        /sbin/depmod -a > /dev/null 2>&1
        rc_status -v -r
    fi
fi
```

14.5.8 depmod

Das Tool `depmod` findet die Abhängigkeiten zwischen ladbaren Kernelmodulen und schreibt die dafür zuständige Konfigurationsdatei `modules.dep`.

```
depmod [OPTIONEN] [MODULLISTE]
```

Optionen

`-a`	Sucht nach Modulen in allen Verzeichnissen, die in `moduls.conf` definiert wurden(*all*)
`-A`	Wie `-a`, führt aber nur ein Update durch(*all*)
`-e`	Zeigt alle nicht aufgelösten Symbole für das Modul
`-n`	Schreibt die Abhängigkeiten auf den Bildschirm und nicht in `modules.dep` (*not*)
`-s`	Fehlermeldungen gehen an `syslog` und nicht `stderr`(*syslog*)
`-v`	Zeit alle bearbeiteten Module an(*verbose*)
`-q`	Arbeitet ohne Meldungen (*quiet*)
`-V`	Zeigt die Versionsnummer an (*Version*)
`-b BASISVERZEICHHNIS`	Setzt eine neues Basisverzeichnis fest(*Basis*)
`-C KONFIGDATEI`	Konfigurationsdatei festlegen (*Config*)
`-F SYMBOLE`	Bei der Bestimmung von Abhängigkeiten für einen nicht laufenden Kernel, müssen hier die passenden Kernelsymbole angegeben werden.

Um z. B. alle Abhängigkeiten auf dem Bildschirm anzuzeigen anstatt in die Konfigurationsdatei zu schreiben, kann folgender Befehl verwendet werden.

```
enterprise:/etc/rc.d # depmod -an
/lib/modules/2.4.10-4GB/kernel/abi/cxenix/abi-cxenix.o:
/lib/modules/2.4.10-4GB/kernel/abi/svr4/abi-svr4.o \
        /lib/modules/2.4.10-4GB/kernel/arch/i386/kernel/abi-machdep.o

/lib/modules/2.4.10-4GB/kernel/abi/ibcs/abi-ibcs.o:
/lib/modules/2.4.10-4GB/kernel/abi/svr4/abi-svr4.o \
        /lib/modules/2.4.10-4GB/kernel/arch/i386/kernel/abi-machdep.o

/lib/modules/2.4.10-4GB/kernel/abi/sco/abi-sco.o:
/lib/modules/2.4.10-4GB/kernel/abi/svr4/abi-svr4.o \
        /lib/modules/2.4.10-4GB/kernel/arch/i386/kernel/abi-machdep.o \
```

```
        /lib/modules/2.4.10-4GB/kernel/abi/cxenix/abi-cxenix.o
...
```

14.5.9 modules.conf oder conf.modules

Die Datei /etc/modules.conf oder /etc/conf.modules, der Name hängt ab von der entsprechenden Distribution, enthält die Liste der Module mit ihren Parametern. Dabei handelt es sich z. B. bei Modulen für Hardware um Parameter wie I/O-Adressen und Interrupts. Das Verhalten des Tools modprobe wird durch diese Einstellungen kontrolliert und gesteuert.

Ein Ausschnitt aus der /etc/modules.conf könnte so aussehen.

```
alias parport_lowlevel    parport_pc
options parport_pc io=0x378 irq=none,none

options ne      io=0x300

alias block-major-1       rd
alias block-major-2       floppy

ptions bttv    pll=1 radio=0 card=0
post-install bttv /sbin/modprobe "-k" tuner;

options dummy0 -o dummy0
options dummy1 -o dummy1

# ppp over ethernet
# the kernel 2.2 uses pppox
# the kernel 2.4 uses pppoe
if ‘kernelversion‘ == "2.2"
alias char-major-144      pppox
post-install pppox insmod mssclampfw
pre-remove   pppox rmmod  mssclampfw
else
alias char-major-108      ppp_async
alias char-major-144      pppoe
alias net-pf-24           pppoe
endif

# agpgart is i386 only right now
pre-install mga /sbin/modprobe "-k" "agpgart"
pre-install r128 /sbin/modprobe "-k" "agpgart"
pre-install radeon /sbin/modprobe "-k" "agpgart"
options agpgart agp_try_unsupported=1
```

Die folgenden Informationen finden Sie in der /etc/modules.conf:

Kommentare
: Leere Zeilen und Zeilen mit einem # am Anfang werden ignoriert.

keep
: Wenn der keep Parameter vor den path-Anweisungen zu finden ist, dann wird der Standardpfad übernommen und zu jeden spezifizierten Pfaden hinzugefügt.

depfile=ABSOLUTER_PFAD
: Dieser Eintrag überschreibt die Standardeinstellung für den Ort der Datei für die Modulabhängigkeiten.

options MODUL OPT1=WERT1 OPT2=WERT2 ...
: Mit dieser Zeile können Optionen für die Module mitgegeben werden. MODUL bezeichnet das einzelne Modul ohne die Endung .so. Die Optionen werden immer als Paar Name=Wert angegeben.

`alias`
: Aliase werden dazu benutzt generische Namen speziellen Modulen zuzuordnen.

`pre-install MODUL KOMMANDO`
: Diese Anweisung führt dazu, daß vor dem Einfügen eines Moduls der entsprechende Shell-Befehl ausgeführt wird.

`install MODUL KOMMANDO`
: Hiermit kann der vorgesehene Befehl zum Einbinden des Moduls durch einen Shell-Befehl ersetzt werden.

`post-install MODUL KOMMANDO`
: Diese Anweisung führt dazu, daß nach dem Einfügen eines Moduls der entsprechende Shell-Befehl ausgeführt wird.

`pre-remove MODUL KOMMANDO`
: Diese Anweisung führt dazu, daß vor dem Entfernen eines Moduls der entsprechende Shell-Befehl ausgeführt wird.

`remove MODUL KOMMANDO`
: Hiermit kann der vorgesehene Befehl zum Entfernen des Moduls durch einen Shell-Befehl ersetzt werden.

`post-remove MODUL KOMMANDO`
: Diese Anweisung führt dazu, daß nach dem Entfernen eines Moduls der entsprechende Shell-Befehl ausgeführt wird.

14.6 Kernelkompilierung

In der heutigen Zeit sind die mit den jeweiligen Distributionen mitgelieferten Kernels, dank der individuell integrierbaren Modulen, so universell, daß der Normalanwender keine speziellen Kernel für sich mehr bauen muß. Welche Gründe gibt es also noch heute einen Kernel selbst zu kompilieren.

1. Sie wollen Ihren Bekannten mit Ihrem Insiderwissen, “Ich habe meinen Kernel selbst kompiliert!”, beeindrucken.[4]

2. Ihre neue Hardware wird nicht von Ihrem alten Kernel unterstützt.

3. Ihre neue Software benötigt neue Funktionalitäten, die der alte Kernel nicht zur Verfügung stellt.

4. Der alte Kernel ist instabil und/oder enthält Sicherheitslücken.

5. Sie wollen einen optimierten Kernel auf Rechner mit speziellen Funktionen wie Router, Druckserver, Firewall u.s.w einsetzen.

6. Sie sind einfach neugierig, was der neue Kernel kann oder wie das Kernelkompilieren geht.

Um zu erfahren, welche Hardware vom Kernel unterstützt wird, lesen Sie das Hardware-HOWTO, schauen Sie in die `config.in` Dateien in den Kernelquellen oder sehen Sie sich die Ausgaben des Befehls `make config` an.

Bevor Sie mit der Kompilierung des Kernels beginnen, sollten Sie sich die Struktur ihres Dateisystems noch einmal in Erinnerung rufen, damit Sie im Notfall Ihr System mit einer Rettungsdiskette wieder zum Laufen bringen können. Dazu können Sie entweder den Befehl `mount` (10.2.1) oder die Datei `/etc/fstab` (10.2.4) verwenden.

[4] Das klappt leider nicht immer, da die meisten Menschen mit dem Begriff Kernel nichts anfangen können.

14.6.1 Kernelquellen

Um einen Kernel kompilieren zu können, brauchen Sie den Quellcode. Bei der SuSE-Distribution sind zwei Pakete vorhanden. Das Paket `kernel-source` enthält die Quellen des veränderten SuSE-Kernels, während das Paket `linux` den Quellcode des Originalkernel enthält.[5]

Die Quellen des neuesten Kernels können über Anonymous-FTP von `ftp.funet.fi` bezogen werden; sie befinden sich dort unterhalb des Verzeichnisses `/pub/Linux/PEOPLE/Linus`. Dieser Server wird an vielen Stellen gespiegelt, es lohnt sich also, zunächst mal auf den lokalen Spiegel-Servern nachzusehen. Einige nationale und internationale Spiegel-Server sind:

Deutschland
`ftp.tu-dresden.de/Linux/sunsite/`
`ftp.gwdg.de/pub/linux/mirrors/sunsite/`
`ftp.rz.uni-karlsruhe.de/pub/sunsite/`
`ftp.ba-mannheim.de/pub/linux/mirror.sunsite/`
`ftp.uni-rostock.de/Linux/sunsite/`
`ftp.rus.uni-stuttgart.de/pub/unix/systems/linux/MIRROR.sunsite/`
`ftp.uni-tuebingen.de/pub/linux/mirrors/ftp.metalab.unc.edu/pub/Linux/`

Österreich
`ftp.univie.ac.at:/systems/linux/sunsite/kernel`

USA
`metalab.unc.edu:/pub/Linux/kernel`
`tsx-11.mit.edu:/pub/linux/sources/system`

UK
`sunsite.doc.ic.ac.uk:/pub/unix/Linux/sunsite.unc-mirror/kernel`

Frankreich
`ftp.ibp.fr:/pub/linux/sources/system/patches`

Australien
`sunsite.anu.edu.au:/pub/linux/kernel`

Generell ist die Spiegelung von `metalab.unc.edu` ein guter Startpunkt. Die Datei `/pub/Linux/MIRRORS` enthält eine Liste mit den bekannten Spiegelungen.

Im angegebenen Verzeichnis befinden sich Unterverzeichnisse mit Namen wie v2.2, v2.4 usw. In den jeweiligen Verzeichnissen stehen dann die Linux-Quellen in Dateien mit den Namen `linux-x.y.z.tar.gz`. `x.y.z` ist dabei die Versionsnummer. Die Datei mit der höchsten Versionsnummer stellt den neuesten Kernel dar. In der Regel sollte dieses auch die beste Version sein.

Sie sollten aber eine schnelle Anbindung ans Internet besitzen oder eine Flatrate und etwas Geduld. In der Version `2.4.18` ist das Tarball-Archiv 28,7 MB groß.[6]

Um die Versionsnummer des aktuellen Kernels zu bestimmen, verwenden Sie den Befehl `uname`. Die Versionsnummer der Quellen können Sie der Datei `Makefile` entnehmen.

```
enterprise:/ # uname -sr
Linux 2.4.10-4GB
enterprise:/ # head -4 /usr/src/linux/Makefile
VERSION = 2
PATCHLEVEL = 4
SUBLEVEL = 18
EXTRAVERSION = Athlon
```

[5] Stand: SuSE 7.3

[6] Bei einer ISDN-Leitung mit 7 kB/s dauert der Download des Archivs ungefähr 70 Minuten. Mit T-DSL und einer Geschwindigkeit von 87 kB/s war das ganze schon in 6 Minuten erledigt.

Der Eintrag `EXTRAVERSION` ist normalerweise leer. Hier können Sie für Ihre spezielle Kernelvariante einen Namen vergeben (z. B. Athlon für einen für diesen Prozessortyp optimierten Kernel).

Die Quellen müssen sich im Verzeichnis `/usr/src/linux` befinden. Wenn Sie mit mehreren Kernelversionen parallel arbeiten, bietet es sich an, die jeweiligen Quellen in individuellen Verzeichnissen zu speichern und `/usr/src/linux` als symbolischen Link auf die jeweiligen Verzeichnisse zeigen zu lassen.

```
enterprise:/usr/src # ls -Gl
insgesamt 2
lrwxrwxrwx    1 root          18 Mär 20 15:18 linux -> linux-2.4.18
drwxr-xr-x   15 root         530 Mär 19 10:58 linux-2.4.10.SuSE
drwxrwxrwx    3 root          91 Mär 20 15:18 linux-2.4.18
```

Neben Linus Thorvald, der den Anstoß zu Linux gab, haben noch viele Menschen am Kernel mitgewirkt. Eine Liste dieser Personen mit ihren Beiträgen finden Sie in der Datei `CREDITS` in den Kernelquellen.

14.6.2 Konfiguration

Um die Elemente des Kernels festzulegen, muß erst eine Konfigurationsdatei `.config` angelegt werden. Dazu wechseln Sie in das Verzeichnis `/usr/src/linux` und benutzen eines der folgenden Tools.

Die einzelnen Tools werden durch Einträge in der Datei `Makefile` gestartet, wie hier in einem Auszug dieser Datei zu sehen ist.

```
xconfig: symlinks
        $(MAKE) -C scripts kconfig.tk
        wish -f scripts/kconfig.tk

menuconfig: include/linux/version.h symlinks
        $(MAKE) -C scripts/lxdialog all
        $(CONFIG_SHELL) scripts/Menuconfig arch/$(ARCH)/config.in

config: symlinks
        $(CONFIG_SHELL) scripts/Configure arch/$(ARCH)/config.in
```

`make config`

Der Befehl `make config` startet ein sehr einfaches Tool, daß Sie nacheinander die einzelnen Elemente abfragt. Dies ist vor allem dann sehr ärgerlich, wenn Sie nur ein Element ändern wollen und dann die ganze Prozedur durcharbeiten müssen.

```
enterprise:/usr/src/linux # make config
rm -f include/asm
( cd include ; ln -sf asm-i386 asm)
/bin/sh scripts/Configure arch/i386/config.in
#
# Using defaults found in arch/i386/defconfig
#
*
* Code maturity level options
*
Prompt for development and/or incomplete code/drivers (CONFIG_EXPERIMENTAL) [N/y/?] N
*
* Loadable module support
*
Enable loadable module support (CONFIG_MODULES) [Y/n/?]
```

`make menuconfig`

Dieses Tool ist ebenfalls für die Textkonsole gedacht. Sie können aber die einzelnen Punkte über ein Menü auswählen. So ist es möglich einzelne Punkte ohne großen Aufwand zu ändern. Das Tool funktioniert aber nur, wenn die ncurses-Bibliotheken installiert sind.

`make xconfig`

Dieses Tool ist sehr anspruchsvoll. Es läuft nur unter X und erwartet auch, daß Tcl/Tk installiert ist. Dafür erfolgt hier die Konfiguration sehr angenehm und mit Mausklicks.

Keines der Tools bewahrt Sie davor Lesen zu müssen, was die einzelnen Punkte bedeuten. Ich werde hier in diesem Skript nicht auf die einzelnen Elemente des Kernels eingehen. Dazu konsultieren Sie bitte das Kernel-HOWTO.

`make oldconfig`

Falls vorher schonmal eine gültige Konfigurationsdatei erzeugt wurde, kann man mit diesem Tool automatisch die alten Einstellungen benutzt werden und nur bei neu hinzugekommenen Einstellungen wird der Nutzer wie in `make config` gefragt.

`make cloneconfig`

Mit diesem Befehl ist es möglich aus dem laufenden Kernel eine Konfigurationsdatei zu erstellen und nur bei neu hinzugekommenen Einstellungen wird der Nutzer wie in `make config` gefragt. Dies ermöglicht den aktuellen Kernel als Vorlage zu nehmen und dann mit den anderen Konfigurationsbefehlen (z.B. `make menuconfig`) diese Einzuladen und nur die benötigten Sachen zu ändern.

14.6.3 Aufräumen, Kompilieren, Installieren

Aufräumen

Vor dem eigentlichen Kompilieren müssen noch die wechselseitigen Abhängigkeiten der Quell- und Include-Dateien festgestellt werden. Der Befehl

`make dep`

erledigt diese Aufgaben. Danach sollten Sie die alten Objektdateien der letzten Kompilierungsrunde auch noch löschen.

`make clean`

Sie können die Befehle auch gemeinsam eingeben. Das doppelte kaufmännische Und bewirkt, daß `make clean` nur dann ausgeführt wird, wenn `make dep` erfolgreich ausgeführt wurde.

`make dep && make clean`

Hier noch einmal ein Auszug aus dem Makefile zu den beiden oben genannten Befehlen.

```
dep-files: scripts/mkdep archdep include/linux/version.h
        scripts/mkdep -- init/*.c > .depend
        scripts/mkdep -- `find $(FINDHPATH) -name SCCS -prune -o -follow -name \
$(MAKE) $(patsubst %,_sfdep_%,$(SUBDIRS))
        $(MAKE) update-modverfile
endif
```

```
ifdef CONFIG_MODVERSIONS
MODVERFILE := $(TOPDIR)/include/linux/modversions.h
else
MODVERFILE :=
endif
export  MODVERFILE

depend dep: dep-files

clean:  archclean
        find . \( -name '*.[oas]' -o -name core -o -name '.*.flags' \) -type f - \
          | grep -v lxdialog/ | xargs rm -f
        rm -f $(CLEAN_FILES)
        rm -rf $(CLEAN_DIRS)
        $(MAKE) -C Documentation/DocBook clean
```

Kernel kompilieren

Und jetzt kommen wir zum eigentlichen Kompilieren. Der Befehl

```
make zImage
```

startet das Kompilieren des Kernels.

Soll der Kernel gleich auf einer Diskette untergebracht werden, so können Sie den Befehl

```
make zImage
```

verwenden. Dabei müssen Sie natürlich darauf achten, daß der Kernel nicht zu groß für die Diskette wird.

Apropos großer Kernel: Ab einer gewissen Größe funktioniert eine Kompilieren mit `make zImage` nicht mehr. Hier kommt der Befehl

```
make bzImage
```

zum Einsatz, der größere Kernels erlaubt.

Kernel installieren

Nach der Installation finden Sie den Kernel im Verzeichnis `/usr/src/linux/arch/i386/boot/`. Die fertige Datei heiß `zImage`. Diese muß in das Verzeichnis `/boot` kopiert werden und eventuell in `vmlinuz` umbenannt werden. Sicherheitshalber sollte man den alten Kernel nicht einfach löschen sondern umbenennen. Danach muß auf jeden Fall der Lilo neu geschrieben werden.

```
enterprise:/ # mv /boot/vmlinuz /boot/vmlinuz.old
enterprise:/ # mv /usr/src/linux/arch/i386/boot/zImage /boot/vmlinuz
enterprise:/ # lilo
```

Wer auf Nummer sicher gehen will, bindet beide Kernels als Bootimages in den Bootloader ein. So kann, falls der neue Kernel nicht funktioniert, mit dem alten Kernel ohne weitere Probleme gebootet werden.

Module

Enthält der Kernel Module, was sehr wahrscheinlich ist, so müssen auch diese erzeugt und installiert werden. Das Erstellen übernimmt der Befehl

```
make modules
```

und mit dem Befehl

`make modules_install`

werden die Moduldateien in das nach der Kernelversion benannte Verzeichnis in `/lib/modules` installiert.

Kernelkompilierung mit SuSE

Der Kernel der SuSE-Distribution ist nicht identisch mit dem von Linus Thorvald freigegebenen Kernel. SuSE hat Elemente des Kernels weiterentwickelt und an die Distribution angepaßt. Auf der Distribution sind daher auch zwei Pakete mit dem Kernelquellcode vorhanden. Das Paket *kernel-source* enthält die SuSE-Linux-Kernelquellen und das Paket *linux* enthält den original Linux-Kernelquellcode.

Die Konfigurationsdatei für den vom System bei der Installation installierten Kernel finden Sie im Verzeichnis `/boot` in der Datei `vmlinuz.config`. Sie können diese Datei z. B. beim Befehl `make menuconfig` importieren und damit die von SuSE vorgegebenen Einstellung als Grundlage für Ihren eigenen Kernel benutzen.

Den neuen SuSE-Kernel können Sie vom SuSE-FTP-Server herunterladen. Sie finden ihn unter der Rubrik `update`.

`ftp://ftp.suse.com/pub/suse/i386/update/`*Distributionsnummer*`/kernel/`*kernelversion*

Es ist aber anzuraten nicht vom Original-SuSE-Server die Dateien herunterzuladen, sondern einen auf der SuSE-Seite angegebenen Spiegelserver zu verwenden. Diese sind meist weniger belastet als der Originalserver und das spielt bei einer Downloadgröße von rund 30 MByte schon eine wichtige Rolle.

Das Paket wird mit dem RPM-Manager ausgepackt und erzeugt ein Verzeichnis mit der Versionsnummer und den Link `/usr/src/linux` auf dieses Verzeichnis.

```
enterprise:~/download # rpm -ivh kernel-source-2.4.16.SuSE-31.i386.rpm
kernel-source               ##################################################
enterprise:~/download # cd /usr/src
enterprise:/usr/src # ls -l linux*
lrwxrwxrwx    1 root     root           17 Mär 26 21:35 linux -> linux-2.4.16.SuSE
drwxr-xr-x   15 root     root          530 Mär 26 21:36 linux-2.4.16.SuSE
enterprise:/usr/src # cd linux
enterprise:/usr/src/linux # make menuconfig
enterprise:/usr/src/linux # ls -l .config
-rw-r--r--    1 root     root        38008 Mär 26 22:12 .config
enterprise:/usr/src/linux # make dep && make clean
.
. (Eine Menge Meldungen)
.
enterprise:/usr/src/linux # make bzImage && make modules && make modules_install
.
. (Eine Menge Meldungen)
.
enterprise:/usr/src/linux # ls -ld /lib/modules/2.4.16-4GB
drwxr-xr-x    5 root     root          313 Mär 26 22:37 /lib/modules/2.4.16-4GB
enterprise:/usr/src/linux # cd /boot
enterprise:/boot # cp /usr/src/linux/arch/i386/boot/bzImage vmlinuz-2.4.16
enterprise:/boot # vi /etc/lilo.conf
enterprise:/boot # lilo
```

Mit dem `vi` wurde in die `/etc/lilo.conf` folgender Block eingefügt:

```
  image  = /boot/vmlinuz-2.4.16
  label  = linux
  root   = /dev/hdb7
  initrd = /boot/initrd
  append = "enableapic vga=0x0317"
```

Notizen:

XVII	Arbeitsblatt Programminstallation und Kernelkompilierung	17.1

1 Welche Bibliotheken benötigt das Programm `passwd`?

2 Bei der Standardinstallation der SuSE-Distribution wird Java nur in der Version 1.1.8 installiert. Laden Sie aus dem Internet (`http://www.javasoft.com`) oder Intranet die neueste Version des Java-Development-Kits als RPM-Paket herunter und installieren Sie das Paket.

3 Für die Ausführung der Programme ist die "virtuelle Kaffeemaschine" `java` verantworlich. Sie befindet sich im Unterverzeichnis `bin` des Java-Verzeichnis. Überprüfen Sie, ob nun die richtige Java-Version läuft. Führen Sie als *normaler* Benutzer den Befehl `java -version` aus. Welche Information bekommen Sie?

4 Die Version ist eindeutig nicht richtig. Woran kann es liegen?

5 Wo befindet sich die Datei, die durch die Eingabe von `java` gestartet wurde.

6 Aus welchem Paket stammt Sie?

7 Finden Sie heraus, wohin die Datei `java` des RPM-Paket installiert worden ist.

8 Damit nun der "richtige" richtige Java-Interpreter aufgerufen werden kann, muß der Soft-Link `/usr/lib/linux` auf das richtige Paket umgelenkt werden.

9 Lassen Sie sich eine alphabetisch sortierte Liste aller installierten Pakete ausgeben.

10 Wieviele Pakete sind installiert?

11 Welchen Kernel verwendet Ihr System?

12 Lassen Sie sich eine Liste aller geladenen Module ausgeben.

13 Laden Sie die RPM-Pakete des aktuellen SuSE-Kernels aus dem Internet bzw. Intranet oder installieren Sie sie von der Distributions-CD/DVD.

14 Erstellen Sie eine neue Konfiguration auf Basis des aktuellen Kernels und optimieren Sie den Kernel für Ihren Prozessortyp.

15 Tragen Sie im `Makefile` bei `EXTRAVERSION` den Prozessornamen ein. Kompilieren Sie auf der Basis der neuen Konfiguration einen neuen Kernel und dessen Module.

16 Installieren Sie den Kernel als `vmlinuz-VERSION-PATCHLEVEL-SUBLEVEL-EXTRAVERSION` im Verzeichnis `/boot` sowie die Module ins entsprechende Verzeichnis.

17 Tragen Sie für den neuen Kernel einen Eintrag in die `lilo.conf` ein und installieren Sie den neuen Bootloader.

18 Testen Sie das neue System und vergleichen Sie es mit dem alten System.

19 Welche Dateisysteme unterstützt der aktuelle SuSE-Kernel als Modul oder fest integriert? Beschreiben Sie mit einem Satz das Einsatzgebiet des Dateisystems.

Notizen:

Kapitel 15

Shell-Scripting

Es ist wichtig für einen Administrator sich mit dem Shell-Scripting auszukennen. Shell-Skripte erlauben eine automatisierte Ausführung von häufig vorkommenden Vorgängen. So können Prozeduren, die umständlich und langwierig sind, in der Ausführung vereinfacht werden.

Wie solche Skripte programmiert werden, hängt von der verwendeten Shell ab. In diesem Kapitel wird das Skripting am Beispiel der Bash erklärt. Die Grundprinzipien sind aber für alle Shells bzw. für die meisten anderen Programmiersprachen gleich. Wer schon in einer anderen Programmiersprache programmiert hat, wird sicherlich keine Probleme mit diesem Kapitel haben. Es gibt einige Eigentümlichkeiten, an die man sich aber schnell gewöhnt.

Auf jeden Fall werden Sie in diesem Kapitel nicht programmieren lernen. Das erfordert ein wesentlich längeres Studium der Materie, als was in diesen wenigen Seiten abgehandelt werden könnte. Es wird Ihnen hier nur die Syntax, die Struktur und die Regeln, der Skript-Programmierung mit der Bash gezeigt. Programmieren umfaßt eine ganze Menge mehr.

15.1 Variablen, Aliase und Funktionen

Den Anfang bilden die Speichermöglichkeiten der Shell. In den Variablen können Werte gespeichert werden. Aliase und Funktionen werden dazu benutzt um einzelne oder mehrere Befehle zu speichern.

15.1.1 Variablen

Auf den Einsatz von Shell-Variablen bin ich schon in einem früheren Kapitel eingegangen. Trotzdem möchte ich an dieser Stelle noch einmal auf ihre Verwendung eingehen.

In Variablen können Werte gespeichert werden. Diese Variablen behalten ihren Wert aber nur temporär. Nach einem Logout oder spätestens nach einem Neustart ist der Wert nicht mehr existent. Umgebungsvariablen müssen daher bei jedem Start oder bei jedem Einloggen neu initialisert werden. Dies erfolgt in den Startskripten. Das Setzen einer Variablen ist einfach. Geben Sie nur den Namen der Variablen gefolgt von dem Gleichheitszeichen und dem Wert an.

```
ole@defiant:~> sinn=42
ole@defiant:~> nachricht="Hallo Welt"
```

Übrigens entspricht das Gleichheitszeichen hier nicht dem Zeichen, was Sie aus der Mathematik kennen. Es bedeutet nicht, daß die Variable gleich dem Wert ist, sondern daß der der Variablen der Wert zugewiesen wird. Es handelt sich also um einen sogenannten Zuweisungsoperator.

Was ist aber ein Operator? Ein Operator ist einfach ausgedrückt eine Regel, wie zwei Werte miteinander verknüpft werden sollen. Einfache Operatoren sind z. B. Addition (+), Subtraktion (-), Multiplikation (*) und Division (/).

Achten Sie bitte darauf, daß keine Leerzeichen zwischen der Variablen, dem Gleichheitszeichen und dem Wert steht. Sollte ein Leerzeichen im Wert stehen, so muß der Wert in einfache oder doppelte Anführungszeichen gesetzt werden.

```
ole@defiant:~> sinn = 42
bash: sinn: command not found
```

Wenn Sie mit dem Wert der Variablen arbeiten wollen, dann stellen Sie ein Dollarzeichen (`$`) vor dem Variablennamen. Dann können Sie sich z. B. mit dem Befehl `echo` (4.6.5) den Inhalt auf dem Bildschirm ausgeben lassen.

```
ole@defiant:~> echo $sinn
42
ole@defiant:~> echo $nachricht
Hallo Welt
```

Allerdings sind die beiden Variablen nur in dieser Shell lokal vorhanden. Um Sie auch Kindershells und anderen Programmen zugänglich zu machen, müssen Sie exportiert werden. Dies geschieht mit Hilfe des Programms `export` (5.2.2). Das folgende Beispiel zeigt den Unterschied.

```
ole@defiant:~> bash
ole@defiant:~> echo $sinn $nachricht

ole@defiant:~>
exit
ole@defiant:~> export sinn
ole@defiant:~> export nachricht
ole@defiant:~> bash
ole@defiant:~> echo $sinn $nachricht
42 Hallo Welt
ole@defiant:~> exit
exit
```

Mit dem Kommando `bash` wird eine neue Shell, in dem Fall die Bash, gestartet. Die Variablen `sinn` und `nachricht` haben keinen Inhalt in dieser Shell. Mit `exit` wird die neue Shell wieder beendet. In der alten Shell werden dann die Variablen exportiert. Jetzt stehen die Variablen auch in der neuen Shell, die aus der alten Shell gestartet wurde, zur Verfügung.

15.1.2 Aliase

Auch auf Aliase wurde in diesem Skript schon einmal eingegangen. Trotzdem auch hier noch eine kurze Wiederholung.

Aliase ermöglichen es neue Befehle zu definieren oder einem alten Befehl eine neue Bedeutung zu geben. Verantwortlich dafür ist der Befehl `alias` (5.2.8).

```
ole@defiant:~> alias more=less
ole@defiant:~> alias ls
alias ls='ls $LS_OPTIONS'
ole@defiant:~> echo $LS_OPTIONS
-N --color=tty -T 0
```

Der erste Befehl sorgt dafür, daß beim Aufruf von `more` das Programm `less` gestartet wird. Bei SuSE sind einige Befehle auch durch Aliase verändert worden. Dies kann man schön beim Befehl `ls` sehen, wo noch zusätzlich eine Umgebungsvariable ins Spiel kommt.

Bei der Definition komplexerer Befehle sollte der Ausdruck in Anführungszeichen gesetzt werden. So ergibt das Semikolon im unteren Beispiel nicht das erwartete Ergebnis.

```
ole@defiant:~> alias heute=date;cal
    November 2002
So Mo Di Mi Do Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30

ole@defiant:~> alias heute="date;cal"
ole@defiant:~> heute
Mit Nov 20 11:10:40 CET 2002
    November 2002
So Mo Di Mi Do Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```

15.1.3 Funktionen

Ergänzend zu den Aliase können Sie mit Funktionen neue Befehle definieren. Während Sie mit Aliase nur einfache Befehle definieren, sind Funktionen für komplexere Aufgaben, die normalerweise aus mehreren Befehlen bestehen, ausgelegt.

Die Syntax für die Erstellung einer Funktion lautet:

```
[function] NAME () { KOMMANDOLISTE; }
```

Diese Deklaration definiert eine Funktion NAME. Das Wort `function` ist nicht notwendig. Sollten Sie das Schlüsselwort weglassen, sind die runden Klammern nach dem Funktionsnamen Pflicht. In den geschweiften Klammern werden dann die Kommandos entweder durch Semikola oder durch das neue Zeile Zeichen getrennt.

```
ole@defiant:~> werda () { echo Du bist $USER; who; }
ole@defiant:~> werda
Du bist ole
ole      :0       Nov 20 10:03 (console)
ole      pts/0    Nov 20 10:04
tapico   pts/1    Nov 20 10:14
walter   pts/2    Nov 20 10:53
```

Wichtig ist es hier, daß Sie nicht das letzte Semikolon vergessen. Auch der letzte Befehl vor der schließenden geschweiften Klammer muß mit einem Semikolon beendet werden.

Das Schreiben der Befehle in einer Zeile ist bei längeren Befehlsfolgen etwas unpraktisch. Sie können anstatt eines Semikolon auch einen Zeilenumbruch machen, um die Kommandoliste einzugeben.

```
ole@defiant:~> function hallo () {
> echo Hallo $USER
> date +"Es ist %H:%M:%S Uhr"
> }
ole@defiant:~> hallo
Hallo ole
Es ist 11:49:26 Uhr
```

Bis jetzt wurden die Funktionen wie Aliase verwendet. Im Gegensatz zu diesen können Funktionen Parameter mitgegeben werden. Diese Parameter stehen dann in den Variablen `$1`, `$2`, `$3` bis `$N`. Die Anzahl der übergebenden Parameter steht in der Variablen `$#`.

```
ole@defiant:~> function sagmal () {
> echo Es wurden $# Parameter eingegeben.
> echo Der erste Parameter lautet: $1
> }
ole@defiant:~> sagmal Hallo Welt
Es wurden 2 Parameter eingegeben.
Der erste Parameter lautet: Hallo
ole@defiant:~> sagmal "Hallo Welt"
Es wurden 1 Parameter eingegeben.
Der erste Parameter lautet: Hallo Welt
```

Sie können im Beispiel deutlich die Auswirkungen der Anführungszeichen sehen.

```
ole@defiant:~> function werist () {
> finger $1
> ps -aux | grep $1
> }
ole@defiant:~> werist root
Login: root                          Name: root
Directory: /root                     Shell: /bin/bash
Last login Wed Oct  9 08:13 (CEST) on tty1
New mail received Thu Oct 10 11:20 2002 (CEST)
     Unread since Fri May  3 18:15 2002 (CEST)
No Plan.
root         1  0.0  0.0   448   64 ?       S    08:14   0:04 init [5]
root         2  0.0  0.0     0    0 ?       SW   08:14   0:00 [keventd]
root         3  0.0  0.0     0    0 ?       SW   08:14   0:00 [kapmd]
root         4  0.0  0.0     0    0 ?       SWN  08:14   0:00 [ksoftirqd_CPU0]
root         5  0.0  0.0     0    0 ?       SW   08:14   0:00 [kswapd]
...
```

15.2 Skripte

So schön Aliase und Funktionen auch sind. Sie haben einen Nachteil. Sie müssen vor der Benutzung erst initialisiert werden. Damit Sie immer zur Verfügung stehen, müssen Sie in den Start- bzw. Login-Dateien definiert werden. Auch sind Sie für größere Projekte nicht so geeignet. Die Kommandos können auch in normalen Textdateien abgelegt werden. Andere Namen für Shell-Skripte in Dateien sind Batch-Dateien oder Stapelverarbeitungsdateien.

Als Beispiel nehmen wir mal die Funktion `werist` aus dem obigen Abschnitt und schreiben die Kommandos in die Datei `werist`. Textabschnitte, die mit einem Schweinegatter `#` beginnen, werden bis zum Ende der Zeile ignoriert. Es handelt sich dabei um Kommentare.

Listing 15.1 *werist*

```
# werda - Zeigt Daten über den Benutzer und seine Prozesse an
#       werda BENUTZERNAME

finger $1
ps -aux | grep $1
```

15.2.1 Ausführen von Skripten

Um Skripte auszuführen gibt es mehrere Methoden. Als erste Methode kann der Bash-Befehl `source` verwendet werden. Er liest die angegebene Datei und führt die in ihr enthaltenen Kommandos aus.

```
ole@defiant:~/skripte> source ./werist ole
```

Die Angabe `./werist` sagt, daß die Datei in dem aktuellen Arbeitsverzeichnis ist. Um dies zu umgehen, können Sie das aktuelle Verzeichnis, dargestellt durch den Punkt "`.`", in den Pfad einfügen.

```
ole@defiant:~/skripte> echo $PATH
/usr/local/bin:/usr/bin:/usr/X11R6/bin:/bin:/usr/games:/opt/gnome/bin:/opt/kde3/bin:
/opt/kde2/bin:/usr/lib/java/bin:/opt/gnome/bin:.
```

Achten Sie bitte darauf, daß das aktuelle Verzeichnis am Ende der Liste steht. Für den Systemadministrator *root* tragen Sie bitte **nie** das aktuelle Verzeichnis in den Pfad ein.

Es gibt auch eine kürzere Variante um ein Skript zu starten. Verwenden Sie einfach anstatt von `source` den Punkt.

```
ole@defiant:~/skripte> . ./werist ole
```

Sie können das Skript auch starten, indem Sie eine neue Instanz der Bash aufrufen.

```
ole@defiant:~/skripte> /bin/bash ./werist ole
```

Diese Methode verhält sich aber anders als die vorher vorgestellten Methoden. Es wird explizit eine neue Shell gestartet, in der dann das Skript ausgeführt wird. Das hat natürlich zur Folge, daß Variablen, die nicht exportiert worden sind, in dieser Shell nicht zur Verfügung stehen.

Ausführbare Datei

Sie kommen aber auch ohne einen zusätzlichen Befehl aus. Machen Sie einfach aus der normalen Textdatei eine ausführbare Datei. Unter Linux ist alles ausführbar, wenn es einen Inhalt besitzt, der durch den Prozessor (nativer Code) oder durch ein anderes Programm (interpretierter Code), wie z. B. die Shell, ausgeführt werden kann.

Um eine Datei ausführbar zu machen, müssen Sie das X-Recht setzen.

```
ole@defiant:~/skripte> ls -l werist
-rw-r--r--    1 ole      users         114 Nov 20 12:48 werist
ole@defiant:~/skripte> chmod a+x werist
ole@defiant:~/skripte> ls -l werist
-rwxr-xr-x    1 ole      users         114 Nov 20 12:48 werist
```

Jetzt können alle (Besitzer, Gruppe und der Rest der Welt) dieses Skript ausführen, indem Sie einfach den Namen eingeben.

```
ole@defiant:~/skripte> werist ole
Login: ole                              Name: Ole Vanhoefer
Directory: /home/ole                    Shell: /bin/bash
On since Wed Nov 20 10:03 (CET) on :0,    idle 200 days 20:28, from console
On since Wed Nov 20 10:04 (CET) on pts/0, idle 5:09
On since Wed Nov 20 10:14 (CET) on pts/1 (messages off)
On since Wed Nov 20 10:53 (CET) on pts/2, idle 2:38 (messages off)
On since Wed Nov 20 12:45 (CET) on pts/3, idle 2:19 (messages off)
New mail received Thu Oct 10 11:20 2002 (CEST)
     Unread since Mon Jul 15 21:28 2002 (CEST)
```

She-Bang	Shell
`#!/bin/sh`	Die Bourne Shell
`#!/bin/bash`	Die Bourne Again Shell
`#!/bin/ksh`	Die Korn-Shell
`#!/bin/csh`	Die C-Shell
`#!/bin/tcsh`	Die erweiterte C-Shell
`#!/bin/sed`	Der Streaming-Editor
`#!/usr/bin/awk`	Die Programmiersprache awk
`#!/usr/bin/perl`	Die Programmiersprache Perl

Tabelle 15.1: Eine Auswahl von häufigen Interpretern

```
No Plan.
root      2049  0.0  1.4  3408 1776 ?        S    10:03   0:00 /usr/X11R6/bin/xconsole -notify
ole       2058  0.0  0.0  2560    0 ?        SW   10:03   0:00 /bin/sh /usr/X11R6/bin/kde
ole       2106  0.0  0.7 19560  924 ?        S    10:03   0:00 kdeinit: Running...
ole       2109  0.0  1.2 19544 1544 ?        S    10:03   0:00 kdeinit: dcopserver --nosid
ole       2112  0.0  2.2 21824 2788 ?        S    10:03   0:00 kdeinit: klauncher
...
```

She-Bang!

Es gibt einen ganzen Haufen von Skripten auf einem Linux-System und es gibt viele verschieden Sprachen, in denen sie geschrieben werden können. Wird der Befehl `source`, der Punkt "`.`" oder eine ausführbare Datei benutzt, dann werden die Kommandos des Skripts in der aktuellen Shell ausgeführt. Bei der Benutzung einer seperaten Shell für die Ausführung des Skripts (`/bin/bash meinSkript`) kann die Umgebung selbst gewählt werden.

Es ist ohne Frage von Vorteil, wenn das Skript selber die Information enthalten würde, durch welchen Interpreter die enthaltenen Kommandos ausgeführt werden sollen. In der Bash ist dies durch die Zeichen "`#!`" realisiert, die am Anfang der ersten Zeile stehen. Diese Konstruktion wird umgangssprachlich *She-Bang* genannt. Diese Wortschöpfung setzt sich aus den Bezeichnungen *sheepgate* für das Doppelkreuz '`#`' und *bang* für das Ausrufungszeichen zusammen.

So beginnt ein Skript, daß für die Bash geschrieben wurde, mit der folgenden Zeile.

```
#!/bin/bash
```

Die Bash untersucht die erste Zeile des Skripts, startet den gefundenen Interpreter und übergibt das Skript an diesen Interpreter zur Ausführung.

Ein falsche She-Bang-Anweisung ist ein häufiger Grund für eine fehlerhafte Ausführung des Skripts. In diesem Fall meldet die Bash und nicht der Interpreter einen Fehler.

In dem im folgenden Beispiel gestarteten Skript wurde ein falscher Interpreter eingetragen. Die Fehlermeldung kommt von der Bash.

```
ole@enterprise:~/test> shebang
bash: ./shebang: bad interpreter: Datei oder Verzeichnis nicht gefunden
```

Die Umgebung

Wenn Sie ein Skript in einer neuen Shell ausführen, dann wird eine komplett neue Instanz dieser Shell erzeugt. Aus der Eltern-Shell werden alle Variablen übernommen, die exportiert worden sind. Außerdem werden die Konfigurationsskripte der jeweiligen Shell ausgeführt, so daß eventuell Einstellungen überschrieben werden können. Wenn Ihr Skript von einer Variablen abhängt, dann sorgen Sie dafür, daß sie entweder in der Shell-Konfigurationsdatei gesetzt wurde oder als Umgebungsvariable exportiert wurde. Bitte machen Sie nicht beides zur gleichen Zeit.

Eine grundlegende Unix-Regel besagt, daß Kinder-Prozesse die Variablen von Ihrem Eltern-Prozeß erben. Die Variablen des Kinder-Prozeß sind aber nur während seiner Ausführung gültig und werden nicht zur Eltern-Shell zurückgegeben. Also haben Variablenänderungen in diesem Prozeß keine Auswirkungen auf die Variablen im Eltern-Prozeß.

Dies Verhalten läßt sich am folgenden Beispiel nachvollziehen.

```
ole@enterprise:~/test> cat shebang
#!/bin/bash

echo $var
var="Neuer Wert"
echo $var
ole@enterprise:~/test> var="Alt"
ole@enterprise:~/test> export var
ole@enterprise:~/test> shebang
Alt
Neuer Wert
ole@enterprise:~/test> echo $var
Alt
```

15.2.2 Rückgabewerte

Für Shell-Skripte ist es sehr wichtig festzustellen, ob die Kommandos ihre Aufgabe erfolgreich abgeschlossen haben. Die meisten Kommandos geben einen *Rückgabewert* oder *Fehlercode* zurück. Dieser Wert ist eine simple Ganzzahl und seine Bedeutung hängt von dem verwendeten Programm ab. Bei allen Programmen bedeutet die Zahl '`0`', daß der Befehl erfolgreich durchgeführt wurde. Eine von Null verschiedene Zahl hingegen deutet auf ein Problem hin.

Der Fehlercode wird in der Variablen '`?`' gespeichert. Da sie für jedes Kommando neu gesetzt wird, muß sie direkt nach der Kommandoausführung ausgelesen werden. Sie können dies direkt am Prompt nachvollziehen.

```
ole@enterprise:~/test> cd next
bash: cd: next: Datei oder Verzeichnis nicht gefunden
ole@enterprise:~/test> echo $?
1
ole@enterprise:~/test> echo $?
0
```

Bei der ersten Ausgabe wird die fehlerhafte Ausführung des Kommandos `cd` durch die Zahl 1 angezeigt. Die zweite Ausgabe der Variablen '`?`' bezieht sich auf den Erfolg des ersten `echo` Befehls.

Befehle können aus mehreren Gründen sich erfolglos beenden. Viele Befehle besitzen daher auch mehr als einen Fehlercode. Schauen Sie sich doch die Verhaltensweise von `grep` (7.7.1) einmal an.

```
ole@enterprise:~/test> grep bash shebang
#!/bin/bash
ole@enterprise:~/test> echo $?
0
ole@enterprise:~/test> grep tapico shebang
ole@enterprise:~/test> echo $?
1
ole@enterprise:~/test> grep bash sheebang
grep: sheebang: Datei oder Verzeichnis nicht gefunden
ole@enterprise:~/test> echo $?
2
```

Wenn `grep` einen oder mehrere Treffer erzielt, dann wird der Fehlercode 0 ausgegeben. Ist der Befehl soweit richtig, aber `grep` findet keine passende Zeichenkette, dann kommt der Fehlercode 1. Kommt es zu einem richtigen Fehler, wie z. B. einer nichtexistenten Datei, dann wird der Fehlercode 2 zurückgegeben.

15.3 Grundstrukturen

Nach all der langen Vorrede geht es nun mit den ersten Skripten los. Dabei geht es erst mal um ganz simple Dinge.

15.3.1 Auswerten

In unserem ersten Beispiel geht es um ein Skript mit dem Namen `addiere`, daß zwei als Parameter eingegebene Zahlen miteinander addiert.

Listing 15.2 *addiere*[1]

```
 1: #!/bin/bash
 2:
 3: # Addiert zwei als Parameter angegebene Zahlen
 4: #   addiere ZAHL1 ZAHL2
 5:
 6: # Berechnen
 7: summe=$(( $1 + $2 ))
 8:
 9: # Ausgabe
10: echo "Die Summe aus $1 und $2 ist $summe."
```

Zeile 1 kennen Sie schon aus dem vorherigen Abschnitt. Der sogenannte She-Bang sorgt dafür, daß die folgenden Shell-Kommandos durch eine neue Instanz der Bash abgearbeitet werden. Das Schweinegatter ist in der Shell ein Kommentarzeichen. Alles von diesem Zeichen bis zum Ende der Zeile wird als Kommentar aufgefaßt. Skripte sollten Sie immer gut kommentieren, um auch später das Skript verstehen zu können.

In Zeile 7 werden die beiden Parameter 1 und 2 addiert und das Ergebnis in die Variable `summe` geschrieben. Dieser Vorgang wird als Arithmetische Ausdehnung (engl. *Arithmetic Expansion*) bezeichnet. Im Klartext heißt das nichts anderes, als daß der Ausdruck in den Klammern berechnet wird und das Ergebnis der Berechnung zurückgegeben wird. Das Format einer Arithmetischen Ausdehnung ist

```
$((AUSDRUCK))
```

Die Ausgabe des Ergebnis in Zeile 10 durch `echo` sollte Ihnen bekannt sein.

Das folgende Beispiel zeigt ein Skript, daß die Zeilen einer Datei durchnummeriert und das Ergebnis in einer HTML-Datei speichert.

Listing 15.3 *tohtml*

```
#!/bin/bash

# Gibt ein Skript als zeilennummerierte HTML-Datei zurück
#   tohtml DATEINAME

# Name der Ausgabedatei
out="$1.html"

# Lege den Anfang der HTML-Datei an
echo -e "<html>\n<head>\n<title>$1</title>\n</head>\n<body>\n" > $out
echo -e "<h1 align='center'>Listing: $1</h1>\n<pre>" >> $out

# Datei durchnummerieren
```

[1] Die Zahlen mit dem Doppelpunkt am Anfang der Zeile dienen nur zur Orientierung und sind nicht Bestandteil des Programms.

Operator	Bedeutung
`wert++`	Post-Inkrement der Variablen
`wert--`	Post-Dekrement der Variablen
`++wert`	Prä-Inkrement der Variablen
`--wert`	Prä-Dekrement der Variablen
`+ -`	Plus und Minus
`! ~`	Logische und Bitweise Negation
`* / %`	Mal, Teilen und Modulo
`<< >>`	linkes und rechtes bitweises Verschieben
`<= >= < >`	Vergleichsoperatoren
`== !=`	Gleich und Ungleich
`& ^ |`	Bitweises UND, exklusives ODER und ODER
`&& ||`	Logisches UND und ODER
`AUSDR1?AUSDR2:AUSDR3`	Wenn Dann Sonst Ausdruck

Tabelle 15.2: Arithmetische Operatoren

```
expand $1 | nl -w 4 -b a -s ": " >> $out

# Ende der HTML-Datei
echo -e "</pre>\n</body>\n</html>\n" >> $out
```

Einfach dem Skript die gewünschte Datei mitgeben und schon ist eine passende Datei entstanden.

```
ole@enterprise:~/test> tohtml addiere
ole@enterprise:~/test> ls -l addiere*
-rwxr-xr-x    1 ole      users         172 Nov 26 18:52 addiere
-rw-r--r--    1 ole      users         356 Nov 26 20:35 addiere.html
```

Natürlich ist dieses Beispiel sehr einfach und hat viele Schwachstellen. So würden die Spitzenklammern in der Datei einfach in die HTML-Datei kopiert und dort dann vom Browser als HTML-Tags interpretiert. Also müssen vorher möglichst alle Spitzen-Klammern in HTML-Sonderzeichen umgewandelt werden. Dies erledigt der Streaming-Editor `sed`.

Listing 15.4 *tohtml Version 2*

```
#!/bin/bash
...
# Datei durchnummerieren und spitze Klammern behandeln
sed "s/</\&lt;/g" $1 | sed "s/>/\&gt;/g" \
  | expand | nl -w 4 -b a -s ": " >> $out

# Ende der HTML-Datei
echo -e "</pre>\n</body>\n</html>\n" >> $out
```

15.3.2 test

Ein wichtiges Element der Programmierung ist die Möglichkeit sich je nach den gegebenen Fakten unterschiedlich verhalten zu können. Dazu müssen Entscheidungen gefällt werden, die auf wahren oder falschen Aussagen beruhen. Ein Instrument für die Ermittlung von wahren und falschen Aussagen ist der Befehl `test`.

```
test AUSDRUCK
[ AUSDRUCK ]
```

Der angegebene Ausdruck wird ausgewertet und ein entsprechender Fehlercode 0 für wahr und 1 für falsch wird ausgegeben. Der untere Befehl ist eine Kurzform von `test`. Die öffnende eckige Klammer entspricht dem Befehl `test`. Die schließende eckige Klammer terminiert den Ausdruck. Daher müssen vor und hinter der eckigen Klammer Leerzeichen stehen.

Ausdrücke

Der Befehl gibt für die folgenden Ausdrücke wahr zurück, wenn die folgenden Aussagen zutreffen.

`AUSDRUCK` : Der Ausdruck ist wahr.

`! AUSDRUCK` : Der Ausdruck ist falsch.

`AUSDRUCK1 -o AUSDRUCK2` : Einer der beiden Ausdrücke oder beide sind wahr.

`AUSDRUCK1 -a AUSDRUCK2` : Beide Ausdrücke sind wahr.

`-n STRING` : Die Zeichenkette umfaßt mindestens ein Zeichen.

`-z STRING` : Die Zeichenkette enthält keine Zeichen.

`STRING1 = STRING2` : Beide Zeichenketten sind gleich.

`STRING1 != STRING2` : Die Zeichenketten sind unterschiedlich.

`INTEGER1 -eq INTEGER2` : Die beiden Ganzzahlen sind gleich.

`INTEGER1 -ne INTEGER2` : Die beiden Ganzzahlen sind nicht gleich.

`INTEGER1 -gt INTEGER2` : Die erste Zahl ist größer als die Zweite.

`INTEGER1 -lt INTEGER2` : Die erste Zahl ist kleiner als die Zweite.

`INTEGER1 -ge INTEGER2` : Die erste Zahl ist größer oder gleich der Zweiten.

`INTEGER1 -le INTEGER2` : Die erste Zahl ist kleiner oder gleich der Zweiten.

`-e DATEI` : Die Datei existiert.

`-s DATEI` : Die Datei existiert und ist größer als 0 Byte.

`-d DATEI` : Die Datei existiert und ist ein Verzeichnis.

`-f DATEI` : Die Datei existiert und ist eine normale Datei.

`-b DATEI` : Die Datei existiert und ist ein Block-Gerät.

`-c DATEI` : Die Datei existiert und ist ein Zeichen-Gerät.

`-L DATEI` : Die Datei existiert und ist ein symbolischer Link.

`-p DATEI` : Die Datei existiert und ist eine Pipeline-Datei (FIFO).

`-S DATEI` : Die Datei existiert und ist ein Socket.

`-r DATEI` : Die Datei existiert und ist lesbar.

`-w DATEI` : Die Datei existiert und ist schreibbar.

`-x DATEI` : Die Datei existiert und ist ausführbar.

`-u DATEI` : Die Datei existiert und SUID ist gesetzt.

`-g DATEI` : Die Datei existiert und SGID ist gesetzt.

`-k DATEI` : Die Datei existiert und Sticky Bit ist gesetzt.

`-O DATEI` : Die Datei existiert und sein Besitzer ist gleich der effektiven UID.

`-G DATEI` : Die Datei existiert und seine Gruppe ist gleich der effektiven GID.

`DATE1 -ef DATEI2` : Beide Dateien besitzen die gleiche Geräte- oder Inode-Nummer.

`DATE1 -ot DATEI2` : Die Datei 1 ist älter als die Datei 2.

`DATE1 -nt DATEI2` : Die Datei 1 ist jünger als die Datei 2.

Mit den Shell-Logik-Operatoren `&&` und `||` können damit einfache Entscheidungen aufgebaut werden. Gerade bei der Ausführung von Skripten wird oft getestet, ob die Skripte überhaupt existieren.

Dieses Beispiel finden Sie in der `~/.bashrc`. Die Datei `~/.alias`, in der die Alias-Einträge eingetragen werden, wird nur ausgeführt, wenn Sie existiert und nicht leer ist.

```
test -s ~/.alias && . ~/.alias
```

Beispiel

Ein Verzeichnis mit vier Dateien dient als Übungsgebiet.

```
ole@enterprise:~/test> ls -l
insgesamt 8
-rw-r--r--    1 ole      users          96 Nov 26 22:14 hebong
-rw-r--r--    1 ole      users           0 Nov 26 22:14 leer
drwxr-xr-x    2 ole      users          35 Nov 26 22:13 ordner
-rwxr-xr-x    1 ole      users          96 Nov 25 22:40 shebang
```

Handelt es sich um Verzeichnisse oder nicht. Der Fehlercode verrät es.

```
ole@enterprise:~/test> test -d ordner; echo $?
0
ole@enterprise:~/test> test -d hebong; echo $?
1
```

Der Fehlercode kann auch direkt ausgewertet werden. Die beiden Operatoren `&&` und `||` machen die Ausführung des zweiten Befehls vom Testergebnis abhängig.

```
ole@enterprise:~/test> test -e leer && echo "Die Datei existiert."
Die Datei existiert.
ole@enterprise:~/test> test -s leer && echo "Die Datei ist nicht leer."
ole@enterprise:~/test> test shebang -ot hebong && echo "Älter"
Älter
```

Das ganze geht auch mit Zahlen.

```
ole@enterprise:~/test> [ 42 -gt 12 ] && echo "Richtig"
Richtig
ole@enterprise:~/test> [ 42 -lt 12 ] || echo "Falsch"
Falsch
```

Bei der Kurzschreibweise sollten Sie auf keinen Fall die Leerzeichen vergessen.

```
ole@enterprise:~/test> [ -x shebang] && echo "Ausführbar"
[: missing ‘]’
```

Der Unterschied zwischen `&&` und `||` ist deutlich sichtbar.

```
ole@enterprise:~/test> [ -x shebang ] && echo "Ausführbar"
Ausführbar
ole@enterprise:~/test> [ -x hebong ] && echo "Ausführbar"
ole@enterprise:~/test> [ -x hebong ] || echo "Nicht Ausführbar"
Nicht Ausführbar
```

15.3.3 if ... then ... elif ... then ... else ... fi

Mit dem Befehl `if` und seinen Unterbefehlen, kann ein Skript aufgrund der Fakten Entscheidungen treffen. Unser Skript `tohtml` hat noch ein paar Schwächen. So kommt es zu Fehlermeldungen von `sed`, wenn ein nichtvorhandene oder nicht lesbare Datei vorliegt. Deshalb soll in Zukunft vor der Ausführung überprüft werden, ob die angegebene Datei lesbar ist. Trifft dies nicht zu, wird eine Fehlermeldung ausgegeben.

Listing 15.5 *tohtml Version 3*

```
#!/bin/bash

# Gibt ein Skript als zeilennummerierte HTML-Datei zurück
#   tohtml SKRIPTNAME

# Existiert die Ausgabedatei ?
if [ -r $1 ]
then
  # Name der Ausgabedatei
  out="$1.html"

  # Lege den Anfang der HTML-Datei an
  echo -e "<html>\n<head>\n<title>$1</title>\n</head>\n<body>\n" > $out
  echo -e "<h1 align='center'>Listing: $1</h1>\n<pre>" >> $out

  # Datei durchnummerieren und spitze Klammern behandeln
  sed "s/</\&lt;/g" $1 | sed "s/>/\&gt;/g" \
    | expand | nl -w 4 -b a -s ": " >> $out

  # Ende der HTML-Datei
  echo -e "</pre>\n</body>\n</html>\n" >> $out
else
  # Existiert die Datei oder ist sie nur nicht lesbar?
  if [ -e $1 ]
  then
    # Fehlerausgabe auf Fehlerkanal
    echo "Die Datei $1 ist nicht lesbar." > /dev/stderr
    exit 1      # Beenden mit Fehlercode
  else
    # Fehlerausgabe auf Fehlerkanal
    echo "Die Datei $1 existiert nicht." > /dev/stderr
    exit 2      # Beenden mit Fehlercode
  fi
fi
```

Das Skript ist durch die Struktur `if ... else` in zwei Teile aufgeteilt. Der erste Teil wird ausgeführt, wenn die Datei lesbar ist. Wenn Sie nicht lesbar ist, wird der zweite Teil nach dem `else` ausgeführt.

Die Anweisungen nach dem `then` werden nur ausgeführt, wenn der Ausdruck hinter dem `if` wahr ist. Optional kann nach diesen Anweisungen noch ein `else` erscheinen, das einen zweiten Anweisungsblock einleitet. Diese Anweisungsblock wird nur dann ausgeführt, wenn der Ausdruck hinter dem `if` falsch ist.

Im zweiten Anweisungsblock wird noch unterschieden, ob die Datei nur nicht lesbar ist oder sie gar nicht existiert. Die Meldungen werden auf dem Fehlerausgabe ausgegeben, die normalerweise wie die Standardausgabe auf das ausführende Terminal zeigt. Danach wird das Skript mit unterschiedlichen Fehlercodes beendet. Dieses Verhalten können Sie im folgenden Beispiel sehen.

```
ole@enterprise:~/test> tohtml /etc/shadow ; echo $?
Die Datei /etc/shadow ist nicht lesbar.
1
ole@enterprise:~/test> tohtml blubb ; echo $?
Die Datei blubb existiert nicht.
2
ole@enterprise:~/test> tohtml shebang ; echo $?
0
ole@enterprise:~/test>
```

Weitere Entscheidungsblöcke mit eigenen Testausdrücken können vor dem `else` eingefügt werden, um noch feinere Entscheidungen treffen zu können. So liest das Skript, wenn kein Parameter eingegeben wurde, direkt von der Konsole seine Daten. Dies soll nicht geschehen, sondern in diesem Fall und wenn mehr als ein Parameter eingegeben wurde, soll das Skript einen Hilfetext ausgeben.

Listing 15.6 *tohtml Version 4*

```
#!/bin/bash

# Gibt ein Skript als zeilennummerierte HTML-Datei zurück
#   tohtml SKRIPTNAME

# Stimmt die Parameteranzahl nicht, dann Hilfetext
if [ $# -ne 1 ]
then
  echo "tohtml - Erzeugt von Dateien Listings im HTML-Format" > /dev/stderr
  echo "  Anwendung:" > /dev/stderr
  echo "      tohtml DATEINAME" > /dev/stderr
  exit 3        # Fehlercode ausgeben
# Existiert die Ausgabedatei ?
elif [ -r $1 ]
then
  # Name der Ausgabedatei
 ...
  echo -e "</pre>\n</body>\n</html>\n" >> $out
else
  # Existiert die Datei oder ist sie nur nicht lesbar?
 ...
  fi
fi
```

Das Skript ist jetzt in drei Teile aufgeteilt. Der erste Teil wird ausgeführt, wenn mehr oder weniger als ein Parameter angegeben wurde. Es wird ein Hilfetext ausgegeben. Der zweite Teil wird nur ausgeführt, wenn ein Parameter angegeben wurde *und* der Parameter eine lesbare Datei bezeichnet. Trifft dies auch nicht zu, dann wird erst der letzte Teil nach dem `else` ausgeführt und die Fehlermeldungen erscheinen. Im Listing sind die Anweisungsblöcke der Teile Zwei und Drei nicht angegeben, da sie mit dem vorherigen Skript identisch sind.

Sie können in eine `if` Struktur so viele `elif ... then ...` Anweisungen einbauen, wie sie wollen.

```
ole@enterprise:~/test> tohtml
tohtml - Erzeugt von Dateien Listings im HTML-Format
  Anwendung:
      tohtml DATEINAME
```

Noch einmal übersichtlich dargestellt sieht die ganze Konstruktion so aus, wobei die `elif ... then` beliebig oft vorkommen können.

```
if AUSDRUCK1
then
  ...
elif AUSDRUCK2
then
  ...
elif AUSDRUCK3
then
  ...
else
  ...
fi
```

15.3.4 case

Bei `case` wird ein Wert mit einer Reihe von Strings verglichen. Stimmen sie überein, so wird der entsprechende Abschnitt ausgeführt. Damit ist `case` ein Spezialfall von `if`. In den meisten Fällen wird `case` zur Ausführung von bestimmten Abschnitten eines Skripts verwendet. Insbesondere beim Starten, Neustarten und Stoppen von Diensten kommt es zum Einsatz. In den sogenannten RC-Skripten, die für das Starten und Stoppen von Dämonen und Serverdiensten verwendet werden, wird diese Konstruktion verwendet.

```
ole@enterprise:~> ls /sbin/rc*
/sbin/rcSuSEfirewall2  /sbin/rcgpm      /sbin/rcnetwork  /sbin/rcsyslog
/sbin/rcdhclient       /sbin/rchotplug  /sbin/rcportmap
ole@enterprise:~> ls /usr/sbin/rc*
/usr/sbin/rcalsasound     /usr/sbin/rckdm                /usr/sbin/rcsmbfs
/usr/sbin/rcapache        /usr/sbin/rcksysguardd         /usr/sbin/rcsmpppd
/usr/sbin/rcapid          /usr/sbin/rclpd                /usr/sbin/rcsnmpd
/usr/sbin/rcatd           /usr/sbin/rcmysql              /usr/sbin/rcsplash
/usr/sbin/rcautofs        /usr/sbin/rcnfs                /usr/sbin/rcsshd
/usr/sbin/rccron          /usr/sbin/rcnfsserver          /usr/sbin/rcxdm
/usr/sbin/rcfam           /usr/sbin/rcnscd               /usr/sbin/rcxfs
/usr/sbin/rcfbset         /usr/sbin/rcpcscd              /usr/sbin/rcxntpd
/usr/sbin/rci4l           /usr/sbin/rcpersonal-firewall  /usr/sbin/rcypbind
/usr/sbin/rci4l_hardware  /usr/sbin/rcpowerfail          /usr/sbin/rcyppasswdd
/usr/sbin/rcinetd         /usr/sbin/rcrandom             /usr/sbin/rcypserv
/usr/sbin/rcisdn          /usr/sbin/rcraw                /usr/sbin/rcypxfrd
/usr/sbin/rcjoystick      /usr/sbin/rcsendmail
/usr/sbin/rckbd           /usr/sbin/rcsingle
```

Das folgende Beispiel zeigt das Prinzip, das hinter dieser Idee steckt.

Listing 15.7 *dienst*

```
#!/bin/bash

# Steuert einen Dienst. Beispiel für case ... in Struktur
#   dienst start|restart|stop

# Nach dem Inhalt des Parameters $1 wird entschieden, was gemacht werden soll.
case $1 in
  start)
    # Der Dienst wird gestartet
```

```
    echo "Der Dienst wird gestartet."
    ;;

  restart)
    # Der Dienst wird neu gestartet
    echo "Der Dienst wird neu gestartet."
    ;;

  stop)
    # Der Dienst wird gestoppt
    echo "Der Dienst wird gestoppt."
    ;;

  *)
    # Falsches Kommando
    echo "Falsches Kommando." > /dev/stderr
    echo "Syntax:" > /dev/stderr
    echo "  dienst start|restart|stop" > /dev/stderr
    exit 1      # Beenden mit Fehlercode
    ;;
esac
```

Jenachdem ob `start`, `restart` oder `stop` hinter dem Skriptnamen steht, wird ein anderer Abschnitt des Skripts ausgeführt. Sollte es zu keiner Übereinstimmung kommen, wird der Abschnitt mit dem Asterisk ausgeführt, der in diesem Fall einen Hilfetext enthält.

```
ole@enterprise:~/test> dienst start
Der Dienst wird gestartet.
ole@enterprise:~/test> dienst restart
Der Dienst wird neu gestartet.
ole@enterprise:~/test> dienst stop
Der Dienst wird gestoppt.
ole@enterprise:~/test> dienst
Falsches Kommando.
Syntax:
  dienst start|restart|stop
```

Hier eine praktische Anwendung um zwei VNC-Server und einen Viewer für eine Vorführung zu starten und zu beenden. Weitere Informationen über den VNC-Server und seine Konfiguration finden Sie in Abschnitt B.1.

Listing 15.8 *vnc*

```
#!/bin/sh
# Startet zwei VNC-Server (Master und Slave)

# Auflösung festlegen
GEOM=950x680
CDEPTH=16

# Variable auswerten
case $1 in
  start)
    # Starten der VNC-Server
    # Master starten
    # Passworddatei ~/.vnc/privat
    vncserver :1 \
      -geometry $GEOM \
```

```
      -depth $CDEPTH \
      -alwaysshared \
      -name master \
      -rfbauth $HOME/.vnc/privat

    # Slave starten
    # Passworddatei ~/.vnc/public
    vncserver :2 \
      -geometry $GEOM \
      -depth $CDEPTH \
      -alwaysshared \
      -name slave \
      -rfbauth $HOME/.vnc/public

    # Viewer zur Kontrolle des Master-Servers starten
    # Falls in einem X-Terminal als anderer Benutzer gestartet zeigt
    # die Display-Variable auf das aktuelle X-Window
    # Der Server muß aber vorher mit
    # xhost localhost
    # freigegeben werden
    DISPLAY=:0.0
    export DISPLAY
    # Viewer starten
    vncviewer :1
    ;;

  stop)
    # Beenden des VNC-Servers
    vncserver -kill :1
    vncserver -kill :2
    ;;

  *)
    # Falsches Kommando
    echo "Syntax: vnc start|stop"
    exit 1
    ;;
esac

# Ende
```

15.3.5 dialog

Um auch auf der Textkonsole ein Gefühl von graphischer Oberfläche zu schaffen, wurde der Befehl `dialog` geschaffen. Er erzeugt verschiedene Arten von Fenstern für Ein- und Ausgabe.

```
dialog --clear
dialog --create-rc DATEI
dialog [OTPIONEN] BOX
```

Boxtypen

Es stehen die folgenden Boxvariationen zur Auswahl:

`--msgbox TEXT HÖHE BREITE`
Eine Box in der Größe HÖHE x Breite und einem TEXT wird dargestellt. Durch Betätigen von `<RETURN>` wird die Box geschlossen.

`--yesno TEXT HÖHE BREITE`
Eine Box in der Größe HÖHE x Breite und einem TEXT. Es besteht die Möglichkeit mit Ja und Nein zu antworten. Bei Ja wird ein Fehlercode von 0 und bei Nein ein Fehlercode von 1 zurückgegeben.

`--infobox TEXT HÖHE BREITE`
Die Infobox ist im Prinzip eine Messagebox. Nur muß die Nachricht nicht mit `<RETURN>` bestätigt werden. Sie bleibt so lange stehen, bis ein neuer Befehl kommt.

`--inputbox TEXT HÖHE BREITE [VORGABE]`
Die Inputbox erlaubt die Eingabe einer Zeichenkette. Die eingegebene Zeichenkette wird über die Standardfehlerausgabe zurückgegeben.

```
ole@enterprise:~/test> dialog --inputbox Eingabe 20 70 2> in.tmp
ole@enterprise:~/test> echo $(cat in.tmp)
Hallo
```

`--textbox DATEI HÖHE BREITE`
Die Textbox stellt den Inhalt einer Datei in einem Nachrichtenfenster dar. Im Gegensatz zum normalen Nachrichtenfenster kann hier durch den Text gescrollt werden. Dieses Fenster wird für größere Texte verwendet. Durch Betätigen von `<RETURN>` wird die Box geschlossen.

`--menu TEXT HÖHE BREITE MENÜPUNKTE [PUNKT BEZEICHNUNG] ...`
Eine Menübox zeigt eine Auswahl von Punkten an, aus denen der Benutzer einen Auswählen kann. Die Bezeichnung des ausgewählten Punkts wird über die Standardfehlerausgabe zurückgegeben. Ein Beispiel für die Menübox finden Sie in Listing 15.9.

Eine Auswahliste für Systembefehle

Die Idee hinter diesem Projekt ist einfach zu beschreiben. Dem Benutzer sollen gewisse Befehle erlaubt werden, ohne daß er Rechte auf andere Befehle bekommt. Hier hilft der Befehl `sudo` (9.3.1) normalerweise weiter. Um das ganze komfortabler zu gestalten, werden alle Befehle, die jemand können muß, in ein Skript geschrieben und können durch ein Auswahldialog ausgewählt werden. Damit das ganze dann auch noch mit *root*-Rechten läuft, wird es erlaubt das Skript mit `sudo` unter *root* zu starten.

Im ersten Schritt wird das Skript von `root` geschrieben und im Verzeichnis `/usr/bin` gespeichert. Die Rechte sind auf 700 gesetzt.

Listing 15.9 *manager*

```
#!/bin/bash

# Führt vordefinierte Befehle aus.
# Die Ausführung dieses Skripts unter sudo als root erlauben

# Temporäre Datei für die Ergebnisse
tempdatei=/tmp/gwinternet.tmp.$$

# Dialogfenster
dialog --menu "Wählen Sie den Dienst" 20 50 10 \
        0       "Swap aktivieren" \
        1       "Swap deaktiviren" \
        2       "Maillogdatei anzeigen" \
        3       "Rechner herunterfahren" \
        4       "Rechner neu starten" \
        2> $tempdatei

eingabe=$(cat $tempdatei)

#Auswerten
```

```
echo "Sie haben ausgewählt: $eingabe"

case $eingabe in
  0) # Swap aktivieren
     /sbin/swapon /swap
     if [ $? -gt 0 ]
     then
       echo "Fehler bei Ausführung" > $tempdatei
     else
       /sbin/swapon -s > $tempdatei
     fi
     dialog --textbox $tempdatei 20 70
     ;;

  1) # Swap deaktivieren
     /sbin/swapoff /swap
     if [ $? -gt 0 ]
     then
       echo "Fehler bei Ausführung" > $tempdatei
     else
       /sbin/swapon -s > $tempdatei
     fi
     dialog --textbox $tempdatei 20 70
     ;;

  2) # /var/log/mail
     dialog --textbox /var/log/mail 20 70
     ;;

  3) # Rechner runterfahren
     # Aufräumen
     rm -f $tempdatei
     echo "Der Rechner wird heruntergefahren"
     /sbin/shutdown -h now
     ;;

  4) # Rechner neu starten
     # Aufräumen
     rm -f $tempdatei
     echo "Der Rechner wird neu gestartet"
     /sbin/shutdown -r now
     ;;

  *) # Falscher Wert
     echo "Falsche Eingabe!"
     ;;
esac


# Aufräumen
rm -f $tempdatei
```

Im zweiten Schritt trägt *root* für den Benutzer mit `visudo` (9.3.3) in die `/etc/sudoers` das Recht ein, daß Skript ausführen zu dürfen.

```
ole             ALL= NOPASSWD: /usr/bin/manager
```

Um das Ganze komfortabler zu machen, wird noch ein Alias angelegt.

```
alias manager="sudo /usr/bin/manager"
```

Jetzt kann der Benutzer *ole* das Skript mit Eingabe von `manager` starten und die angebotenen Befehle ausführen.

15.3.6 while ... do ... done

Oft müssen Aktionen wiederholt werden. Dazu dienen Schleifen. Eine einfache Schleife ist `while`. Solange der getestete Ausdruck wahr ist, solange wird ein Anweisungsblock ausgeführt.

```
while AUSDRUCK
do
  ....
done
```

Ein Beispiel ist das folgende Skript, daß die Adressen eines C-Netzes nach angeschlossenen durchsucht. Dabei wird die Host-Adresse so lange hochgezählt, bis der Endwert erreicht ist.

Listing 15.10 *pingall*

```
#!/bin/bash

# Pingt die Rechner eines C-Klasse-Netzes an

# Werte festlegen
netz='217.89.70.'  # Fester Teil der IP-Nummer
start=1            # Startwert
ende=254           # Endwert

# Schleife initialisieren
i=$start
# Testen, ob der Anweisungsblock noch einmal ausgeführt werden soll
while [ $i -le $ende ]
do
  adresse=$netz$i                        # Adresse zusammensetzen
  echo -ne "\n$adresse"                  # Adresse ausgeben

  # Pingen und auf Erfolg testen
  ping -w 2 $adresse | grep ' 0% loss' &> /dev/null
  if [ $? -eq 0 ]
  then
    echo -n "  --- Rechner vorhanden ---"
  fi

  # Host hochzählen
  i=$(($i+1))
done

echo -e "\n\nFertig"
```

15.3.7 until ... do ... done

Die Until-Schleife ist das Gegenteil von `while`. Während bei `while` die Schleife läuft, wenn der Ausdruck wahr ist, läuft `until` wenn der Ausdruck falsch ist.

```
until AUSDRUCK
do
  ....
done
```

15.3.8 for ... in ... do ... done

Diese Schleife arbeitet alle Elemente einer Liste ab. Jedes Element wird in die angegebene Variable gesteckt und dann mit diesem Wert der Anweisungsblock ausgeführt.

```
for VARIABLE in LISTE
do
  ...
done
```

Listing 15.11 *tolowercase*

```
#!/bin/bash

# Benennt alle angegebenen Dateien im aktuellen Verzeich so um,
# daß sie klein geschrieben sind

for datei in *
do
  mv $datei $(echo $datei | tr A-Z a-z) 2> /dev/null
done
```

15.3.9 read

Der Befehl `read` liest eine Zeile von der Standardeingabe und packt jedes Wort einzeln in eine Variable. Sind mehr Worte als Variablen vorhanden, dann landen die restlichen Worte in der letzten Variablen.

```
read VARIABLENLISTE
```

Listing 15.12 *saghallo*

```
#!/bin/bash

# Test für die Eingabe
#   saghallo

# Ausgabe
echo -n "Gegen Sie Vorname und Nachname ein > "

# Einlesen
read vorname nachname

# Ausgabe
echo "Guten Tag, $vorname $nachname."
echo "Oder darf ich $vorname sagen?"
```

15.4 Weitere Builtin-Befehle der Bash

15.4.1 shopt

Das Kommando `shopt` zeigt und ändert den Status der Variablen zur Steuerung der optionalen Funktionen der Bash.

`shopt [OPTIONEN] [OPTIONSVARIABLENLISTE]`

Optionen	
-p	Zeigt den Status der Variablen an (*print*)
-s	Aktiviert die Option(en) (*set*)
-u	Deaktiviert die Option(en) (*unset*)
-q	Gibt den Status einer Optione durch den Fehlercode zurück (*quiet*)

Die Liste der Variablen und ihre Bedeutung können Sie der Manual-Page zur Bash (`man 1 bash`) entnehmen.

Beispiel

Der Befehl ohne Parameter zeigt den Status der Variablen an.

```
ole@enterprise:~> shopt
cdable_vars     off
cdspell         off
checkhash       off
checkwinsize    on
cmdhist         on
dotglob         off
execfail        off
expand_aliases  on
extglob         on
histreedit      off
histappend      off
histverify      off
hostcomplete    on
huponexit       off
interactive_comments    on
lithist         off
mailwarn        off
no_empty_cmd_completion off
nocaseglob      off
nullglob        off
progcomp        on
promptvars      on
restricted_shell        off
shift_verbose   off
sourcepath      on
xpg_echo        off
ole@enterprise:~> shopt sourcepath
sourcepath      on
```

Mit `-s` und `-u` werden die Variablen gesetzt und wieder abgeschaltet.

```
ole@enterprise:~> shopt -u sourcepath
ole@enterprise:~> shopt sourcepath
sourcepath      off
ole@enterprise:~> shopt -s sourcepath
```

```
ole@enterprise:~> shopt sourcepath
sourcepath      on
```

15.4.2 source

Der Befehl `source` dient zur Ausführung eines Scripts. Er liest die Kommandos aus einer Datei aus und führt dieser in der aktuellen Shell-Umgebung aus. Es wird als Fehler der Fehlercode des letzten ausgeführten Befehls genommen.

```
source DATEINAME [PARAMETER]
```

Enthält der angegebene Dateiname keinen Schrägstrich, so werden die Verzeichnisse aus der Variablen `PATH` nach der Datei durchsucht. Die Dateien müssen nicht ausführbar sein.

Sollte sich die Bash nicht im Posix-Modus befinden, so wird das aktuelle Verzeichnis durchsucht, wenn die Suche in den Pfad-Verzeichnissen zu keinem Erfolg geführt hat. Das Durchsuchen der Pfad-Variablen hängt ab von der Option "`sourcepath`" des `shopt` Befehls (15.4.1). Bei gesetzter Option (`shopt -s sourcepath`) wird der Pfad durchsucht.

Kapitel 16

Allerlei Wissenswertes

Dieses Kapitel ist ein Sammelsurium von Wissen zu Linux, die ich für mich erst einmal notiert habe und noch keinem Themengebiet zugewiesen habe. Trotzdem finde ich diese Informationen "wissenswert" und möchte Sie Ihnen deshalb nicht vorenthalten.

16.1 Dateien und Verzeichnisse

16.1.1 Login Begrüßungstext: `/etc/issue`

Die Datei `/etc/issue` enthält den Begrüßungstext, der vor dem Loginprompt der Konsole steht. Neben reinem Text können aber auch variable Werte durch maskierte Zeichen gesetzt werden.

`\l`	Names des aktuellen Terminals
`\r`	Versionsnummer des Kernels

16.1.2 Nachricht des Tages: `/etc/motd`

Der Inhalt der Datei `/etc/motd` wird nach dem Login durch das Programm `login` dargestellt, bevor die Shell ausgeführt wird. Der Name `motd` steht für "*Message Of The Day*" also "Nachricht des Tages" und wird auch genau dafür verwendet. Dieses Verfahren benötigt wesentlich weniger Ressourcen als eine eMail und informiert doch jeden Benutzer beim Einloggen.

16.1.3 Login-Terminals für *root*: `/etc/securetty`

Die Datei `/etc/securetty` enthält die Terminalbezeichnung für alle Terminals, auf denen sich *root* ins System einloggen darf.

16.1.4 Prozesse sind auch nur Dateien: `/proc`

Schon seit einigen Jahren gibt es ein Pseudo-Dateisystem unter dem Verzeichnis `/proc`. Es wird als Pseudo-Dateisystem bezeichnet, da es sich um ein virtuelles Dateisystem handelt, daß nur im Speicher existiert und kein Äquivalent auf einem Datenträger besitzt. Es enthält Informationen über die im Moment laufender Prozesse und andere Systeminformationen.

Für jeden laufenden Prozess wird ein Verzeichnis mit der PID als Namen angelegt, wie Sie im folgenden Ausschnitt der Dateiliste sehen können.

```
root@defiant:/proc # ls -l
insgesamt 2
dr-xr-xr-x    3 root     root            0 Jan 12 17:52 1
dr-xr-xr-x    3 tapico   users           0 Jan 12 18:00 1001
```

```
dr-xr-xr-x    3 tapico   users           0 Jan 12 18:00 1002
dr-xr-xr-x    3 root     root            0 Jan 12 18:00 11
dr-xr-xr-x    3 root     root            0 Jan 12 18:00 1226
dr-xr-xr-x    3 root     root            0 Jan 12 18:00 13
dr-xr-xr-x    3 root     root            0 Jan 12 18:00 2
dr-xr-xr-x    3 root     root            0 Jan 12 18:00 3
dr-xr-xr-x    3 root     root            0 Jan 12 18:00 324
dr-xr-xr-x    3 root     root            0 Jan 12 18:00 328
...
```

Im Verzeichnis selber befinden sich wiederum Dateien, die nun Informationen zu den Prozeß enthalten. Diese Dateien sind eigentlich nur Verknüpfungen zu Programmen, die diese Informationen zur Verfügung stellen. Schauen wir uns doch mal den `init`-Prozeß mit der PID 1 an.

```
root@defiant:/proc # cd 1
root@defiant:/proc/1 # ls -l
insgesamt 0
-r--r--r--    1 root     root            0 Jan 12 18:05 cmdline
lrwxrwxrwx    1 root     root            0 Jan 12 18:05 cwd -> /
-r--------    1 root     root            0 Jan 12 18:05 environ
lrwxrwxrwx    1 root     root            0 Jan 12 18:05 exe -> /sbin/init
dr-x------    2 root     root            0 Jan 12 18:05 fd
-r--r--r--    1 root     root            0 Jan 12 18:05 maps
-rw-------    1 root     root            0 Jan 12 18:05 mem
lrwxrwxrwx    1 root     root            0 Jan 12 18:05 root -> /
-r--r--r--    1 root     root            0 Jan 12 18:05 stat
-r--r--r--    1 root     root            0 Jan 12 18:05 statm
-r--r--r--    1 root     root            0 Jan 12 18:05 status
root@defiant:/proc/1 # head cmdline environ maps stat statm status
==> cmdline <==
init [5]
==> environ <==
HOME=/ TERM=linux BOOT_IMAGE=linux BOOT_FILE=/boot/vmlinuz
==> maps <==
08048000-080ad000 r-xp 00000000 03:04 9011       /sbin/init
080ad000-080b1000 rw-p 00064000 03:04 9011       /sbin/init
080b1000-080b7000 rwxp 00000000 00:00 0
bffff000-c0000000 rwxp 00000000 00:00 0

==> stat <==
1 (init) S 0 0 0 0 -1 4 59 42642 50 71867 0 468 1279 623 9 0 0 0 28 458752 39
4294967295 134512640 134924332 3221225216 3221223412 134581070 0 0 1467013372
680207875 3222540817 0 0 0 0

==> statm <==
42 39 34 35 0 4 5

==> status <==
Name:   init
State:  S (sleeping)
Pid:    1
PPid:   0
TracerPid:      0
Uid:    0       0       0       0
Gid:    0       0       0       0
FDSize: 32
Groups:
VmSize:      448 kB
```

cmdline Diese Datei enthält die komplette Kommandozeile für den Prozeß. Es sei denn er wurde ausgelagert oder er ist ein Zombie. In diesem Fall enthält die Datei nur das Null-Zeichen.

environ Diese Datei enthält die Umgebungsvariablen für diesen Prozeß. Die Einträge werden durch das Null-Zeichen getrennt.

exe Diese Link ist ein Wegweiser zu der für diesen Befehl ausgeführten Binärdatei.

fd Dieses Verzeichnis enthält Links zu den Datei, die durch diesen Prozeß geöffnet worden sind.

map Diese Datei enthält die für diesen Prozeß genutzten Speicherbereiche und die dafür gültigen Zugriffsrechte. Die möglichen Zugriffsrechte sind
r = lesen
w = schreiben
x = ausführen
s = gemeinsam genutzt (*shared*)
p = privat

stat Diese Datei enthält die Statusinformationen des Prozesses. Der Befehl `ps` (12.6.1) benutzt diese Informationen. Die Informationen sind die PID, Dateiname der Programmdatei in Klammern, Status (z. B. R für Running), PPID, Prozeßgruppen-ID, Session ID, Terminal, Prozeßgruppen-ID des Prozesses, der gerade im Terminal läuft, Flags, 9 Zähler für Speicheroperationen, `nice`-Wert plus 15, Zeit zum nächsten Timeout, Zeit zum nächsten SIGALRM, Startzeit des Prozesses, Größe des virtuellen Speichers, Anzahl der Speicherseiten im Arbeitsspeicher u.s.w.

Weitere Informationen zu den Prozeßinformationen erhalten Sie über die Manualpages (`man 5 proc`).

Neben den Prozesse enthält das Verzeichnis `/proc` auch Dateien mit allgemeineren Informationen über das System. Die folgende gekürzte Liste zeigt eine Auswahl dieser Dateien und Verzeichnisse.

```
root@defiant:/proc # ls -ld [a-z]*
-r--r--r--    1 root     root            0 Jan 12 18:46 cmdline
-r--r--r--    1 root     root            0 Jan 12 18:46 cpuinfo
-r--r--r--    1 root     root            0 Jan 12 18:46 devices
-r--r--r--    1 root     root            0 Jan 12 18:46 dma
-r--r--r--    1 root     root            0 Jan 12 18:46 filesystems
-r--r--r--    1 root     root            0 Jan 12 16:49 interrupts
-r--r--r--    1 root     root            0 Jan 12 18:46 ioports
-r--------    1 root     root     67047424 Jan 12 18:46 kcore
-r--------    1 root     root            0 Jan 12 16:47 kmsg
-r--r--r--    1 root     root            0 Jan 12 18:46 ksyms
-r--r--r--    1 root     root            0 Jan 12 18:46 loadavg
-r--r--r--    1 root     root            0 Jan 12 18:46 meminfo
-r--r--r--    1 root     root            0 Jan 12 18:46 modules
-r--r--r--    1 root     root            0 Jan 12 18:46 mounts
dr-xr-xr-x    4 root     root            0 Jan 12 18:46 net
-r--r--r--    1 root     root            0 Jan 12 18:46 partitions
-r--r--r--    1 root     root            0 Jan 12 18:46 pci
dr-xr-xr-x    2 root     root            0 Jan 12 18:46 scsi
-r--r--r--    1 root     root            0 Jan 12 18:46 stat
-r--r--r--    1 root     root            0 Jan 12 18:46 swaps
-r--r--r--    1 root     root            0 Jan 12 18:46 uptime
-r--r--r--    1 root     root            0 Jan 12 18:46 version
```

cpuinfo Informationen über die CPU und ihre Leistung

devices Liste der wichtigsten Gerätetreiber

dma Liste der verwendeten DMA-Kanäle

filesystems Liste der bekannten Dateisysteme

interrupts Liste der Interrupts und wie häufig sie aufgerufen worden sind.

ioports Lister der bekannten und definierten IO-Schnittstellen

kcore Gerädtedatei für den Zugang zum physikalischen Hauptspeicher

kmsg Hier können Systemfehlermeldungen ausgelesen werden

ksyms Liste der Kernelsymbole und ihrer Speicheradressen

loadavg Durchschnittliche Belastung des Systems in den letzten Minuten

meminfo Enthält Informationen über die aktuelle Speicherauslastung

modules Liste der geladenen und aktuelle genutzten Module

mounts Liste der eingehängten Dateisysteme und deren Mount-Points

net Verzeichnis mit Informationen zur Netzwerkauslastung

partitions Liste der vorhandenen Partitionen

pci Liste der gefundenen PCI-Geräte

scsi Verzeichnis mit Informationen über SCSI-Geräte

stat Allgemeine Informationen über das System

swaps Liste der Swap-Partitionen

uptime Zeit in Sekunden seit Systemstart und die Leerlaufzeit seitdem

version Aktuelle Version des Linux-Kernels

16.2 YaST

Das "Yet another Setup Tool" gibt es in zwei Versionen. Die Urversion ist rein textbasiert. Sie ist einfach zu bedienen und vor allem schnell. Yast 2 hingegen ist graphikorientiert (kommt daher den Mausschubsern sehr entgegen) und bietet auch einige verbesserte und zusätzliche Funktionen an. Yast 2 kann auch in einem Textmodus betrieben werden, allerdings ist die Handhabung dort nicht sehr effektiv. Seit SuSE 8.0 gibt es kein YaST mehr sondern nur noch Yast 2.

16.2.1 Benutzerverwaltung unter YaST

Starten Sie zunächst eine Konsole im X-Window-System oder wechseln Sie zu einer Textkonsole (STRG + ALT + F1). Starten Sie dort YaST indem Sie `yast` eintippen und mit der Return-Taste bestätigen. In dem daraufhin erscheinenden Programm wählen Sie durch die Cursor-Tasten den Menüpunkt: *Administration des Systems*. Dann wählen Sie den Menüpunkt *Benutzerverwaltung*.

Geben Sie im ersten Feld den Namen *walter* ein. Die folgenden Felder werden von YAST automatisch ausgefüllt. Bestätigen Sie einfach die Angaben mit der Return-Taste. Als Passwort geben Sie `hallo` ein und wiederholen das Passwort im Wiederholungsfeld. Mit der Taste `F4` legen Sie dann den Benutzer an. Mit der Taste `F10` können Sie die Maske wieder verlassen.

16.2.2 Installation von weiteren Paketen mit YaST

Da nicht alle Pakete bei einer Standardinstallation installiert werden, ist es oft notwendig weitere Pakete von den SuSE-CDs zu installieren. Auch diese Aufgabe übernimmt `yast`. Starten Sie wieder als *root* `yast`, wie Sie es auch bei der Benutzereinrichtung gemacht haben.

Auswahl der Installationsquelle

Unter anderem zeigt das Menüfenster das aktuelle Installationsmedium an (Quellmedium). Steht dort nichts oder der Hinweis `[NOT OK]`, dann müssen Sie das Installationsmedium noch einmal angeben. Wählen Sie dazu den Menüpunkt *Einstellungen zur Installation* und dort den Punkt *Installationsquelle auswählen*. Wenn Sie von CD oder DVD installieren wollen, wählen Sie die passende Rubrik. Dann müssen Sie noch den passenden Gerätenamen angeben. Drücken Sie einmal die Taste `<ESC>` und schon sind Sie wieder im Startmenü.

Installation

Wählen Sie diesmal den Menüpunkt *Paketverwaltung*. Über den Menüpunkt *Konfiguration ändern/erstellen* gelagen Sie nun zur Paketauswahl. Das kann etwas länger dauern, da alle Paketbeschreibungen eingelesen werden müssen (Über 2000 Stück bei SuSE 7.3 Professional). Leider bietet `yast` in diesem Menü keine Suche in den Paketen an. Die Pakete sind in Serien thematisch sortiert. Um jetzt z. B. die Korn-Shell nachzuinstallieren, schauen Sie ins Paket *ap*. Da es sich hier um die *Public Domain Korn Shell* handelt, müssen Sie das Paket `pdksh` markieren. Dies erfolgt mit der Leertaste. Die Buchstaben in den eckigen Klammern vorm Paketnamen bedeuten:

X	Zur Installation vorgesehen
i	Ist bereits installiert
D	Zur Deinstallation vorgesehen
R	Zum Updaten vorgesehen

Mit `<F10>` verlassen Sie das jeweilige Menü wieder. Mit der Taste `<F4>` können Sie die Sortierung von den Serien zu einer alpabetischen Aufzählung aller Pakete ändern. Dies ist insofern nützlich, da Sie nicht immer wissen, in welcher Serie sich ein Paket befindet.

Nachdem Sie nun die zu installierenden Pakete ausgewählt haben und mit `<F10>` die Auswahl verlassen haben, können Sie die Installation nun mit dem Menüpunkt *Installation starten* beginnen. Nach der Installation der Pakete startet nun *SuSEConfig* um die Eintragungen in die Konfigurationsdateien für die installierten Programme zu erledigen. Dies kann etwas länger dauern.

16.3 Grafik und Bilder

16.3.1 convert

Das Kommandozeilentool `convert` konvertiert Bilddateien von einem Bildformat zu einem anderen. Dabei können Filter auf die Bilder angewendet werden.

```
convert [OPTIONEN] QUELLBILDDATEI ZIELBILDDATEI
```

Das Format für die Zieldatei ergibt sich aus der Endung. Neben den gängigen Formaten wie JPEG, PNG, BMP und TIFF werden eine Vielzahl weiterer Formate unterstützt. Eine genaue Liste können Sie der Manpage entnehmen.

Aus der Vielzahl von Optionen möchte ich Ihnen nur ein paar vorstellen.

Optionen

`-colorspace RAUM`	Erzeugt einen anderen Farbraum für das neue Bild. Mögliche Werte sind GRAY, OHTA, RGB, Transparent, XYZ, YCbCr, YIQ, YPbPr, YUV, oder CMYK.
`-monochrome`	Wandelt das Bild in ein 1-Bit S/W-Bild um.
`-verbose`	Liefert Informationen über das Bild.

Beispiele

Um ein PNG-Bild in ein Postscript-Bild zu konvertieren, wenden Sie folgenden Befehl an:
`convert screenshot.png screenshot.ps`

Um das Bild dann noch in ein Schwarz-Weiß-Bild umzuwandeln verwenden Sie
`convert -monochrome screenshot.png screenshot.ps`

Um das Bild dann noch in ein Schwarz-Weiß-Bild umzuwandeln verwenden Sie
`convert -colorspace GRAY screenshot.png screenshot.ps`

16.4 Internet

16.4.1 wget

Jeder kennt sicherlich das Problem eine große Datei herunterzuladen. Gerade wenn man bei 99,99% angekommen ist, bricht der Download ab und man muß wieder von vorne beginnen. Das Kommandozeilentool `wget` ermöglicht einen abgebrochenen Download wieder aufzunehmen. Und nicht nur das. Auch komplette Seiten mit Bildern und sogar ganze Homepages können zum Offline-Lesen heruntergeladen werden. Da es über die Shell gestartet wird, steht einer zeitgesteuerten Nutzung mit den Dämonen `atd` (13.1.1) und `crond` (13.1.7) auch nichts im Wege. So kann `wget` als Download-Manager wie auch als Offline-Reader eingesetzt werden.

`wget [OPTIONEN] [URL]`

GNU Wget unterstützt die Protokolle HTTP, HTTPS, und FTP sowie die Arbeit mit HTTP Proxies.

Optionen	
`-V`	Version (`--version`)
`-b`	Arbeitet im Hintergrund (`--background`)
`-c`	Abgebrochenen Download wieder aufnehmen (`--continue`)
`-r`	Folgt den Links (`--recursive`)
`-k`	Konvertiert Links um fürs Offline-Lesen (`--convert-links`)
`-p`	Alle Elemente (z. B. Bilder) einer Webseite werden ebenfalls heruntergeladen (`--page-requisites`)
`-l TIEFE`	Gibt die Tiefe der Verfolgung an (`--level=TIEFE`)
`-m`	Erstellt exakte Kopie der Seite (`--mirror`)
`-H`	Folgt auch Links zu anderen Hosts (`--span-hosts`)
`-np`	Folgt nicht den Links ins Elternverzeichnis (`--no-parent`)
`-t VERSUCHE`	Anzahl der Downloadversuche (`--tries=Versuche`)
`-A MUSTER`	Lädt nur Dateien, die das MUSTER enthalten (`--accept MUSTER`)

Beispiele

Download abgebrochen Sie wollen die große Datei `http://www.fibel.org/download/lfo-0.4.pdf` (1,5 MB) herunterladen.
`wget http://www.fibel.org/download/lfo-0.4.pdf`
Bricht der Download ab, können Sie durch Eingabe des gleichen Befehle den Download fortsetzen.
`wget http://www.fibel.org/download/lfo-0.4.pdf`
Explizit können Sie `wget` mit der Option `-c` dazu auffordern, den Rest einer Datei zu laden.
`wget -c http://www.fibel.org/download/lfo-0.4.pdf`
Dabei schaut `wget` nach, ob eine Datei `lfo-0.4.pdf` existiert und beginnt dann den restlichen Teil vom Server zu laden. Dies funktioniert natürlich nur bei Servern, die dies auch unterstützen.

Download über schlechte Leitung Bei einer schlechten Leitung erfordert es oft mehrere Versuche um eine Datei komplett herunterzuladen. Daher kann `wget` angewiesen werden, den Download öfter zu versuchen.
`wget -t 45 http://www.fibel.org/download/lfo-0.4.pfd.org`

Download einer Seite inklusive Bilder Wollen Sie nicht nur die reine HTML-Seite herunterladen, sondern auch alle anderen Elemente, die zur Darstellung dazugehören, dann kommt der Schalter `-p` ins

Spiel.
`wget -p http://www.fibel.org/index.html`

Offline-Version einer Homepage erstellen Sie haben im Netz eine interessante Webpräsenz entdeckt und wollen diese mit ihrer gesamten Struktur auf Ihrer Platte speichern. Das erreichen Sie durch folgenden Befehl:
`wget -rkpl 5 http://www.oleswelt.de/rezepte/`
Das `-r` sorgt dafür, daß auch die verlinkten Seiten heruntergeladen werden. Über wie viele Webseiten den Links gefolgt werden soll, kann mit dem Schalter `-l` und der Angabe der Tiefe geregelt werden. Der Schalter `-k` sorgt dafür, daß absolute Links in relative Links umgewandelt werden. Wurde die Seite nicht mit heruntergeladen, so wird ein absoluter Link mit mit Protokoll und Domain-Name (z. B. aus `apfelmus.html` wird `http://www.oleswelt.de/rezepte/apfelmus.html`) erzeugt.

Spiegelung einer Homepage erstellen Eine Spiegelung (Mirror) ist eine exakte Kopie einer Homepage um sie zusätzlich auf einem oder mehreren Servern anzubieten. Dies erfolgt in der Regel um die Last bei stark frequentierten Webpräsenzen zu verteilen.
`wget -m 5 http://www.oleswelt.de/rezepte/`
Entspricht den Schaltern `-r -N -l inf -nr`.

Nur bestimmte Dateien herunterladen Um z. B. nur die JPEG-Bilder aus einem FTP-Server-Verzeichnis herunterzuladen, kommt der Schalter `-A` ins Spiel.
`wget -rl 1 -np -A .jpg,.jpeg ftp://ftp.heidibilder.de/pic/`

16.5 Fernbedienung

16.5.1 X-Window-Programme vom Server

Eine SSH-Verbindung erlaubt Ihnen bekanntlich das Arbeiten auf einem anderen entfernten Rechner über Ihren Rechner. Das gilt aber nur für Kommandozeilenprogrammen. Eine der Stärken des X-Window-Systems ist es, daß auch X-Window-Programme auf dem Server laufen, aber die Darstellung (die Fenster) auf dem Client erfolgt.

Bezeichnen wir den Rechner an dem wir sitzen als **Client** (`defiant`) und den Rechner, der die Rechnerpower zur Verfügung stellen soll, als **Server** (enterprise).

Als erstes erlauben wir, daß der Server Daten an das X-Window-System liefern darf. Dies erfolgt mittels des Befehls `xhost`.

```
barclay@defiant:~> xhost enterprise
enterprise being added to access control list
```

Anstelle des Namens kann natürlich auch die IP-Adresse verwendet werden.

Danach öffnen wir eine SSH-Verbindung vom Client zum Server.

```
barclay@defiant:~> ssh enterprise
barclay@enterprise's password:
Last login: Fri May 24 23:08:18 2002 from defiant
Have a lot of fun...
barclay@enterprise:~>
```

Nun müssen wir noch dem Server mitteilen an welche Adresse die X-Window-Daten für diese Sitzung geliefert werden sollen. Dies erfolgt über die Umgebungsvariable `DISPLAY`.

```
ole@enterprise:~> echo $DISPLAY
```

```
10:0.0
ole@enterprise:~> DISPLAY=defiant:0.0; export DISPLAY
ole@enterprise:~> nedit
```

Und schon öffnet sich das Fenster des Editors NEdit auf dem Bildschirm des Clients. Ein Blick in die Dateiliste über den Menüpunkt *File/Open* zeigt, daß wir auf dem Server arbeiten.

Übrigens muß auf dem Serverrechner das X-Window-System nicht laufen, damit dies funktioniert.

16.5.2 xhost

Das Tool `xhost` legt fest, welcher Rechner auf den X-Server zugreifen darf.

```
xhost [[+|-][NAMENSLISTE]]
```

Optionen

`+NAME`	Fügt den angegebenen Namen zur Liste der erlaubten Namen hinzu. Das Plus-Zeichen ist optional.
`-NAME`	Entfernt den angegebenen Namen aus der Liste der erlaubten Namen.
`+`	Schaltet die Liste aus; jeder hat Zugriff
`-`	Schaltet die Liste ein

Weitere Informationen entnehmen Sie bitten den Manual-Pages.

Beispiele So wird ein Name zur Liste hinzugefügt.

```
barclay@defiant:~> xhost +enterprise
enterprise being added to access control list
```

Der Befehl ohne Parameter gibt eine Zustandsmeldung aus.

```
barclay@defiant:~> xhost
access control enabled, only authorized clients can connect
INET:enterprise.fibel.org
```

Teil II

Anhang

Anhang A

Listen

A.1 Die Shell-Befehle

A		
`alias`	Legt eine Kurzbezeichnung für einen Befehl an	5.2.8
`apropos`	Durchsucht die Whatis-Datenbank nach einem Stichwort	6.3.5
`at`	Startet einen Job zu einem bestimmten Zeitpunkt	13.1.1
`atq`	Zeigt die aktuellen Jobs an	13.1.2
`atrm`	Löscht wartende `at`-Jobs	13.1.3
B		
`batch`	Legt einen Hintergrundprozeß mit sehr geringer Prioriät an	13.1.5
`bg`	Startet einen im Hintergrund gestoppten Job	12.7.6
C		
`cal`	Zeigt einen Kalender an	4.6.2
`cat`	Zeigt den Inhalt (mehrerer) Dateien an	4.5.2
`cd`	Wechselt das Arbeitsverzeichnis	4.3.2
`chgrp`	Ändert die Gruppe einer Datei	9.2.2
`chmod`	Ändert die Rechte für eine Datei	9.2.3
`chown`	Ändert Besitzer und Gruppe einer Datei	9.2.1
`chpasswd`	Ändert die Paßwörter mehrerer Benutzer anhand einer Liste	8.2.5
`chsh`	Ändert die Login-Shell eines Benutzers	4.1.1
`cksum`	Erzeugt eine CRC-Checksumme für eine Datei	7.4.3
`clear`	Löscht den Bildschirm	4.6.1
`comm`	Vergleicht zwei sortierte Dateien miteinander	7.5.2
`compress`	Komprimiert einzelne Dateien; Endung `.Z`	13.5.3
`configure`	Bei vielen Quellpaketen mitgeliefertes Skript zur Vorbereitung der Installation	14.1.3
`cp`	Legt Kopien von Dateien an	4.5.3
`cpio`	Kopiert Dateien in und aus einem Archiv	13.4.3
`crontab`	Konfigurationsprogramm für `crond`	13.1.6
`cut`	Gibt Spalten aus einer Ausgabe oder Datei aus	7.6.1
D		
`date`	Zeigt die aktuelle Uhrzeit/Datum an und setzt sie auch	4.6.3
`dd`	Kopiert den Inhalt kompletter Geräte in eine Datei und umgekehrt	4.5.4
`depmod`	Erstellt die Datei `modules.dep` mit den Modulabhängigkeiten	14.5.8
`df`	Zeigt die Nutzung der Datenträger an	10.6.2
`diff`	Vergleicht zwei Textdateien und gibt die Unterschiede in einem für `patch` lesbaren Format aus	7.7.7
`dir`	Zeigt den Inhalt eines Verzeichnis an	4.3.4

L		
`last`	Zeigt an, wer sich zuletzt eingeloggt hat	8.7.6
`lastlog`	Zeigt die letzten Einlogzeitpunkte der Benutzer an	8.7.7
`ldconfig`	Aktualisiert die Liste der genutzten Bibliotheken für `ld.so`	14.2.3
`ldd`	Zeigt eine Liste der für ein Programm benötigten Bibliotheken an	14.2.2
`less`	Zeigt Dateien seitenweise an, komfortabler als `more`	4.5.10
`lesskey`	Konfiguriert die Steuerbefehle von `less`	4.5.11
`lilo`	Einrichten des Bootloaders Lilo	12.3.1
`ln`	Legt einen harten oder symbolischen Link an	10.5.3
`locate`	Durchsucht das Dateisystem nach Dateien anhand einer Datenbank	11.3.2
`logname`	Name, unter dem eingeloggt wurde	8.7.3
`logout`	Beendet die aktuelle Sitzung	4.6.7
`logrotate`	Konfiguriert `syslogd`, Verwaltung der alten Logs	13.2.4
`ls`	Zeigt den Inhalt von Verzeichnissen an	4.3.3
`lsmod`	Zeigt alle geladenen Kernelmodule an	14.5.2
M		
`make`	Tool für die Installation von Programmen	14.1.4
`man`	Betrachter für die Manual-Pages	6.1.2
`mkdir`	Legt Verzeichnisse an	4.3.6
`mkfifo`	Erzeugt eine Named Pipe	10.4.1
`mkfs`	Formatiert eine Partition mit einem Dateisystem	10.3.1
`modinfo`	Informationen über eingebundene Module	14.5.5
`modprobe`	Einbinden von Modulen in den Kernel mit erweiterten Funktionen	14.5.6
`more`	Zeigt Dateien seitenweise an	4.5.9
`mount`	Bindet ein Dateisystem in den aktuellen Verzeichnisbaum ein	10.2.1
`mv`	Ändert Pfad und Namen von Dateien (Verschieben, Umbenennen)	4.5.6
N		
`newgrp`	Wechselt die aktuelle Hauptgruppe eines Benutzers	8.4.5
`nice`	Legt eine Prozeßpriorität beim Starten fest	12.7.1
`nl`	Nummeriert die Zeilen vor der Ausgabe durch	7.1.2
O		
`od`	Gibt eine Binärdatei in verschiedenen Notationen aus	7.1.3
P		
`passwd`	Ändert das Paßwort eines Benutzers	8.2.4
`paste`	Fügt Dateien spaltenweise zusammen	7.6.3
`patch`	Wandelt eine Datei mit einer Patchdatei zu einer neuen Version um	7.7.8
`pr`	Formatiert eine Textdatei für die Druckausgabe	7.2.2
`printenv`	Zeigt die Umgebungsvariablen an	5.2.4
`ps`	Zeigt die aktuell laufenden Prozesse an	12.6.1
`pstree`	Zeigt die laufenden Prozesse als Baumstruktur an	12.6.2
`pwconv`	Aktiviert das Shadow-System und konvertiert die Paßwörter	8.5.3
`pwd`	Zeigt den Namen und Pfad des aktuellen Arbeitsverzeichnis an	4.3.1
`pwunconv`	Deaktiviert das Shadow-System und konvertiert die Paßwörter	8.5.4
Q		
`quota`	Zeigt die Quoten (Plattenplatz) für die Benutzer an	11.4.1
`quotacheck`	Aktualisiert die Quoteninformationen	11.4.6
`quotaoff`	Schaltet die Quotenüberwachung aus	11.4.3
`quotaon`	Schaltet die Quotenüberwachung ein	11.4.2
`quotastats`	Kalkuliert den benutzten Platz für jeden Benutzer und für jede Gruppe	11.4.7
R		
`renice`	Verändert die Prozeßpriorität laufender Prozesse	12.7.2

A.2 Verzeichnisse

/	Das oberste Verzeichnis	4.4
/bin	Die wichtigsten Programme	4.4
/boot	Kernel und Boot-Dateien	4.4
/dev	Gerätedateien	4.4
/etc	Konfigurations- und Informationsdateien	4.4
/etc/rc.d	Skripte und rc-Verzeichnisse für den Bootvorgang	4.4
/etc/skel	Musterdateien für den Benutzer	4.4
/etc/X11	Konfigurationsdateien X-Window-System	4.4
/home	Die Heimatverzeichnisse der Benutzer	4.4
/lib	Bibliotheken des C-Compilers (Shared Libraries)	4.4
/lib/modules	Elternverzeichnis für die Modulverzeichnisse	14.5.1
/Lost+found	Verzeichnis für wiederhergestellte Dateien	4.4
/mnt	Hilfsverzeichnis zum Mounten	4.4
/proc	Virtuelles Dateisystem zur Verwaltung der Systemprozesse	4.4
/root	Heimatverzeichnis des Superusers	4.4
/sbin	Die wichtigsten Programme für den Superuser	4.4
/tmp	Verzeichnis für temporäre Dateien	4.4
/usr	Wichtige Programme, die nicht zum Booten nötig sind, sowie Dokumentationen	4.4
/usr/bin	Befehle für alle Benutzer	4.4
/usr/doc/FAQ	Frequently Asked Questions (FAQ)	6.1
/usr/doc/HOWTO	Die HOWTO-Dokumentationen	6.1
/usr/doc/<Programmname>	Dokumentationen für Programme	6.1
/usr/include	Standard C/C++-Header-Dteien	4.4
/usr/info	Dateien für TexInfo (info)	6.1
/usr/lib	Statische Programmbibliotheken verschiedener Programmiersprachen	4.4
/usr/local	Programme, die nicht Bestandteil des Systems sind	4.4
/usr/local/bin	Binärdateien für die Programme, die nicht Bestandteil des Systems sind	4.4
/usr/local/sbin	Lokal installierte Administrator-Tools	4.4
/usr/man	Dateien für Man-Pages	6.1
/usr/sbin	Analog zu /usr/bin für den Superuser	4.4
/usr/share/doc/FAQ	Frequently Asked Questions (FAQ) (SuSE)	6.1
/usr/share/doc/howto	Die HOWTO-Dokumentationen (SuSE)	6.1
/usr/share/doc/packages	Dokumentationen für Programmpakete (SuSE)	6.1
/usr/share/info	Dateien für TexInfo (info) (SuSE)	6.1
/usr/share/man	Dateien für Man-Pages (SuSE)	6.1
/usr/src	Quellcodes der Programme	4.4
/usr/src/linux	Quellcode des Linux-Kernels	14.6.1
/usr/X11R6	Dateien für das X-Window-System	4.4
/var	Informationsdateien für das System	4.4
/var/tmp	Verzeichnis für temporäre Dateien	4.4
/var/spool	Warteschlangen	4.4
/var/spool/lp	Druckaufträge	4.4
/var/spool/mail	Mails	4.4
/var/spool/uucp	Kommunikationsdienst uucp	4.4
/var/spool/cron	Aufträge des Daemons crond	4.4

A.3 Wichtige Dateien

/etc/at.allow	Liste der erlaubten Benutzer für at	13.1.4
/etc/at.deny	Liste der verbotenen Benutzer für at	13.1.4
/etc/conf.modules	Konfigurationsdatei für die Module (siehe auch modules.conf)	14.5.9
/etc/cron.allow	Liste der erlaubten Benutzer für crond	13.1.8
/etc/cron.deny	Liste der verbotenen Benutzer für crond	13.1.8
/etc/bashrc	Konfigurationsdatei für interaktive Shells, die nicht die Login-Shell sind	8.6.2
/etc/DIR_COLORS	Konfigurationsdatei für die Farbausgabe von ls	4.3.3
/etc/fstab	Konfigurationsdatei für das Einbinden von Dateisystemen in den Verzeichnisbaum	10.2.4
/etc/group	Liste der Gruppenkonten	8.4.1
/etc/gshadow	Liste der Gruppenpaßwörter für das Shadow-System	8.5.2
/etc/inittab	Konfigurationsdatei für den Init-Daemon	12.2.1
/etc/inputrc	Konfigurationsdatei für die Eingabezeile	5.1.2
/etc/issue	Enthält die Identifikationsmeldung, die vor dem Login-Prompt erscheint	16.1.1
/etc/issue.net	Enthält die Identifikationsmeldung für Telnet-Sitzungen	
/etc/ld.so.cache	Liste der Bibliotheken für den Runtime Linker ld.so	14.2.1
/etc/ld.so.conf	Konfigurationsdatei für ldconfig	14.2.3
/etc/lilo.conf	Konfigurationsdatei für den Bootmanagerersteller lilo	12.3.2
/etc/login.defs	Konfigurationsdatei für den Login-Vorgang	8.2.6
/etc/logrotate.conf	Konfigurationsdatei für das Programm logrotate	13.2.4
/etc/magic	Signaturen für Dateitypen	4.5.13
/etc/man.config	Konfigurationsdatei für man	6.1.2
/etc/manpath.config	Konfigurationsdatei für man	6.1.2
/etc/modules.conf	Konfigurationsdatei für die Module (siehe auch conf.modules)	14.5.9
/etc/motd	Enthält die Meldung, die nach dem Einloggen erscheint	16.1.2
/etc/mtab	Liste der zur Zeit gemounteten Dateisysteme	10.2.5
/etc/passwd	Liste der Benutzerkonten	8.2.1
/etc/profile	Konfigurationsdatei für die Shell	8.6.1
/etc/securetty	Liste der Terminals für *root*-Anmeldung	16.1.3
/etc/shadow	Liste der Paßwörter für das Shadow-System	8.5.1
/etc/shells	Liste der erlaubten Shells	4.1.1
/etc/shutdown.allow	Liste der Benutzer, die das System herunterfahren dürfen	12.4.1
/etc/syslog.conf	Konfigurationsdatei für den den Daemon syslogd	13.2.2
/usr/share/misc/magic	siehe /etc/magic	4.5.13
/var/lib/locatedb	Datenbank für den Befehl locate	11.3.3
/var/log/lastlog	Speichert den letzten Zeitpunkt des Einloggens des Benutzers	13.2.3
/var/log/messages	Meldungen über das Hochfahren des Rechners	13.2.3
/var/log/wtmp	Speichert Logzeiten und Systemstarts	13.2.3
/var/run/utmp	Speichert die aktuell eingeloggten Benutzer	13.2.3
.bash_history	Speicher für die zuletzt ausgeführten Befehle	5.3
.bash_profile	Individuelle Konfigurationsdatei für interaktive Shells Shell	5.2.6
.bashrc	Individuelle Konfigurationsdatei für interaktive Shells Shell	8.6.2
.dir_colors	Individuelle Konfigurationsdatei für die Farbausgabe von ls	4.3.3
.inputrc	Individuelle Konfigurationsdatei für die Eingabezeile	5.1.2
.profile	Individuelle Konfigurationsdatei für die Shell	8.6.1
Makefile	Installationsanleitung für das Tool make	14.1.4
modules.dep	Liste der Modulabhängigkeiten	14.5.7
quota.group	Konfigurationsdatei für die Gruppenquoten einer Partition	11.4

`quota.h`	Konfigurationsdatei für die Standardeinstellung der Quoten einer Partition	11.4.4
`quota.user`	Konfigurationsdatei für die Userquoten einer Partition	11.4

A.4 Glossar

Header-Dateien Dateien, die die Schnittstellenbeschreibungen von Funktionen (meistens C oder C++) enthält, aber nicht den dazugehörigen Code. Wenn ein Programm gegen externe Bibliotheken gelinkt werden, werden die entsprechenden Header-Dateien unbedingt benötigt.

Patch (engl. *Flicken*) Kleine Software-Pakete, mit denen die Entwickler Fehler in Produkten beseitigen. Da sie nur die Korrektur der fehlerhaften Programmabschnitte erhalten, sind sie viel kleiner. Es werden lediglich einige Daten und Funktionen ausgetauscht.

Anhang B

Der Unterricht

Dieses Skript ist als Begleitmaterial zu meinem Unterricht gedacht. Deshalb möchte ich an dieser Stelle Hinweise und Ratschläge für den Unterricht unterbringen.

B.1 Exportieren des X11-Desktops

Gerade bei der Arbeit mit dem X-Window-System ist es mühsam den Kursteilnehmern die Arbeitsweise der GUI-Programme zu erklären. Besser ist es, wenn alle Teilnehmer den Desktop beobachten können, während der Dozent die Funktionen vormacht. Meistens wird in solchen Fällen ein sogenannter Beamer eingesetzt, der das Monitorbild auf eine Leinwand wirft. Diese Geräte sind natürlich nicht gerade billig und geraten bei ungünstigen Raum- und Lichtverhältnissen schnell an ihre Grenzen.

Unter Windows ermöglicht das Programm Netmeeting nicht nur das telefonieren übers Netz, sondern auch der Desktop kann auf andere Rechner übertragen werden.

Unter Linux bietet sich natürlich der Einsatz eines X11-Servers an. Das Projekt Virtual Network Computing (VNC) der University of Cambridge, des Department of Engineering und AT&T arbeitet an einem virtuelle X11-Server, der auf anderen Rechnern bedient werden kann.

Dieser X-Server und seine Viewer sind für verschiedene Betriebssysteme wie u.a. Linux und Windows erhältlich. Das erlaubt z. B. die Arbeit auf einem Linux-System, während der Benutzer auf einem Windows-Rechner arbeitet.

Das Paket `vnc` ist auf der SuSE 8.0 enthalten. Nach der Installation steht der X-Server (`Xvnc`), das Startprogramm für den Server (`vncserver`), der Client `vncviewer`) und das Passwort-Programm (`vncpasswd`) zur Verfügung.

Die aktuelle Software und weitergehende Informationen erhalten Sie über die Webseite des Projekts `http://www.uk.research.att.com/vnc/` im Internet.

B.1.1 Xvnc

`Xvnc` ist der Unix VNC-Server, der auf einem normalen X-Server basiert. Die Programme können diesen Server genau wie einen normalen X-Server für die Darstellung ihrer Fenster verwenden. Allerdings erscheinen die Fenster auf jedem VNC-Betrachter (`vncviewer`), der mit dem Server verbunden ist. Auch auf dem lokalen Rechner, auf dem Server und Programm laufen, wird ein Betrachter benötigt.

Damit ist `Xvnc` in Wirklichkeit zwei Server in einem. Für Programme ist er ein X-Server und für den VNC-Betrachter ein VNC-Server. Die Display-Bezeichnung beim VNC-Server ist an den normalen X-Server angelehnt. So bezeichnet `defiant:3` das dritte Display auf dem Rechner mit dem Namen *defiant*. Natürlich können auch IP-Nummern wie in `217.89.70.60:13` verwendet werden um ein Display auf einen entfernten Rechner anzusprechen.

Der VNC-Server wird normalerweise nicht direkt über das Programm `Xvnc` gestartet. Das mitgelieferte Perl-Skript `vncserver` vereinfacht den Start des Servers wesentlich.

```
Xvnc [OPTIONEN] [:DISPLAYNUMMER]
```

Die Optionen für den `Xvnc`-Befehl sind teilweise speziell, teilweise sind sie aber mit denen des normalen X-Servers identisch. Hier eine Auswahl der wichtigsten Optionen.

Optionen	
`-h`	Zeigt die Hilfe an
`-name`	Name des Servers bzw. exportierten Desktops, erscheint im Viewer
`-geometry BIPxHIP`	Größe des zu exportierenden Desktops in Pixeln (Standard 1024x768)
`-depth TIEFE`	Farbtiefe der Anzeige in Bit (Standard 8 bit)
`-inetd`	Ermöglicht den Start des Servers über inetd
`-alwaysshared`	Es können mehrere Clients zur gleichen Zeit am gleichen Server arbeiten
`-nevershared`	Es kann nur ein Client zur Zeit am Server arbeiten
`-dontdisconnect`	Bei einer eingehenden "not-shared" Verbindung wird der neue Client zurückgewiesen; ohne diese Option wird der alte Client vom Server getrennt
`-localhost`	Nur Betrachter vom eigenen Rechner dürfen den Server kontaktieren
`-rfbauth DATEI`	Verwendet die DATEI als Passwort-Datei

B.1.2 vncserver

Dieses Perl-Skript erlaubt einen wesentlich einfacheren Start des VNC-Servers. Es benutzt die gleiche Optionen wie Xvnc.

```
vncserver [OPTIONEN] [:DISPLAYNUMMER]
vncserver -kill :DISPLAYNUMMER
```

Die Dateien für VNC werden im Heimatverzeichnis in dem versteckten Verzeichnis `.vnc` abgelegt. Für jeden gestarteten Server wird dort eine Log-Datei und eine PID-Datei abgelegt. Mit Hilfe dieser PID-Datei kann das Skript über die Option `-kill` einen VNC-Server beenden.

Dort befinden sich auch die Passwort-Dateien, denn nur nach Eingabe eines gültigen Passworts kann sich der Betrachter mit dem Server verbinden.

Sehr wichtig ist dort die Datei `xstartup`, denn diese wird nach dem Starten des X-Servers ausgeführt und sorgt für den Start des Windowmanagers und bestimmter Programme.

Listing B.1 *Standard xstartup*

```
#!/bin/sh

xrdb $HOME/.Xresources
xsetroot -solid grey
xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
twm &
```

Im obigen Beispiel wird das X-Window-System mit `xrdb` konfiguriert unter Benutzung der normalen X-Window-Ressourcen-Datei des Benutzers. Der Hintergrund wird mit `xsetroot` auf grau gesetzt. Danach werden ein xterm-Terminal und der Windowmanager `twm` gestartet. Die Variable `$VNCDESKTOP` enthält den Namen des laufenden Servers bzw. Desktops.

B.1.3 vncpasswd

Beim ersten Starten eines VNC-Servers fragt das System nach einem Passwort. Mit diesem Passwort wird sichergestellt, daß nur autorisierte Personen Zugang zum X11-Server erlangen. Das Passwort wird in der Datei `~/.vnc/passwd` gespeichert. Der Befehl `vncpasswd` ändert dieses Passwort bzw. legt weitere Passwortdateien an.

```
vncpasswd [PASSWORTDATEI]
```

Beispiel

Sie wollen das Standard-Passwort für die VNC-Server ändern. Dann geben Sie einfach folgendes ein.

```
ole@enterprise:~> vncpasswd
Password:
Verify:
```

Sie werden noch zweimal nach dem neuen Passwort gefragt und schon ist die Datei angelegt. Bei der Angabe eines anderen Dateinamens für die Passwortdatei muß der Name mit Pfad angegeben werden. So legt

```
ole@enterprise:~> vncpasswd public
Password:
Verify:
```

zwar eine Passwortdatei namens `public` an. Diese befindet sich aber im Heimatverzeichnis und nicht im Unterverzeichnis `.vnc`. Richtig ist folgender Befehl.

```
ole@enterprise:~> vncpasswd .vnc/public
Password:
Verify:
```

B.1.4 vncviewer

Der Client für den VNC-Server ist der `vncviewer`. Er wird in der Kommandozeile gestartet und stellt den Desktop mit Hilfe des X-Servers der Client-Maschine dar.

```
vncviewer [HOST][:NUMMER] [OPTIONEN]
```

Der `vncviewer` kann ohne Parameter aufgerufen werden. In diesem Fall fragt das Programm interaktiv nach dem Server und der Desktop-Nummer.

Optionen

`-shared`	Normalerweise kann sich nur ein Client mit dem VNC-Server verbinden; diese Option erlaubt mehrfache Verbindungen
`-display NUMMER`	Gibt den Desktop an, auf dem der Viewer sein Fenster darstellen soll
`-passwd DATEI`	Verwendet die angegebene DATEI als Passwortdatei anstatt den Benutzer nach dem Passwort zu fragen
`-viewonly`	Erlaubt keine Maus- oder Tastaturaktivitäten vom Client aus
`-fullscreen`	Startet den Viewer im Vollbildmodus
`-geometry GEO`	Hier kann die Geometrie und Position des Viewers angegeben werden.

Beispiele

Dieser Befehl ruft den VNC-Server mit der Nummer 10 auf dem lokalen Rechner auf.

```
vncviewer :10
```

Hier entgegen wird der Desktop 0 des Rechners *defiant* angesprochen. Dies ist identisch mit der Angabe `defiant:0`.

```
vncviewer defiant
```

Dieser Browser wird als “shared” im Betrachtungsmodus gestartet. Die Passworteingabe erfolgt über eine Passwortdatei.

```
vncviewer defiant:11 -shared -viewonly -passwd ~/.vnc/public
```

B.1.5 Beispiel: Einrichtung als Server für zwei Personen

Walter und Willi, die sich an verschiedenen Orten befinden, wollen gemeinsam an einem Programmierproblem arbeiten. Deswegen wollen beide an der Datei gleichzeitig arbeiten können.

Hierfür wird auf der Servermaschine von *walter* der VNC-Server gestartet und das Passwort aus der Datei `~/.vnc/work` verwendet.

```
walter@enterprise:~> vncserver :10 -alwaysshared -rfbauth ~/.vnc/work

New 'X' desktop is enterprise:10

Starting applications specified in /home/walter/.vnc/xstartup
Log file is /home/walter/.vnc/enterprise:10.log
```

Walter startet den Viewer auf seinem Rechner. Er braucht den Server nicht anzugeben, da er sich auf dem gleichen Rechner befindet.

```
walter@enterprise:~> vncviewer :10
VNC server supports protocol version 3.3 (viewer 3.3)
Password:
VNC authentication succeeded
Desktop name "walter's X desktop (enterprise:10)"
Connected to VNC server, using protocol version 3.3
```

Willi startet den Viewer auch auf seinem Rechner. Er muß den Server mit anzugeben.

```
willi@defiant:~> vncviewer enterprise:10
VNC server supports protocol version 3.3 (viewer 3.3)
Password:
VNC authentication succeeded
Desktop name "willi's X desktop (enterprise:10)"
Connected to VNC server, using protocol version 3.3
```

Nun können beide zur gleichen Zeit auf dem Server arbeiten und gemeinsam an ihrem Programm arbeiten.

B.1.6 Beispiel: Einrichtung als Klassenraumserver

Eine der interessantesten Aufgaben des VNC-Servers ist der Export eines X11-Desktops für den Untericht. Hier tritt das Problem auf, daß ein Betrachter, nämlich der des Dozenten, auf dem X-Server arbeiten muß, während die Kursteilnehmer nicht auf dem Server arbeiten dürfen. Leider gibt es keine Option, die eine solche Einstellung erlaubt. Das Setzen des Viewonly-Flags erfolgt nur über den Viewer.

Der Trick an der Sache ist es zwei X-Server aufzusetzen. Einen "Master"-Server auf dem gearbeitet werden darf und einen "Slave"-Server der nur einen Viewer auf den "Master"-Server enthält. Dieser Viewer ist auf Viewonly gestellt.

Mit diesem Start-Skript werden die Server gestartet.

Listing B.2 *vnc*

```
#!/bin/sh
# Startet zwei VNC-Server (Master und Slave)

# Auflösung festlegen
GEOM=950x680
CDEPTH=16

# Variable auswerten
```

```
case $1 in
  start)
        # Starten der VNC-Server
        # Master starten
        # Passworddatei ~/.vnc/privat
        vncserver :1 \
                -geometry $GEOM \
                -depth $CDEPTH \
                -alwaysshared \
                -name master \
                -rfbauth $HOME/.vnc/privat

        # Slave starten
        # Passworddatei ~/.vnc/public
        vncserver :2 \
                -geometry $GEOM \
                -depth $CDEPTH \
                -alwaysshared \
                -name slave \
                -rfbauth $HOME/.vnc/public

        # Viewer zur Kontrolle des Master-Servers starten
        # Falls in einem X-Terminal als anderer Benutzer gestartet zeigt
        # die Display-Variable auf das aktuelle X-Window
        # Der Server muß aber vorher mit
        # xhost localhost
        # freigegeben werden
        DISPLAY=:0.0
        export DISPLAY
        # Viewer starten
        vncviewer :1
        ;;

  stop)
        # Beenden des VNC-Servers
        vncserver -kill :1
        vncserver -kill :2
        ;;
  *)    # Falsches Kommando
        echo "Syntax: vnc start|stop"
        ;;
esac

# Ende
```

Mit dem Parameter `start` werden die zwei Server und der Master-Viewer gestartet.

```
ole@enterprise:~> vnc start

New 'master' desktop is enterprise:1

Starting applications specified in /home/ole/.vnc/xstartup
Log file is /home/ole/.vnc/enterprise:1.log


New 'slave' desktop is enterprise:2
```

```
Starting applications specified in /home/ole/.vnc/xstartup
Log file is /home/ole/.vnc/enterprise:2.log

VNC server supports protocol version 3.3 (viewer 3.3)
Password:
VNC authentication succeeded
Desktop name "ole's master desktop (enterprise:1)"
Connected to VNC server, using protocol version 3.3
```

Der Parameter `stop` sorgt dafür, daß die beiden Server beendet werden.

```
ole@enterprise:~> vnc stop
Killing Xvnc process ID 1994
Killing Xvnc process ID 2014
```

Vorm Einsatz des Skripts müssen mit

```
ole@enterprise:~> vncpasswd ~/.vnc/privat
Password:
Verify:
ole@enterprise:~> vncpasswd ~/.vnc/public
Password:
Verify:
```

zwei Passwortdateien angelegt werden. Die Kursteilnehmer erhalten natürlich nur das Passwort aus der Datei `public`.

Damit die Server unterschiedlich in ihrer Ausführung sind, muß die Datei `xstartup` die Server unterscheiden.

Listing B.3 *xstartup*

```
#!/bin/sh

# Start ist abhängig vom Servernamen
case $VNCDESKTOP in
  slave)
        # Slave-Server nur mit Viewer
        echo "Slave Modus: $VNCDESKTOP"
        vncviewer -viewonly -share -fullscreen \
                -passwd $HOME/.vnc/privat :1
        ;;

  master)
        # Master-Server mit KDE
        echo "Master Modus: $VNCDESKTOP"
        xrdb $HOME/.Xresources
        xsetroot -solid grey
        #xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
        kde &
        ;;
  *)
        # Normaler Server zum Arbeiten
        echo "Normaler Server"
        xrdb $HOME/.Xresources
        xsetroot -solid grey
        xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
        twm &
        ;;
esac
```

Anhang C

Open Publication License

C.1 Englische Version

1. **REQUIREMENTS ON BOTH UNMODIFIED AND MODIFIED VERSIONS**

   The Open Publication works may be reproduced and distributed in whole or in part, in any medium physical or electronic, provided that the terms of this license are adhered to, and that this license or an incorporation of it by reference (with any options elected by the author(s) and/or publisher) is displayed in the reproduction.

   Proper form for an incorporation by reference is as follows:

   > Copyright (c) 2000-2003 by Ole Vanhoefer. This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, v1.0 or later (the latest version is presently available at http://www.opencontent.org/openpub/).

   The reference must be immediately followed with any options elected by the author(s) and/or publisher of the document (see section 6).

   Commercial redistribution of Open Publication-licensed material is permitted.

   Any publication in standard (paper) book form shall require the citation of the original publisher and author. The publisher and author's names shall appear on all outer surfaces of the book. On all outer surfaces of the book the original publisher's name shall be as large as the title of the work and cited as possessive with respect to the title.

2. **COPYRIGHT**

   The copyright to each Open Publication is owned by its author(s) or designee.

3. **SCOPE OF LICENSE**

   The following license terms apply to all Open Publication works, unless otherwise explicitly stated in the document.

   Mere aggregation of Open Publication works or a portion of an Open Publication work with other works or programs on the same media shall not cause this license to apply to those other works. The aggregate work shall contain a notice specifying the inclusion of the Open Publication material and appropriate copyright notice.

   SEVERABILITY. If any part of this license is found to be unenforceable in any jurisdiction, the remaining portions of the license remain in force.

   NO WARRANTY. Open Publication works are licensed and provided "as is" without warranty of any kind, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose or a warranty of non-infringement.

4. **REQUIREMENTS ON MODIFIED WORKS**

   All modified versions of documents covered by this license, including translations, anthologies, compilations and partial documents, must meet the following requirements:

   (a) The modified version must be labeled as such.

   (b) The person making the modifications must be identified and the modifications dated.

   (c) Acknowledgement of the original author and publisher if applicable must be retained according to normal academic citation practices.

   (d) The location of the original unmodified document must be identified.

(e) The original author's (or authors') name(s) may not be used to assert or imply endorsement of the resulting document without the original author's (or authors') permission.

5. **GOOD-PRACTICE RECOMMENDATIONS**

   In addition to the requirements of this license, it is requested from and strongly recommended of redistributors that:

   (a) If you are distributing Open Publication works on hardcopy or CD-ROM, you provide email notification to the authors of your intent to redistribute at least thirty days before your manuscript or media freeze, to give the authors time to provide updated documents. This notification should describe modifications, if any, made to the document.

   (b) All substantive modifications (including deletions) be either clearly marked up in the document or else described in an attachment to the document.

   (c) Finally, while it is not mandatory under this license, it is considered good form to offer a free copy of any hardcopy and CD-ROM expression of an Open Publication-licensed work to its author(s).

6. **LICENSE OPTIONS**

   The author(s) and/or publisher of an Open Publication-licensed document may elect certain options by appending language to the reference to or copy of the license. These options are considered part of the license instance and must be included with the license (or its incorporation by reference) in derived works.

   (a) To prohibit distribution of substantively modified versions without the explicit permission of the author(s). SSubstantive modificationïs defined as a change to the semantic content of the document, and excludes mere changes in format or typographical corrections.

   To accomplish this, add the phrase 'Distribution of substantively modified versions of this document is prohibited without the explicit permission of the copyright holder.' to the license reference or copy.

   (b) To prohibit any publication of this work or derivative works in whole or in part in standard (paper) book form for commercial purposes is prohibited unless prior permission is obtained from the copyright holder.

   To accomplish this, add the phrase 'Distribution of the work or derivative of the work in any standard (paper) book form is prohibited unless prior permission is obtained from the copyright holder.' to the license reference or copy.

C.2 Deutsche Version

Inoffizielle deutsche Übersetzung des englischen Originals (von Stefan Meretz).

1. **ERFORDERNISSE FÜR UNMODIFIZIERTE UND MODIFIZIERTE VERSIONEN**

   Open-Publication-Arbeiten dürfen als Ganzes oder in Teilen reproduziert und verteilt werden, in beliebigen Medien, physisch oder elektronisch, vorausgesetzt, die Bedingungen dieser Lizenz gehören dazu, und diese Lizenz oder ein Verweis auf diese Lizenz (mit jeder Option, die von dem Autor / den Autoren und/oder dem Herausgeber gewählt wurde) wird in der Reproduktion angezeigt.

   Eine geeignete Form einer Aufnahme durch Verweis lautet wie folgt:

   > Copyright (c) 2000-2003 by Ole Vanhoefer. Dieses Material darf nur gemäß der Regeln und Bedingungen wie sie von der Open Publication Licence, Version v0.4, festgelegt werden, verteilt werden (die letzte Version ist gegenwärtig verfügbar unter http://www.opencontent.org/openpub/).

   Die kommerzielle Weiterverbreitung von Open Publication lizensiertem Material ist zu den aufgeführten Bedingungen ausdrücklich gestattet.

   Jegliche Publikation im Standard- (Papier-) Buch-Format erfordert die Zitierung der Original-Herausgeber und Autoren. Die Namen von Herausgeber und Autor/en sollen auf allen äußeren Deckflächen des Buchs erscheinen. Auf allen äußeren Deckflächen des Buchs soll der Name des Original-Herausgebers genauso groß sein wie der Titel der Arbeit und so einnehmend genannt werden im Hinblick auf den Titel.

2. **COPYRIGHT**

   Das Copyright jeder Open Publication gehört dem Autor / den Autoren oder Zeichnungsberechtigten.

3. **GÜLTIGKEITSBEREICH DER LIZENZ**

   Die nachfolgenden Lizenzregeln werden auf alle Open-Publication-Arbeiten angewendet, sofern nicht explizit anders lautend im Dokument erwähnt.

   Die bloße Zusammenfassung von Open-Publication-Arbeiten oder eines Teils einer Open-Publication-Arbeit mit anderen Arbeiten oder Programmen auf dem selben Medium bewirkt nicht, dass die Lizenz auch auf diese anderen Arbeiten angewendet wird. Die zusammengefaßte Arbeit soll einen Hinweis enthalten, die die Aufnahme von Open-Publication-Material und eine geeignete Copyright-Notiz angibt.

ABTRENNBARKEIT. Wenn irgendein Teil dieser Lizenz durch irgendeine Rechtsprechung außer Kraft gesetzt werden, bleiben die verbleibenden Teile der Lizenz in Kraft.

KEINE GEWÄHRLEISTUNG. Open-Publication-Arbeiten werden lizensiert und verbreitet "wie sie sind" ohne Gewährleistung jeglicher Art, explizit oder implizit, einschließlich, aber nicht begrenzt auf, der impliziten Gewährleistung des Vertriebs und der Geignetheit für einen besonderen Zweck oder eine Gewähleistung einer non-infringement.

4. **ERFORDERNISSE FÜR MODIFIZIERTE ARBEITEN**

   Alle modifizierten Versionen, die durch diese Lizenz abgedeckt werden, einschließlich von Übersetzungen, Anthologien, Zusammenstellungen und Teildokumenten, müssen die folgenden Erfordernisse erfüllen:

   (a) Die modifizierte Version muss als solche gekennzeichnet werden.

   (b) Die Person, die die Modifikationen vornimmt, muss genannt und die Modifikationen müssen datiert werden.

   (c) Danksagungen der Original-Autors und -Herausgebers - sofern vorhanden - müssen in Übereinstimmung mit der normalen akademischen Zitierungspraxis erhalten bleiben.

   (d) Der Ort des originalen unmodifizierten Dokuments muss benannt werden.

   (e) Die Namen der Original-Autoren dürfen nicht benutzt werden ohne die Erlaubnis des Original-Autors / der Original-Autoren.

5. **EMPFEHLUNGEN EINER GUTEN PRAXIS**

   In Ergänzung zu den Erfordernissen dieser Lizenz, wird von den Weiterverteilenden erwartet und ihnen stark empfohlen:

   (a) Wenn Sie Open-Publication-Arbeiten als Hardcopy oder auf CD-ROM verteilen, schicken Sie eine E-Mail-Ankündigung Ihrer Absicht der Weiterverteilung mindestens dreißig Tage bevor Ihr Manuskript oder das Medium endgültig festgelegt ist, um den Autoren Zeit zu geben aktualisierte Dokumente anzubieten. Die Ankündigung sollte die Änderungen beschreiben, die gegebenenfalls am Dokument vorgenommen wurden.

   (b) Alle substantiellen Modifikationen (einschließlich Löschungen) sind entweder im Dokument klar zu kennzeichnen oder sonst in einem Anhang zu beschreiben.

   (c) Schließlich, obwohl nicht erforderlich unter dieser Lizenz, ist es, eine vorgeschlagene gute Form eine kostenlose Kopie jedes Hardcopy- und CD-ROM-Ursprungs einer unter Open Publication lizensierten Arbeit dem Autor / den Autoren anzubieten.

6. **LIZENZ-OPTIONEN**

   Der/die Autor/en und/oder der Herausgeber eines unter Open Publication lizensierten Dokuments darf bestimmte Optionen durch Anhängen von Regelungen an den Lizenz-Verweis oder die Lizenz-Kopie wählen. Diese Optionen sind empfohlener Teil der Lizenzbestimmungen und müssen in abgeleiteten Arbeiten in die Lizenz eingefügt werden.

   (a) Verhindern der Verteilung von substantiell modifizierten Versionen ohne explizite Erlaubnis des Autors / der Autoren. "Substantielle Modifizierung" ist definiert als eine Änderung des semantischen Inhalts des Dokuments und schließt bloße Format-Änderungen oder typographische Korrekturen aus.

   Zur Anwendung fügen Sie den Satz 'Distribution of substantively modified versions of this document is prohibited without the explicit permission of the copyright holder' (Verbreitung von substantiell modifizierten Versionen dieses Dokuments ist ohne die explizite Erlaubnis des Copyright-Inhabers untersagt) dem Lizenz-Verweis oder der Lizenz-Kopie hinzu.

   (b) Verhindern jeglicher Veröffentlichung dieser Arbeit oder abgeleiteter Arbeiten im Ganzen oder in Teilen in Standard- (Papier-) Buchform für kommerzielle Zwecke ohne vorherige Erlaubnis durch den Copyright-Inhaber.

   Zur Anwendung fügen Sie den Satz 'Distribution of the work or derivative of the work in any standard (paper) book form is prohibited unless prior permission is obtained from the copyright holder' (Verbreitung dieser Arbeit oder abgeleiteter Arbeiten in Teilen in Standard- (Papier-) Buchform für kommerzielle Zwecke ohne vorherige Erlaubnis durch den Copyright-Inhaber ist untersagt) dem Lizenz-Verweis oder der Lizenz-Kopie hinzu.

Index