Vorlesung über Linux

Unvollständige Preview-Version.
Do not distribute

Im SS 2000

Prof. Dr. V. Aurich
Dipl. Phys. A. Beck

Abteilung für Informatik
Heinrich-Heine-Universität Düsseldorf
April 2000

Inhaltsverzeichnis

Kapitel 1

LINUX - Was ist das ?

Das Betriebssystem LINUX hat in den letzten Jahren zunehmend an Bedeutung gewonnen. Es findet zunehmend Anwendung in den verschiedensten Bereichen, von einfachen Workstations über Steuerungssysteme von Industrie- und Forschungsanlagen bis hin zu großen Serversystemen.

Kenntnisse im Umgang mit diesem System und insbesondere dessen Administration werden auf dem Arbeitsmarkt in zunehmendem Maße nachgefragt.

Diese Vorlesung soll Ihnen die notwendigen Grundkenntnisse im Umgang mit und der Administration von LINUX basierten Installationen vermitteln.

Das Ziel ist dabei nicht, die Bedienung von Anwenderprogrammen zu erläutern, sondern Problemlösungsstrategien zu vermitteln.

1.1 Die Geschichte von LINUX

Die Geschichte von LINUX beginnt im Sommer 1991 als "Hobbyprojekt" eines finnischen Informatikstudenten namens Linus Torvalds.

Ursprünglich handelte es sich um einige Zusatzroutinen zu "Minix", einem anderen UNIXähnlichen Betriebssystem des Betriebssystem-"Gurus" Andrew S. Tannenbaum.

Das Ergebnis dieser Arbeiten wurde auf dem Internet publiziert und dort von vielen interessierten Programmierern aufgegriffen und erweitert. Aufgrund der GNU General Public License (GPL), unter der der Quellcode verteilt wurde konnten all diese Erweiterungen in spätere Versionen eingebaut werden, so daß letztlich ein sehr mächtiges Gesamtsystem entstanden ist.

Aus einer Nachricht von Linux Torvalds vom 25. August 1991:

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
```

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

Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and
professional like gnu) for 386(486) AT clones.  This has been brewing
since april, and is starting to get ready.  I'd like any feedback on
things people like/dislike in minix, as my OS resembles it somewhat
(same physical layout of the file-system (due to practical reasons)
among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work.
This implies that I'll get something practical within a few months, and
I'd like to know what features most people would want.  Any suggestions
are welcome, but I won't promise I'll implement them :-)

              Linus (torvalds@kruuna.helsinki.fi)

PS.  Yes - it's free of any minix code, and it has a multi-threaded fs.
It is NOT protable (uses 386 task switching etc), and it probably never
will support anything other than AT-harddisks, as that's all I have :-(.
```

Die ersten Quelltexte erschienen dann Mitte September im Internet und am 5. Oktober Version 0.02 und bis zum Dezember erreichte die Versionsnummer dann bereits 0.11 und die erste “stabile” Version 0.12 erschien am 5. Januar 1992 und im April erfolgte der Versionssprung auf 0.96, der deutlich zeigte, daß das System inzwischen recht benutzbar geworden war.

Inzwischen steht Version 2.4.0 vor der Tür, und Linux ist dank ausgefeilter Distributionen und tausenden verfügbarer Applikationen inzwischen so leicht zu installieren, zu bedienen und universell einsetzbar, wie kommerzielle Systeme.

1.2 LINUX und UNIX

LINUX ist ein UNIX-artiges Betriebssytem. Das heißt, daß es sich in seiner inneren Struktur am POSIX-Standard orientiert, der beschreibt, welche Funktionen ein Betriebssystem den Anwendungsprogrammen zur Verfügung stellen soll.

In diesem Sinne sind alle Unixe "kompatibel" - es gibt zwar kleinere Unterschiede im Funktionsumfang oder der genauen Implementation und einigen Systembefehlen, aber grundsätzlich sollten Applikationen zwischen verschiedenen UNIXen wie LINUX, FreeBSD, OpenBSD, Solaris, HPUX, AIX, Unixware etc. leicht portierbar sein.

Wer mit einem UNIX umgehen kann, wird sich auch schnell in einem anderen zurechtfinden.

1.3 Wie sieht LINUX aus ?

UNIX besitzt traditionell eine textbasierte Oberfläche - was nicht weiter verwundert, wenn man das Alter dieses Betriebssystems bedenkt. Die Frage "Wie sieht LINUX aus ?", ist also gar nicht so leicht zu beantworten.

Das "Aussehen" definiert sich weniger durch eine bestimmte grafische Oberfläche, wie bei MacOS oder Windows, als vielmehr durch seinen inneren Aufbau und den damit verbundenen Eigenschaften.

Um diesen Aufbau zu verstehen, werden wir uns im nächsten Kapitel mit den Aufgaben eines Betriebssystemes und exemplarisch mit einigen Hardwareeigenschaften von Rechnersystemen beschäftigen, die Einfluß auf das Design eines Betriebssystemes haben.

Kapitel 2

Aufgaben eines Betriebssystemes

2.1 Die Grundaufgaben

Ein Betriebssytem soll es dem Benutzer eines Rechners ermöglichen, die angeschlossene Hardware (oder andere Ressourcen) anzusprechen, um so die Möglichkeiten der Hardware für Nutzerprogramme verfügbar zu machen.

Bei den ersten Rechnersystemen wurde die Hardware direkt von den Benutzerprogrammen angesteuert. Diese Methode ist nicht nur sehr mühselig, da sie den Programmierer quasi zwingt für jedes Programm "das Rad neu zu erfinden", sie ist auch fehleranfällig und bindet die entstandenen Programme an die vorhandene Hardware.

Eine der Aufgaben eines Betriebssystemes ist es daher, von der Hardware zu abstrahieren. Das heißt, daß Teile des Betriebssystemes (die sogenannten Gerätetreiber - englisch "device drivers") den Zugriff auf die Hardware übernehmen und den Nutzerprogrammen in Form einer einheitlichen Schnittstelle präsentieren.

Diese **Abstraktion** bedeutet einen großen Vorteil für die Anwenderprogramme: Die Programme müssen nicht mehr für verschiedene Rechnerkonfigurationen angepaßt werden. Man tauscht lediglich den Gerätetreiber aus, wenn die Rechnerkonfiguration sich ändert.

Die zweite wesentliche Aufgabe eines Betriebssystemes ist der **Schutz** des Systemes vor fehlerhaften Programmen.

Diese Aufgabe erscheint zunächst nicht völlig einsichtig: Wie soll man ein System vor fehlerhaften Programmen schützen, und warum sollte man dies wollen — es schränkt ja offensichtlich die Freiheit des Programmierers ein.

In der Tat ist es für sogenannte "single-user" Syteme, also Rechner mit nur einem Benutzer nicht ganz so wichtig, diese Schutzmechanismen zu haben, da letztlich eben nur dieser Benutzer **seine** Daten und **seine** Hardware durch solche fehlerhaften Programme beschädigen kann.

Beispiele von Betriebssystemen ohne solche Schutzfunktionen sind DOS sowie die DOS-basierten Versionen von Windows (bis 3.11).

Völlig einsichtig ist dieser Wunsch nach Schutz aber bei Mehrbenutzer- (multi-user-) Systemen. Schließlich möchte niemand z.B. die Datei mit seiner Diplomarbeit verlieren, weil jemand anderes einen Programmierfehler gemacht hat.

Wird ein solcher Schutz implementiert, so ist es zumeist ohne hohen Zusatzaufwand möglich, ausserdem **Zugriffsbeschränkungen** einzuführen, die z.B. verhindern, daß andere Benutzer Dateien lesen, die nicht für sie bestimmt sind, oder auf Geräte zugreifen, für die sie keine entsprechende Berechtigung besitzen.

Erst in einem solchen Umfeld macht die Benutzung eines einzelnen Rechnersystems durch mehrere Personen (ob nun gleichzeitig oder nacheinander) Sinn.

Wird ein Rechnersystem **gleichzeitig** von mehreren Personen benutzt, so ergibt sich ein weiteres Problem: Bestimmte **Ressourcen**, wie die verfügbare Rechenleistung, der Hauptspeicher, die Festplatten etc. müssen zwischen den einzelnen Benutzern gerecht verteilt werden.

Das Betriebssystem erhält damit weitere Aufgaben:

1. **Arbitrierung von Ressourcen,** also die Zuteilung verfügbarer Geräte bzw. Dienstleistungen an die einzelnen Benutzer. z.B. wird der Zugriff auf ein Bandlaufwerk gesperrt, während ein anderer Benutzer damit arbeitet, bzw. das Wechseln einer CD gesperrt, solange noch Benutzer darauf zugreifen.

2. **Virtualisierung.** Viele Ressourcen können scheinbar "gleichzeitig" von mehreren Benutzern angesprochen werden, indem man sie jeweils einem von Ihnen der Reihe nach zur Verfügung stellt. So werden zum Beispiel die Festplatten von allen Nutzern gemeinsam verwendet, wobei zwar nur jeweils ein Benutzer zur gleichen Zeit zugreifen kann, das Betriebssystem aber dafür sorgt, daß alle Nutzer (entsprechend ihren Rechten und Prioritäten) zum Zuge kommen.

Um nun das Systemdesign verstehen zu können, das UNIX wählt, um diese Anforderungen zu befriedigen, betrachten wir im folgenden einige Grundlagen der Ansteuerung von Hardware.

2.2 Wie funktioniert ein Hardwarezugriff ?

Die Steuerung von Peripheriegeräten ist, wenn man die obigen Grundaufgaben eines Betriebssystemes betrachtet, das Hauptproblem bei der Realisierung..

Solche Hardwarezugriffe erfolgen durch sogenannte “Portzugriffe”. Das heißt, daß der Prozessor durch Lesen und Schreiben spezieller “Speicherzellen” die Hardware anweist, eine bestimmte Aktion (z.B. das Bewegen des Lesekopfes eines Laufwerkes, das Umschalten einer Leitung auf einem IO-Port, etc.) auszuführen.

Ein einzelner solcher Zugriff löst allerdings normalerweise nur sehr wenig “Aktion” aus, so daß nützliche Aktionen eine ganze Reihe von Zugriffen erfordern.

Hier liegt bereits der erste Vorteil der Verwendung eines Betriebssystemes (im Gegensatz zur direkten Hardwaresteuerung durch die Anwendungsprogramme):

Der Gerätetreiber des Betriebssystems faßt gängige Zugriffssequenzen zusammen und erleichtert damit das Schreiben von Anwendungen. So müssen z.B. zur Ausgabe eines einzelnen Zeichens auf einem PC-Druckerport mindestens 3 Portzugriffe und eine Warteschleife verwendet werden, wohingegen man bei gängigen Betriebssystemen mit einem einzigen Systemaufruf ganze Zeichenketten ausgeben kann, wobei das System auch noch auf mögliche Fehler (z.B. PaperOut,OffLine) prüft.

Diese Vereinfachung vermindert auch das Fehlerrisiko bei der Anwendungserstellung, da häufig gebrauchte, aber oft kniffelig zu codierende Routinen bereits im Systemtreiber enthalten sind.

Außerdem wird vermieden, daß Programmierfehler auftreten, die einen Systemabsturz verursachen können, weil z.B. Fehlerabfragen vergessen worden sind.

2.3 Das Konzept von Unix

Ein einfaches Betriebssystem wird sich nun damit zufriedengeben, den Hardwarezugriff wie oben beschrieben zu vereinfachen.

Strebt man aber auch einen **geschützten** Betrieb an, so ergibt sich ein neues Problem. Selbst angenommen, die Betriebssystemroutinen sind korrekt programmiert und damit absturzsicher, so bleibt es den Anwenderprogrammen überlassen, diese Funktionen zu verwenden, oder aber selbst auf die Hardware zuzugreifen. Besonders alte DOS-Spiele bedienten sich dieser Technik, um durch den direkten Zugriff unter Umgehung des Betriebssystemes eine höhere Performance zu erreichen.

Um geschützten Betrieb zu erreichen, muß direkter Hardwarezugriff verhindert werden. Mit frühen Prozessoren, wie dem 65xx und auch dem original PC/XT Chip 8086 war dies nicht möglich. Ein Anwenderprogramm konnte jederzeit alle Befehle verwenden und damit auch direkte Hardwarezugriffe ausfuehren.

In späteren Prozessoren wurde daher ein sogenannter “Protected Mode” (geschützter Modus) eingeführt. In diesem Modus dürfen Anwenderprogramme nicht mehr alle Befehle ausfuehren, insbesondere nicht Befehle, die auf externe Hardware zugreifen oder das System zum Absturz bringen könnten.

Das Betriebssystem selbst muß natürlich weiterhin solche Befehle ausführen können, damit überhaupt noch ein Peripheriezugriff möglich ist. Aus diesem Grund führt man ein Privilegiensystem ein, das bestimmten Programmen erweiterte Rechte einräumt.

Die einfachste Form eines solchen Privilegiensystems ist ein einfaches "alles oder nichts": Es gibt Programme mit vollen Hardwarezugriffsrechten, und solche ohne diese Rechte.

Auf diesem System basiert der traditionelle Aufbau eines UNIX-ähnlichen Betriebssystemes:

Es teilt die Rechte auf in den **Systemkern (Kernel)** und die Anwendungsprogramme. Alle elementaren Betriebssystemfunktionen werden vom Kernel übernommen, von dem die Anwenderprogramme alle benötigten Dienste anfordern.

[Anm.: In den meisten real existierenden Unix-Systemen wird diese strenge Grenze zwischen Kern und Anwendungen etwas "aufgeweicht". Man führt "Module" ein, die es ermöglichen zur Laufzeit Treiber ins System zu integrieren, und besonders privilegierte Anwenderprogramme ("Daemonen"), die bestimmte Steuerfunktionen übernehmen, die nicht zwingend die besonders hohen Privilegien des Kerns benötigen.]

Mit einem solchen System ist es nun möglich alle oben beschriebenen Grundfunktionen eines Betriebssystems zu erfüllen:

- Die Hardwaretreiber im Kernel sorgen für die **Abstraktion**.
- Die Privilegientrennung erlaubt dem Kernel, seine **Schutz**funktion auszuüben.
- Da alle Ressourcen über den Kernel angesprochen werden, kann er diese **arbitrieren** oder **virtualisieren**.

Kapitel 3

Installation von Linux

Die Installation von Linux ist dank ausgefeilter Distributionen wie SuSe, RedHat, Caldera, Debian etc. inzwischen sehr einfach geworden.

Hier werden keine speziellen Installationshinweise für einzelne Distributionen gegeben, sondern allgemeine Hinweise für die Installation eines Betriebssystemes auf PC-Hardware.

3.1 Die Wahl der Distribution

Die verbreitetsten LINUX-Distributionen in Deutschland sind die SuSe, RedHat und Debian. Grundsätzlich kann man nicht sagen, daß eine Distribution besser oder schlechter ist - sie unterscheiden sich in Umfang und den Administrationsutilities.

Für den absoluten Einsteiger ist vermutlich die SuSe Distribution am geeignetsten. Sie zeichnet sich durch den reichhaltigen Umfang und ein sehr einfaches Administrationssystem (YAST) aus.

Genau das sind auch die Nachteile. Der große Umfang erschwert die Auswahl, sofern man nicht ca. 10GB nur für Linux verfügbar hat, und YAST ist zwar sehr einfach zu bedienen, aber auch problematisch, wenn man versucht, nicht in YAST vorgesehene Konfigurationen einzustellen - dabei ergeben sich oft Konflikte.

Die RedHat ist erheblich kompakter und benutzt Linuxconf als Konfigurationstool. Damit entstehen weniger Entscheidungsschwierigkeiten bei der Paketauswahl (auf Kosten von weniger mitgelieferter Software) und Linuxconf kommt besser mit manuellen Konfigurationsänderungen zurecht wie YAST, ist dafür aber etwas schwieriger zu bedienen.

Debian wird besonders von "harten" Anhängern der freien Software genutzt. Einige Konfigurationstools sind nicht ganz so ausgereift wie bei den kommerziellen Gegenstücken, dafür gibt es auch einige sehr gute Tools wie apt-get. Ebenfalls fehlen dort einige Anwendungen, die keine freie Lizenz besitzen.

All allgemeine Daumenregel gilt: Für die erste Linuxinstallation ist SuSe aufgrund der sehr einfachen Installation eine gute Wahl. Der etwas fortgeschrittenere Anwender wird RedHat oder Debian bzw. eine der davon abgeleiteten Distributionen wie die Mandrake bevorzugen.

3.2 Installationsvorbereitung

3.2.1 Backup machen

Bevor man ein neues Betriebssystem auf einen Rechner aufspielt, gilt als allerwichtigste Grundregel: Sofern sich bereits Daten auf dem Rechner befinden, muß unbedingt ein Backup angefertigt werden. Auch wenn es technisch möglich ist, ein Betriebssystem nachträglich ohne Datenverlust auf der bestehenden Installation einzurichten, bleibt immer ein Restrisiko, insbesondere durch mögliche Bedienfehler während der Installation.

Also bevor man mit der Installation beginnt unbedingt ein Backup anfertigen, testen ob dieses auch intakt ist, und sicherstellen, daß alles notwendigen Zubehör (wie Bootdisketten, Seriennummern, Original-CDs) für ein Zurückspielen des Backups vorhanden sind.

3.2.2 Partitionen

Verschiedene Betriebssysteme verwalten den Speicherplatz auf der Festplatte völlig unterschiedlich. Die Installation eines Betriebssystems geht daher nicht ganz so einfach von statten wie die Installation einer Anwendung.

Der vorhandene Platz auf einer Festplatte muß zwischen den einzelnen Betriebssystemen aufgeteilt werden.

Dazu teilt man die Festplatte in sogenannte Partitionen auf. Eine Partition beschreibt einen zusammenhängenden Bereich der Festplatte, der einem Betriebssystem zugeordnet wird. Ein einzelnes Betriebssystem kann sich über mehrere Partitionen verteilen, aber zumeist können nicht mehrere Betriebssysteme auf einer Partition gleichzeitig installiert sein.

Eine Ausnahme bilden hier miteinander kompatible Betriebssysteme, wie DOS und Windows95/98 oder sogenannte "Huckepackinstallationen", wie z.B. eine Linuxinstallation auf dem sog. UMSDOS-Dateisystem.

Zum Verständnis des Partitioniervorgangs kurz einige Worte zum Aufbau einer Festplatte. Die ersten Festplatten bestanden aus Stapeln von Magnetscheiben, auf deren Oberfläche mit Hilfe von Schreib-/Leseköpfen die Daten abgelegt werden. Die einfachste Möglichkeit die Zweidimensionalität der Scheibenoberfläche auszunutzen, besteht darin, die Scheiben rotieren zu lassen und den Schreib-/Lesekopf nach innen und außen zu bewegen.

Durch diese Konstruktion ergab sich eine natürliche Aufteilung der Daten, die auch benutzt wurde, um die Daten "wiederzufinden", sie zu adressieren.

Da jede Scheibe zwei Seiten hat, und man zumeist Stapel aus mehreren solchen Scheiben verwendet, ist das erste Kriterium die Oberfläche, auf der die Daten zu finden sind, bzw. der Lesekopf (**head**), der zu dieser Oberfche gehört.

Das zweite ergibt sich aus der Position des Kopfes zur Achse des Plattenstapels. Man spricht von einer Spur (**track**) bzw. einem Zylinder (cylinder). Letzterer Begriff ergibt sich, wenn man sich die Figur vorstellt, die die Spuren auf den einzelnen Oberflächen zusammen ergeben.

Auch innerhalb einer Spur gibt es eine weitere Einteilung in sogenannte Sektoren (**sectors**). Dabei handelt es sich eben um Winkelsegmente.

Im PC-Bereich werden in einem Sektor normalerweise 512 Byte Nutzdaten gespeichert. Dazu kommen noch Fehlerkorrekturdaten und Markierungen, die das Wiederfinden der Sektoren ermöglichen.

Die Kapazität einer Platte berechnet sich also zu 512 Byte pro Sektor × Anzahl Sektoren pro Spur × Anzahl Spuren pro Kopf × Anzahl Köpfe.

[Anm.: Bei modernen Platten haben die auf der Platte angegebenen Werte normalerweise nichts mehr mit der tatsächlichen Geometrie der Platte zu tun. Die Ansteuerung stammt eben immer noch aus "uralten Zeiten", zu denen man kompatibel bleiben möchte. Heutige Platten benutzen variable Sektorzahlen - je nach Länge der Spur und sehr hohe Zylinderzahlen, die aber nach außen oft durch große Kopfzahlen wiedergegeben werden, um zu alten Programmen kompatibel zu bleiben. Die sinnvollere Lösung wäre das Konzept von SCSI gewesen, das jedem Plattensektor lediglich eine laufende Nummer zuweist (lineare Adressierung).]

Die Daten werden normalerweise so angeordnet, dass zunächst die Sektoren innerhalb einer Spur beschrieben werden, und dann als nächstes der Kopf gewechselt wird. Dies ist lediglich ein elektronischer Umschaltvorgang, der sehr schnell ist. Das Verändern der Kopfposition erfordert eine mechanische Bewegung, die ungleich länger dauert.

Daher erfolgt die Partitionierung von Platten **zylinderweise**, um eine optimale Performance für das gerade gewählte Betriebssystem sicherzustellen, da Kopfbewegungen so minimiert werden.

Das Tool, mit dem man Platten partitioniert, heißt traditionell "fdisk". Es steht sowohl unter DOS/Windows zur Verfügung als auch unter Linux.

Man wird zumeist mit dem Linux-Tool arbeiten wollen, da es wesentlich mächtiger ist. Das DOS-Pendant kann z.B. keine Linux-Partitionen anlegen, wohingegen Linux-fdisk durchaus DOS-Partitionen anlegen kann.

Es gibt alternative Programme wie Partition Magic, Disk Druid etc., die teilweise sehr nützliche Zusatzfunktionen haben, wie das zerstörungsfreie Verkleinern oder Vergrößern einer Partition. Auf solche Funktionen sollte man sich allerdings nie verlassen.

Die "sauberste" Methode bleibt immer noch die komplette Neupartitionierung. Ich selber pflege neue Platten zunächst in relativ viele Partitionen einzuteilen. Diese können auch nachträglich noch einzelnen Betriebssystemen zugeordnet werden, wobei eben nur die Daten auf diesen Partitionen verlorengehen, wohingegen eine Veränderung von Partitionsgrössen gewöhnlich alle beteiligten Partitionen unbrauchbar macht.

Was kennzeichnet nun eine Partition ? Werfen wir einfach einen Blick auf eine Partitionstabelle, wie sie das Linux-Tool fdisk mit dem Befehl "p"rint ausgibt:

```
Festplatte /dev/sda: 255 Koepfe, 63 Sektoren, 527 Zylinder
Einheiten: Zylinder mit 16065 * 512 Bytes

   Geraet boot.  Anfang      Ende    Bloecke  Id  Dateisystemtyp
/dev/sda1   *         1       128   1028128+   6  FAT16
/dev/sda2           129       511   3076447+   5  Erweiterte
/dev/sda3           512       527    128520   82  Linux Swap
/dev/sda5           129       256   1028128+   6  FAT16
/dev/sda6           257       358    819283+  83  Linux
/dev/sda7           359       435    618471   83  Linux
/dev/sda8           436       511    610438+  83  Linux
```

Es handelt sich hier um eine 4 GB SCSI Platte im System, auf dem dieses Script entstanden ist. Die Platte meldet sich mit 255 Köpfen (was natürlich glatt gelogen ist), 63 Sektoren und 527 Zylindern an.

Diese 527 Zylinder können nun auf verschiedene Betriebssysteme verteilt werden. Dazu gibt es eine Tabelle im allerersten Sektor der Festplatte, den sogenannten **Master-Boot Record (MBR)**. Da dieser allerdings neben den Partitionierungsdaten auch noch das Programm speichern muß, mit dessen Hilfe der Rechner startet, geht es dort recht beengt zu, und es haben dort lediglich 4 Einträge Platz. Nun sehen Sie auf dem obigen Beispiel aber mehr als 4 Einträge.

Man erkannte bald, daß lediglich 4 Einträge etwas wenig waren, und schuf daher den speziellen Partitionstyp "Erweiterte Partition", der weitere Einträge aufnehmen kann.

Wie sieht nun ein Partitionseintrag aus ?

Jeder Eintrag erhält von Linux eine Nummer bzw. ein sogenanntes Gerät ("block-device") zugeordnet. Primäre Partitionen erhalten Nummern von 1-4, sogenannte logische Partitionen, die sich auf der erweiterten Partition befinden, werden ab 5 aufwärts gezählt.

Sie sehen im obigen Beispiel, daß die Platte in 3 primäre Partitionen (sda1-sda3) aufgeteilt ist, von denen eine die erweiterte Partition ist. Diese enthält die logischen Partitionen (sda5-sda8). Beachten Sie, daß dort der Bereich, den die erweiterte Partition belegt, quasi noch einmal "unterverteilt" wird.

Linux ist es übrigens egal, ob es in einer primären oder einer erweiterten Partition abgelegt ist — im Gegensatz zu DOS/Windows, das unbedingt von einer primären Partition starten will, aber erweiterte Partitionen durchaus als Datenlaufwerke akzeptiert.

Zu jeder Partition gehören nun noch bestimmte Daten, die sie eindeutig charakterisieren.

Das erste Feld im obigen Beispiel ist das "boot-flag". Dieses sollte nur bei genau einer Partition auf der ersten Platte gesetzt sein, und es bestimmt, von welcher Partition und damit welches Betriebssystem gebootet wird. Nach der Installation von Linux stimmt diese Aussage aber unter Umständen nur noch begrenzt, da Sie zumeist mit Linux auch einen Bootloader installieren werden, der dieses Flag ignoriert und statt dessen den Benutzer fragt.

Natürlich gehört zu einer Partition auch die Information, wo sie beginnt und endet. Es ist hier theoretisch möglich, Überlappungen anzugeben, was aber sinnvollerweise von allen fdisk Programmen verhindert wird, da zwei verschiedene auf einen Festplattenbereich zugreifende Betriebssysteme ein heilloses Chaos anrichten können. Hingegen ist es möglich, "Lücken" zu lassen, und dies kann auch sinnvoll sein, z.B. um Platz für ein später zu installierendes Betriebssystem zu lassen.

Die oben ausgewiesene Blockzahl ist nicht im Partitionsblock gespeichert, sondern dient lediglich dem Anwender zur Orientierung. Aus diesem Grund wird auch nicht die Sektorenzahl ausgewiesen, sondern vielmehr 1kB große Blöcke.

Ein weiteres wichtiges Datum ist die Partitions-ID, die in der obigen Tabelle sowohl in ihrer numerischen Form, als auch in eine Klartextform (unter Dateisystemtyp) angegeben wird. Mit Hilfe dieser ID finden manche Betriebssysteme beim Startvorgang automatisch die Partition, auf der sich die weiteren Dateien befinden. Diese Technik erscheint auf den ersten Blick recht nützlich, hat aber den Haken, daß man nicht 2-mal das gleiche Betriebssystem völlig unabhängig voneinander auf derselben Platte installieren kann. DOS und Windows verwenden diese Technik (Windows-NT nicht), während Linux die Partitions-ID ignoriert und nur seine eigenen Konfigurationsdaten verwendet.

Dennoch gibt es zwei für Linux reservierte Partitions-IDs - Hexadezimal 82 und 83 (hier sieht man Linux seine Wurzeln in Minix an, das hex 80 und 81 verwendet). Das hat den Sinn, anderen Betriebssystemen zu signalisieren, daß diese Daten nicht für sie bestimmt sind. Würde man einen für DOS "bekannten" Typ eintragen wie z.B. ID 06, würde DOS versuchen diese Partition als "Laufwerk" einzubinden, dann aber bei Zugriffen auf dieses Laufwerk Fehler melden, da es die völlig andere Datenstruktur auf dem Linuxlaufwerk nicht "versteht".

Analysieren wir also einmal obigen Partitionsblock: Die Platte enthält 3 primäre Partitionen, sda1-3. sda1 ist vom Typ FAT16 (einem DOS-Dateisystem) und hat eine Größe von einem Gigabyte. Diese Partition beherbergt eine Windows95-Installation. sda2 enthält die erweiterte Partition von 3GB, auf die wir gleich noch zu sprechen kommen. sda3 ist vom Typ 0x82 (Linux Swap) und lediglich 128MB groß. Wie Sie leicht erraten, werden auf einer so kleinen Partition zumeist keine Daten gespeichert werden.

Eine Swappartition dient als eine Art "erweiterter Hauptspeicher". Wenn nicht mehr alle laufenden Programme gleichzeitig in das RAM des Rechners passen, werden einzelne Programme zeitweise "ausgelagert" - auf eben diese Swap-Partitionen (es können mehrere solche Partitionen angelegt werden).

Linux ist zwar auch in der Lage eine Datei zum Swappen zu benutzen, wie Windows dies tut, aber das Swappen in eine Partition ist effizienter, da eine Partition auf der Festplatte zusammenhängend ist, so daß weniger Kopfbewegungen nötig sind. Entsprechend wurde unter Windows 3.11 die Swapdatei auch nach Möglichkeit zusammenhängend angelegt (wenn man permanente Swapdatei ausgewählt hatte). Ob und wie groß man einen Swapbereich bzw. andere Partitionen wählt, behandeln wir später.

Kommen wir nun zum Inhalt der erweiterten Partition. sda5 enthält eine weitere DOS-Partition von 1GB, die dort als Laufwerk D: auftaucht. sda6-8 sind jeweils 3 etwas kleinere Linuxpartitionen.

Auf einem bisher "Linux-freien" System wird man leider selten eine solche Situation antreffen, in der man bereits eine linuxtaugliche Partitionierung vorfindet — dann heißt es erst einmal "Platz schaffen".

3.2.3 Platz schaffen

Bevor man also ein Linuxsystem installieren kann, muß Platz dafür auf der Platte vorhanden sein. Es gibt nun viele Methoden, diesen Platz zu schaffen — je nach vorhandener Vorkonfiguration.

Im einfachsten Fall hat man ein "leeres" System vor sich. Das heißt es befindet sich noch kein Betriebssystem auf der Festplatte, oder man ist bereit, es zu opfern und ebenfalls neu zu installieren.

Prinzipiell könnte man nun einfach der Reihe nach alle gewünschten Betriebssysteme installieren. In der Praxis zeigen einige Systeme (insbesondere Windows) allerdings einige

“Unarten”, die diese direkte Variante nicht empfehlenswert machen.

Eine bewährte Vorgehensweise besteht darin, zunächst die Linuxinstallation zu starten und unter Linux die Partitionierung vorzunehmen. Damit vermeidet man, daß Windows allzu besitzergreifend wird und die komplette Platte ohne Rückfrage und Eingriffsmöglichkeit für sich besetzt.

[Allerdings gibt es auch hier einen kleinen Haken: Befand sich vor der Umpartitionierung eine einzige große Windowspartition auf der Platte und sieht man für die Windowspartition die Partition Nummer 1 vor, so kann dies Windows verwirren, da es noch einmal intern die Größe der Partition speichert. Falls die Installation von Windows mit seltsamen Fehlermeldungen bei der Formatierung der Windows-Partition abbricht, hilft zumeist der Linux-Befehl “dd if=/dev/zero of=/dev/hda1 count=100” (wobei noch hda1 durch die entsprechende Partition ersetzt werden muß). Um diesen Befehl auch ausführen zu können, muß allerdings erstmal das Unix laufen. Bei einigen Distributionen kann man allerdings auch an geeigneter Stelle aus dem Installationsvorgang ausbrechen um diesen Befehl abzusetzen.]

Wenn außer Linux noch ein weiteres Betriebssystem auf der Platte installiert werden soll, ist zu empfehlen, die Linuxinstallation nach der Partitionierung abzubrechen, und erst einmal die anderen Betriebssysteme zu installieren. Der Grund hierfür liegt darin, daß manche Systeme gerne den Linux-eigenen Boot-Loader üeberschreiben, so daß Linux hinterher nicht mehr zu starten ist. Linux dagegen rechnet mit einem parallel installierten Betriebssystem und installiert einen Boot-Loader, der dem Benutzer die Wahl läßt, welches System er starten will.

Also bei “leerem System”: Installation von Linux starten, nach Wunsch partitionieren, Linuxinstallation abbrechen, erst andere Betriebssysteme, dann Linux installieren.

Eine geringfügige Variation des “leeren Systems” stellt die Installation von Linux auf einer zweiten Platte dar. Dabei stellt sich allerdings in den meisten Fällen die Frage nach der Installation eines anderen Betriebssystemes nicht, da außer Linux nur wenige Systeme von einer anderen als der ersten Platte starten können. Ansonsten gilt aber dieselbe Vorgehensweise wie oben.

Kommen wir nun zu den häufigeren Situationen - Rechner bei denen der gesamte vorhandene Plattenplatz bereits einem anderen System zugeordnet ist. Es gibt in diesem Fall zwei mögliche Konfigurationen:

1. Der gesamte Platz auf der Platte befindet sich in einer einzigen Partition (unter Windows gibt es nur das Laufwerk C:).

   In diesem Fall wird es besonders schwierig den benötigten Platz zu schaffen. Mit Tools wie Partition Magic oder FIPS ist es zwar im Prinzip möglich, diese existierende Partition zu schrumpfen, aber wie bereits erwähnt, sollte man vor solchen Aktionen unbedingt ein Backup machen. Ist es auf diese Weise gelungen, Platz für Linux zu reservieren, kann man mit der Linuxinstallation forfahren wie oben. Wenn nein, bleibt nur die Möglichkeit einer kompletten Neuinstallation, bei der man dann natürlich gleich Platz für Linux reserviert.

2. Der Platz auf der Platte bzw. den Platten ist auf einige Laufwerke verteilt.

   In diesem Fall kann man auf einem Laufwerk Platz schaffen und es für Linux verwenden. Aber Achtung: Windows vergibt seine Laufwerksbuchstaben in einer festgelegten Reihenfolge und viele Programme merken sich die Position von Programmbestandteilen, inklusive Laufwerksbuchstabe. Die einzige Partition, die man relativ "sicher" umwidmen kann, ist die, die den letzten Laufwerksbuchstaben trägt. Von dieser muß man dann alle noch benötigten Daten retten (dort installierte Programme installiert man meist besser neu, ihre intern gemerkten Pfade stimme nach einer Verschiebung zumeist nicht mehr, was zu Problemen führt).

   Damit man auch die richtige Partition erwischt, kurz ein paar Worte zur Vergabe von Laufwerksbuchstaben durch Windows:

   Windows beginnt bei der ersten Platte mit der Vergabe. Die primäre DOS-Partition dieser Platte erhält den Laufwerksbuchstaben C:. Danach werden der Reihe nach alle Platten durchlaufen, und die primären DOS-Partitionen als Laufwerke hinzugefügt.

   Dann erfolgt ein erneuter Durchgang durch alle Platten, wobei nun die logischen Laufwerke auf erweiterten Partitionen hinzugefügt werden.

   Um einigermaßen sicher testen zu können, ob Sie auch die richtige Partition erwischt haben, machen Sie erst einmal einen Testlauf. Löschen Sie die Partition nicht, sondern ändern Sie nur Ihre ID auf 0x83. Dadurch wird die Partition für DOS unsichtbar. Booten Sie dann wieder Windows und sehen Sie nach, ob Sie den gewünschten Effekt erreicht haben, sprich, ob die "richtige" Partition bzw. das richtige Laufwerk nun verschwunden ist.

   Danach können Sie die Partition wirklich löschen und wie sonst auch durch neue Partitionen ersetzen.

Wie startet man nun die Installation von Linux ?

3.3 Booten der Linux-Installation

Wir haben im letzten Kapitel bereits mehrfach angenommen, daß sie die Linux-Installations-CD irgendwie zum "laufen" bekommen.

Wie dies im einzelnen geht, sollte sich im Handbuch der jeweiligen Distribution finden - wir besprechen hier dennoch einige Standardtechniken, die man in verschiedenen Situationen anwendet.

3.3.1 Booten vom CDROM

Auf neueren Rechnern und mit aktuellen Distributionen ist die sicherste und einfachste Methode des Installationsstartes das Booten von CDROM. Dazu muß man im allgemeinen die BIOS-Einstellungen des Rechners verändern, da man gewöhnlich nicht von CDROM bootet.

Die Vorgehensweise unterscheidet sich je nach einsetztem Plattencontroller. IDE-Platten und CDROMs werden zumeist vom BIOS des Mainboards kontrolliert, da der IDE-Controller heute dort integriert ist.

Nach dem Einschalten des Rechners werden Sie zumeist für kurze Zeit eine Aufforderung der Form “Press [Del] to enter SETUP” sehen. Drücken Sie während dieser Anzeige auf die Taste [Entf]. Sie landen dann in einem Menügeführten System, mit dem Sie die Rechnereinstellungen ändern können. ACHTUNG: Diese Einstellungen sind **sehr wichtig** für das korrekte Funktionieren des Rechners und teilweise sogar für das Wohlergehen der Hardware. Ändern Sie auf keinen Fall Einstellungen, von denen Sie sich nicht absolut sicher sind, was sie bedeuten.

Um vom CDROM booten zu können, benötigen sie meist den Menüpunkt [Advanced BIOS Setup]. Dort finden sich Einstellungen wie [Boot Sequence] oder [First Boot Device]. Sorgen Sie dafür, daß CDROM dort an erster Stelle steht.

Das Setup wird mit dem Punkt “Save and Exit” verlassen, wobei noch einmal eine Sicherheitsabfrage stattfindet. Denken Sie daran, daß das BIOS keine Ahnung vom deutschen Tastaturlayout hat, bei dem y und z gegenüber dem US-Layout vertauscht sind. Sie werden also “z” drücken müssen, um “yes” zu sagen.

Bei SCSI-Systemen müssen Sie das BIOS des Hostadapters bemühen. Es wird davon ausgegangen, daß Besitzern von SCSI-Systemen dieser Vorgang geläufig ist. Eine gängige Stolperfalle ist hier, daß das SCSI-BIOS das Booten von CDROM (wie bei IDE auch) durch Erzeugen eines “Diskettenlaufwerkes” simuliert. Es kann daher notwendig sein, im Haupt-BIOS die Bootsequenz auf “Diskette zuerst” einzustellen.

Legen Sie nun die CD in das Laufwerk und starten Sie den Rechner neu - es sollte nach kurzer Zeit der Startbildschirm der jeweiligen Distribution erscheinen.

Falls diese Methode nicht funktioniert (z.B. weil der Rechner recht alt ist, und kein CD-Boot unterstützt), bleiben noch zwei andere Möglichkeiten.

3.3.2 Starten via DOS

Eine ist die Installation via DOS. Die meisten Distributionen besitzen ein DOS-basiertes Startsystem. Dazu befindet sich auf der CD irgendwo eine Datei die meist setup.exe oder ähnlich heißt. Starten Sie dieses Programm von DOS oder Windows aus und folgen Sie den Instruktionen. Genaueres zu dieser Installationsart finden Sie in den Handbüchern zu Ihrer Distribution.

3.3.3 Booten von Disk

Die andere Methode ist die Benutzung von Bootdisketten. Auch diese sollte in den Handbüchern Ihrer Distribution erklärt sein. Entweder sind Startdisketten im Umfang der Distribution enthalten, oder man muß sie selbst erstellen. Dazu benutzt man ein Utility namens rawrite, um speziell formatierte Dateien (Diskimages) auf eine oder mehrere Disketten zu kopieren, von denen dann gebootet wird.

Natürlich muß dazu in jedem Fall im BIOS als Bootsequenz "Diskette zuerst" eingestellt sein.

3.4 Die ersten Installationsschritte vor der Partitionierung

Diese sind recht distributionsspezifisch. Zumeist beantwortet man einige einfache Fragen zur gewünschten Sprache, dem Tastaturlayout, und ob ein Farbmonitor vorhanden ist bzw. ob eine grafische Installation verwendet werden soll. Wählen Sie im Zweifel die textbasierte Installation - sie ist meist ausgereifter.

Wenn Sie nach Hardware und deren Konfiguration gefragt werden, versuchen Sie immer, Linux die Hardware suchen zu lassen. Das kann es zumeist sehr sehr gut — zumindest findet es den Interrupt von NE2000 Karten von selbst, statt wie ein anderes Betriebssystem grundsätzlich blind 3 einzutragen, was fast immer falsch ist.

Nur bei älterer ISA Hardware müssen Sie evtl. einige Daten, wie die Basisadresse wissen. Sehen Sie in der Dokumentation nach, in der Windows-Systemsteuerung (unter System—Gerätemanager), oder fragen Sie jemanden, der sich mit sowas auskennt.

3.5 Partitionieren

Wann und warum des Partitionierens haben wir ja bereits behandelt - kommen wir nun zum "wie". Die Partitioniertools unterscheiden sich wieder je nach Distribution, einige lassen einem sogar die Wahl. "Alte Hasen" bevorzugen hier den Klassiker fdisk, da er sehr viele Freiheiten läßt, dem Anfänger sei eher ein anderes Tool, wie z.B. Disk-Druid empfohlen. Wählen Sie einfach den Defaultwert der Distribution - der ist meist am anfängerfreundlichsten.

Aber Achtung: die Wahl von "automatisches Partitionieren" tut meist auch wirklich genau das, und läßt keinen Platz für andere Betriebssysteme.

Wie viele und welche Partitionen sollte man nun für Linux anlegen ?

Auch hierauf gibt es keine einheitliche Antwort. Die erste Frage ist: Will/brauche ich eine Swap-Partition ?

Generell ist eine Swap-Partition empfehlenswert. Sie ermöglicht mehr Programme gleichzeitig zu benutzen als in den Arbeitsspeicher passen.

Allerdings sollte sich die Swap-Partition in einer vernünftigen Relation zum Arbeitsspeicher bewegen. Es macht im Normalfall relativ wenig Sinn, mehr für die Swap-Partition anzulegen als man Hauptspeicher hat. Ausnahmen hiervon sind Systeme mit sehr wenig RAM (RAM plus SWAP sollte mindestens um die 40MB ergeben, um arbeiten zu können) und Systeme die oft größere Applikationen fahren, die zwar gleichzeitig offen sind, aber selten gleichzeitig genutzt werden.

Als Daumenregel hat sich "SWAP-size=RAM-size" bewährt.

Die zweite Frage ist: auf wie viele Partitionen will ich meine Linuxinstallation verteilen ?

Genau wie bei Windows ist es nicht nötig mehrere Partitionen (Laufwerke) anzulegen - man kann alles in eine große Partition speichern.

Warum also sollte man mehrere Partitionen anlegen ?

Es hat Vorteile bei der Datensicherung. Legt man sich selten ändernde Bereiche (z.B. installierte Programme) getrennt von sich häufig ändernden (z.B. eigene Dateien) ab, so kann man die Häufigkeit der Sicherung einzelner Bereiche entsprechend anpassen.

Unix kennt allerdings keine Laufwerksbuchstaben, wie Windows. Alle Partitionen finden sich in einem einzigen Verzeichnisbaum, wobei man allerdings frei entscheiden kann, wo eine einzelne Partition dort erscheint.

Eine schon relativ umfangreiche, aber bewährte Variante der Partitionierung besteht in einzelnen Partitionen für folgende Pfade:

/ Die sogenannte Wurzel (root) des Dateisystems. Dort befinden sich alle Dateien, die nicht in die noch folgenden Kategorien fallen. Insbesondere sind dort üblicherweise alle zum Hochlauf (booten) des Systems notwendigen Dateien. Sofern man die unten angegebenen weiteren Partitionen anlegt, muss dieser Bereich nicht sehr gross sein. So ca. 100MB genügen. Legt man eine dieser Partitionen nicht an, so muß die dort angegebene Größe jeweils hier zugeschlagen werden. Bei älteren Rechnern kann hier noch eine Eigenheit der Plattenansteuerung Probleme machen. So ist es mit älteren BIOSen nicht möglich, Daten von Zylindern mit Nummern größer 1024 zu lesen. Das BIOS wird von Linux zwar im laufenden Betrieb nicht benötigt, da dann die Hardware direkt angesprochen wird, wohl aber zum Booten. Da die zum Booten nötigen Dateien sich zumeist auf der root-Partition befinden, sollte man versuchen, diese unterhalb der 1024er-Grenze zu plazieren. Ist dies nicht möglich, gibt es allerdings auch noch einige Ausweichtechniken, auf die wir später noch zu sprechen kommen.

/usr In diesem Pfad werden alle Programme und deren fixe Daten gespeichert. Je nach verwendeter Distribution und gewähltem Umfang benötigt man hierfür zwischen ca. 1GB und 20GB. Durch bewußtes "Abspecken" der zu installierenden Programme kann man allerdings bis auf einige hundert Megabyte herunterkommen. Für ein gut nutzbares System mit allen wichtigen Programmen kommt man normalerweise mit 2GB aus. Legen Sie diese Partition im Zweifel als letztes und dann so gross wie möglich an.

/var Hier legt das System häufig veränderliche Dateien ab, wie z.B. die EMail der Benutzer, die Newsartikel, laufende Druckaufträge, etc. Hier kommt es stark auf die geplante Nutzung an. Für einen Privatanwender werden hier einige hundert MB genügen, möchte man eine größere Site mit z.B. vielen Newsartikeln oder vielen Nutzern mit hohem Mailaufkommen betreiben, so muß man hier entsprechend mehr einkalkulieren.

/home Hier — und nur hier — dürfen Benutzer Ihre eigenen Dateien ablegen - nirgends anders besitzt ein normaler Nutzer die nötigen Rechte. Je nach den von Ihnen verwendeten Anwendungen sollten Sie um die 100MB pro "aktivem" Benutzer einkalkulieren, der das System häufig und vielfältig nutzt. Gelegenheitsbenutzer dürften mit ca. 20MB gut bedient sein, "Power-user", die selbst Programme schreiben oder umfangreiche Dokumente erstellen, benötigen auch schon einmal einige hundert MB.

Falls es sich um Ihre erste Linuxinstallation handelt, probieren Sie erst einmal die "Einfachvariante": Legen Sie nur eine Partition an, die Rootpartition (/). Als Größe sollte man mindestens ca. 1GB, besser 2GB einkalkulieren, damit man beim Aufspielen der Software nicht vor der Qual der Wahl steht, was man weglassen soll bzw. darf. Mehr schadet natürlich nie.

Vergessen Sie ggf. nicht, Platz für andere Betriebssysteme zu reservieren, wenn Sie solche zusätzlich installieren wollen.

3.6 Andere Betriebssysteme aufspielen

Wenn die Partitionierung beendet ist, empfiehlt es sich, falls nötig zunächst einmal die Installation von Linux abzubrechen, um andere Betriebssysteme zu installieren. Der Grund für diese seltsam anmutende Vorgehensweise wurde ja bereits erläutert.

Oft ist kein explizites Abbruchkommando vorgesehen. Drücken Sie in diesem Fall, nachdem die Partitionierungsdaten gespeichert wurden, die Tastenkombination für den Warmstart des PCs, Strg-Alt-Entf (Ctrl-Alt-Del). Es kann ein bißchen dauern, bis der Rechner darauf mit einem Reboot antwortet. Falls die Distribution auf diese Tastenkombination nicht reagiert, drücken Sie den Reset-Taster. Da zu diesem Zeitpunkt noch keine Dateien auf die Festplatte gespielt wurden, brauchen Sie keine Bedenken zu haben, daß Daten beschädigt werden könnten.

Die anderen Systeme werden dann wie gewohnt installiert. Ist dieser Vorgang abgeschlossen, so beenden Sie die anderen Betriebssysteme korrekt und schalten Sie den Rechner aus und wieder ein. Beginnen Sie nun erneut mit der Linuxinstallation — sie sollten jetzt allerdings bereits eine korrekte Partitionierung vorfinden, an der Sie nun nichts mehr ändern dürfen.

3.7 Paketauswahl

Die Paketauswahl fällt insbesondere dem Anfänger sehr schwer. Verlassen Sie sich zunächst auf Ihre Distribution und wählen Sie ein Standard-Setup. Die meisten Distributionen bieten inzwischen vorkonfigurierte Varianten für bestimmte Anwendungen an. Wählen Sie dort z.B. “personal Workstation”.

Paßt das gewählte Modell nicht in die angelegten Partitionen, so werden Sie von der Distribution gewarnt und gebeten Ihre Auswahl anzupassen.

Dann gibt es einige gute Kandidaten, die Sie mit an sicherheit grenzender Wahrscheinlichkeit nie brauchen werden. Das beste Beispiel sind Manuals und Hilfetexte (HOWTOs) in Sprachen, die Sie nicht sprechen. Schalten Sie diese Pakete am besten in allen Sprachen außer Ihrer Muttersprache und Englisch ab.

Es empfiehlt sich immer, die englischen Varianten auch zu behalten, da manchmal die Übersetzung nicht völlig geglückt ist.

Die Hilfetexte zu Themen, die Sie (noch) nicht interessieren, können Sie auch erst einmal ausschalten.

Weitere Bereiche, die Sie evtl. ausklammern können, wenn Sie sie nicht schon kennen, sind TEX (ein sehr gutes Textsatzsystem, mit dem z.B. dieses Script erstellt wurde), sowie viele Anwendungen für X-Windows und die meisten Programmiersprachen, außer C, C++ und den gängigen Scriptsprachen TCL/TK und Perl, da letztere von einigen Systemutilities verwendet werden.

Die Distributionen warnen gewöhnlich, wenn Sie versuchen sollten, sehr wichtige Teile des Systems auszuschalten.

Deaktivieren Sie keinesfalls die Netzwerkkomponenten, auch wenn Sie kein Netzwerk besitzen. Unix bedient sich für viele Funktionen des "localnet", eines Pseudonetzes innerhalb einer Maschine. Außerdem ermöglicht Ihnen das localnet fast alle Netzadministrationsfunktionen auszuprobieren, ohne ein "echtes" Netz aufbauen zu müssen.

Auf ein "richtiges" Linuxsystem gehören auch unbedingt die Kernelquellen. Wozu diese dienen erfahren Sie später in diesem Kurs.

3.8 Weitere Installationsfragen

Die Distribution wird nach oder während der Installation der Pakete noch einige Fragen zur gewünschten Systemkonfiguration und persönlichen Einstellungen wie dem Rechnernamen stellen. Zumeist sind hier sinnvolle Startwerte eingestellt.

Treten Fragen auf, bei denen Sie nicht wissen, wie Sie antworten sollen, so werfen Sie erst einmal einen Blick in die Dokumentation der jeweiligen Distribution. Oft finden Sie dort eine ausreichende Erklärung. Im Zweifel akzeptieren Sie einfach den Defaultwert der Distribution.

3.8.1 Netzwerkkonfiguration

Wenn Ihr Rechner mit dem Internet verbunden ist (nicht zu empfehlen, bis Sie sich gut mit dem System auskennen und es entsprechend gegen Angriffe schützen können), tragen Sie bei der Netzkonfiguration die entsprechenden Daten (Rechnername, Domain, IP-Adresse, Netzmaske, Gateway) ein, die Ihnen Ihr Netzwerkverwalter genannt hat.

Installieren Sie für ein "privates" Netzwerk (einige miteinander vernetzte Rechner), so wählen Sie bitte IP-Adressen aus den "freien" Bereichen 10.x.x.x (Netzmaske 255.0.0.0) bzw. 192.168.x.x (Netzmaske 255.255.255.0). Diese Bereiche werden im Internet garantiert nicht benutzt, so daß sich bei einem späteren Anschluß einzelner Rechner an das Internet keine Konflikte ergeben.

3.8.2 Benutzer anlegen

Üblicherweise werden Sie gebeten, Passwörter für mindestens zwei Benutzer anzulegen: "root" und einen von Ihnen frei wählbaren.

Root ist der sogenannte Superuser (in anderen Systemen auch Administrator genannt), der **alle** Rechte besitzt. Im Verlauf dieses Kurses werden wir oft als root arbeiten, um Verwaltungsaufgaben zu erledigen.

Für **alles andere** sollten Sie unbedingt als **normaler Benutzer** arbeiten. Das schützt Sie davor, durch eine kleine Unachtsamkeit großen Schaden anzurichten. Unix nimmt die Tatsache, daß root alle Rechte besitzt, sehr ernst - Sie können ohne Sicherheitsabfrage beliebig viel zerstören. Als normaler Benutzer können Sie allenfalls Ihre eigenen Dateien gefährden.

3.8.3 X Windows konfigurieren

Neuere Distributionen konfigurieren auch direkt die grafische Oberfläche (X-Windows).

Neuere Grafikkarten werden dabei automatisch erkannt. Allerdings kann es sein, daß für exotischere oder sehr neue Karten keine Treiber vorhanden sind. In seltenen Fällen kann es auch zu fatalen Fehlkonfigurationen (wenn Karten falsch erkannt werden) kommen, bei denen das System hängenbleibt.

Die sicherste Methode besteht darin, diesen Teil der Installation zu uebergehen und sich erst später, wenn alles andere läuft, damit zu beschäftigen.

Wollen Sie dennoch X direkt installieren, beachten Sie bitte folgende Ratschläge:

- Wenn Ihre Grafikkarte nicht automatisch erkannt wird, lassen Sie äußerste Sorgfalt walten, wenn Sie manuelle Einstellungen treffen. Eine falsch eingestellte Grafikkarte kann das System zum Absturz bringen oder in seltenen Fällen zur Zerstörung der Grafikkarte führen.

- Falls Ihre Grafikkarte in verschiedenen Versionen verfügbar ist (z.B. mit externen DACs oder Clocks, wie viele S3-basierte Karten), überprüfen sie genauestens, ob die richtige Version gewählt wurde. Diese Zusatzchips lassen sich nur sehr schlecht detektieren und können bei falscher Einstellung zur Zerstörung der Grafikkarte führen.

- Da die XFree Server normalerweise kein DDC beherrschen, müssen Sie einige Informationen zu Ihrem Monitor bereithalten, wie die maximale Horizontal- und Vertikalfrequenz. Diese finden Sie in Ihrem Monitorhandbuch.

3.8.4 Startmodus festlegen

Neuere Distributionen bieten an, Linux direkt mit der grafischen Oberfläche zu starten.

Es empfiehlt sich jedoch, dies zumindest zunächst nicht zu tun. Funktioniert etwas beim Systemstart nicht korrekt, so ist dies auf der Textmodusoberfläche leichter zu diagnostizieren und manche Fehler können auch dazu führen, daß die grafische Oberfläche nicht korrekt startet. In diesem Fall muß man unter Umständen sehr tief in die Trickkiste greifen, um wieder Zugriff auf das System zu erhalten.

3.8.5 Konfiguration des Bootloaders

Eine der letzten Stationen bei der Installation ist die Konfiguration des Bootloaders. Es handelt sich dabei um ein Programm, das beim Systemstart ausgeführt wird und es ermöglicht, zwischen verschiedenen Betriebssystemen zu wählen. Unter Linux kommt zumeist LILO zum Einsatz. Falls Linux nicht das einzige Betriebssystem ist, vergessen Sie nicht, Einträge für die anderen Systeme hinzuzufügen - sonst können diese nicht gebootet werden.

Meist wird auch angeboten, eine Bootdisk zu erstellen. Machen Sie davon Gebrauch. Gerade die Installation von LILO kann bei bestimmten BIOSen recht kniffelig sein, und eine Bootdisk ist die einzige Möglichkeit, das System bei einer misslungenen LILO-Installation wieder zu starten und dann den vorherigen Zustand wiederherszustellen bzw. LILO dann korrekt zu installieren.

Die dritte Alternative zum Booten von Linux außer über Bootloader wie LILO oder einer Bootdisk besteht in LOADLIN. LOADLIN lädt einen Linux-Kernel aus einer laufenden Windows-Umgebung und startet ihn. Wenn Sie Linux nur sehr selten verwenden oder LILO Probleme macht, ist LOADLIN eine Alternative. Dazu muss allerdings der Linux-Kernel auf die DOS-Partition kopiert werden. Alles weitere finden Sie in der Dokumentation zu LOADLIN.

Wenn alles soweit geklappt hat, sollten Sie nun beim nächsten Reboot (wobei Sie natürlich die Installations-CD aus dem Laufwerk entfernen und auch das Booten von CD/Disk/... im BIOS wieder ausschalten sollten) auf die eine oder andere Weise Linux booten können:

- Bei installiertem LILO bleibt das System beim Bootvorgang kurz stehen und erlaubt Ihnen nach dem sogenannten Boot-Prompt (LILO:) das zu bootende System anzugeben. Drücken Sie die Tabulatortaste, um alle LILO bekannten Systeme aufzulisten.

- Mit einer Bootdisk erreichen Sie ein Booten von Linux durch Einlegen dieser Disk in das Floppylaufwerk und Einstellen der Bootreihenfolge im BIOS, so daß zunächst von Floppy gestartet wird.

- Mit LOADLIN starten Sie wie gewöhnlich Windows und wechseln dann mit Hilfe von LOADLIN nach Linux.

Kapitel 4

Die Benutzung von Linux

Wir nehmen im folgenden an, daß Sie ein erfolgreich installiertes Linuxsystem (auf dem Sie root-Rechte haben) besitzen.

4.1 Einloggen

Nach dem Booten des Systems, bei dem sehr viele Statusmeldungen über erfolgreich (oder auch nicht) gestartete Systemdienste angezeigt werden, die Sie zunächst einmal ignorieren können, sollten Sie einen Bildschirm mit einem sogenannten Login-Prompt vorfinden:

```
Welcome to Linux Version x.y.zz, Distribution xyz.

login:
```

Bevor Sie mit dem Rechner arbeiten können, müssen Sie sich anmelden. Die Anmeldung ist zwingend, sie kann nicht umgangen werden. Auch gibt es normalerweise keine "Gastaccounts". Solche Zugänge kann man zwar einrichten, es empfiehlt sich aber aus Sicherheitsgründen nicht.

Durch die Eingabeaufforderung `login:` bittet Sie der Rechner, Ihren Benutzernamen einzugeben. Nachdem Sie dies getan haben (und die Eingabe mit [Enter] abgeschlossen), antwortet der Rechner mit `Password:`.

Dabei ist es völlig egal, ob der Account existiert oder nicht. Dieses Verhalten verhindert, daß ein Angreifer auf diese Art und Weise Kenntnis über die (Nicht-)Existenz einzelner Benutzer erhält.

Bei der Eingabe von Benutzername und Paßwort funktionieren die Editiertasten (Backspace, Delete, Cursor etc.) zumeist noch nicht korrekt. Wenn Sie sich vertippt haben, drücken Sie einfach [Enter] und starten von vorne.

Keine Angst, Linux wird Ihren Zugang nicht z.B. nach 3 Fehleingaben sperren, da man auf diese Art und Weise sehr leicht ein System für einen Benutzer unzugänglich machen könnte (eine sogenannte DoS - Denial of Service - Attacke), indem man sich wiederholt unter seinem Namen mit falschen Paßwort einzuloggen versucht.

Noch ein Hinweis: Unix unterscheidet grundsätzlich Groß- und Kleinschreibung. Die Benutzer MeinUser, Meinuser, meinuser und MEINUSER sind verschieden. Achten Sie bei der Eingabe von Benutzernamen, genauso wie bei sonstigen Eingaben unter Unix unbedingt auf die richtige Groß- und Kleinschreibung. Wenn Sie sich mysteriöserweise nicht einloggen können, hilft u.U. ein Blick auf den Status der ShiftLock-Taste.

4.2 Benutzer anlegen

Sie sollten wie schon erwähnt unbedingt einen "normalen" Benutzer anlegen, als der Sie normale Arbeiten erledigen. Sofern Sie das nicht schon bei der Installation getan haben, holen Sie dies als erstes nach:

Dazu loggen Sie sich als "root" ein und benutzen das Kommando `useradd -m meinuser` (wobei sie meinuser durch einen beliebigen Benutzernamen ersetzen dürfen, der Ihnen gefällt - er sollte nur aus Groß- und Kleinbuchstaben und Ziffern bestehen und nicht mehr als 8 Zeichen lang sein). Sollte Ihre Linuxdistribution das Kommando useradd nicht kennen, werden Sie etwas weiter hinten im Script nachschauen müssen, wie man per Hand Benutzer anlegt.

Ein auf diese Weise neu angelegter Nutzer ist gesperrt, indem sein Paßworteintrag auf einen unmöglichen Wert gesetzt ist. Vergeben Sie also ein neues Paßwort durch Eingabe von `passwd meinuser`.

Sie werden nun zweimal nach einem neuen Paßwort für den Benutzer meinuser gefragt. Die doppelte Abfrage dient als Schutz vor Tippfehlern — sich zweimal identisch zu vertippen ist schon relativ schwierig. Da Sie Administratorprivilegien besitzen (sie haben sich als root angemeldet), dürfen Sie Paßwörter anderer Nutzer ändern und müssen auch nicht das alte Paßwort kennen.

Wenn alles geklappt hat, loggen Sie sich aus. Arbeiten Sie so wenig wie möglich als root. Arbeiten als root heißt arbeiten ohne Netz und doppelten Boden — niemand schützt Sie vor sich selbst. Werden Sie nur root, um administrative Aufgaben zu erledigen, die unbedingt root-Privilegien erfordern und loggen Sie sich danach sofort wieder aus.

Zum Ausloggen geben sie einfach `exit` ein.

4.3 Einloggen als normaler Benutzer

Versuchen Sie, sich nun als der eben oder bei der Installation angelegte Benutzer einzuloggen. Wenn sich dabei Probleme ergeben, versuchen Sie, den Benutzer wie eben beschrieben neu anzulegen (Sie erhalten eine Fehlermeldung, wenn der Nutzer schon existiert) und das Passwort neu zu setzen.

Wenn sie als normaler Benutzer eingeloggt sind, sollten Sie einen sog. Kommandoprompt sehen, der je nach Voreinstellungen der Distribution recht verschieden aussehen kann:

```
[meinuser@meinrechner meinuser]$ _
meinuser$ _
meinrechner$ _
meinuser@meinrechner$ _
bash$ _
```

Er sollte aber immer auf `$` enden - damit wird angezeigt, daß ein normaler Benutzer aktiv ist. Root hat hier statt dessen das Zeichen `#`. Dieser Unterschied ist als optische Erinnerung gedacht, damit man sich weder wundert warum man bestimmte Administratorkommandos nicht ausführen kann, noch versehentlich zu lange als root arbeitet.

Loggen sie sich am besten testweise noch einmal aus und wieder ein. Zum Ausloggen geben sie einfach `exit` ein.

4.4 Das Hilfesystem

Wir möchten hier nicht alle Unix-Kommandos mit den zugehörigen Erklärungen auflisten, sondern zeigen, wie Sie selbst die Kommandos, die Sie benötigen, finden und Informationen über deren Benutzung erhalten können.

4.4.1 Manpages

Zu jedem Unixkommando sollten sogenannte “Manpages” installiert sein, also Bedienungsanleitungen.

Sie erhalten diese Anleitungen, indem Sie `man [Befehlsname]` eingeben.

Es gibt Kommandos, die in mehreren Kontexten Bedeutung haben (z.B. als Befehl einer Programmiersprache und als Shellbefehl). In diesem Fall verwenden Sie die Option “-a” im Aufruf von “man”, also `man -a [Befehlsname]`.

Probieren Sie es ruhig einmal mit einigen gängigen Kommandos aus, z.B. mit `pwd, cp, ls, mv` und `rm`.

Sie landen in einem sogenannten “Pager”, einem Programm zur seitenweisen Anzeige der Gesamtdatei. Sie können mit den Cursortasten zeilenweise durch den Text gehen oder mit der Leertaste und “b” seitenweise blättern. Mit “q” beenden Sie die Anzeige der aktuellen Hilfeseite.

4.4.2 Das Info-System

Für kompliziertere Programme ist das Manpage-System oft unzureichend. Da alles auf einer einzigen Seite dargestellt ist, wird es oft unübersichtlich und es ist schwer Kreuzreferenzen zu anderen Programmen etc. leicht zu verfolgen. Solche Referenzen sind zwar unter `SEE ALSO` jeweils angegeben, aber man muss zurück zur Kommandozeile und ein neues Man-Kommando eingeben, um sie zu betrachten.

Das Info-System ist diesen Anforderungen besser gewachsen. Es erlaubt die Dokumente zu strukturieren, sie in einzelne Kapitel aufzuteilen, Hinweise auf verwandte Themen unterzubringen usw. . Aufgerufen wird es ganz ähnlich wie die Manpages: “`info [Befehlsname]`”

Der Pager ist etwas anders aufgebaut, versteht aber im wesentlichen die gleichen Kommandos. Die Cursortasten bewegen hier allerdings einen “echten” Cursor, da man im Info-System auch Querverweise verfolgen kann. Dazu bewegt man den Cursor auf den entsprechenden Eintrag und drückt `Enter`.

4.4.3 apropos

Oft weiß man nicht genau, wie das gewünschte Kommando heißt. Für diesen Fall gibt es das Kommando `apropos`. Es durchsucht die Titel aller vorhandenen Manpages nach den angegebenen Stichworten und zeigt eine Liste mit Treffern an.

Haben Sie auf diese Weise das gewünschte Kommando gefunden, können Sie die dazugehörige Manpage wie gewohnt aufrufen. Die Zahl, die in Klammern auf den Namen des Kommandos folgt, gibt die sogenannte “Section” an, aus der dies Page stammt. Diese Bereiche sind:

1 **Benutzerkommandos** Alle Befehle, die ein normaler Benutzer verwenden darf. Leider sind nicht immer alle Befehle auch mit Manpages versehen.

2 **Systemaufrufe** beschreiben die Schnittstelle zwischen den Applikationen un dem Systemkern. Sie werden beim Programmieren, insbesondere systemnaher Software benötigt.

3 **Bibliotheksfunktionen** Die Funktionen, die von installierten Programmbibliotheken zur Verfügung gestellt werden. Auch diese werden zum Schreiben von Programmen

benötigt. Das Manpage-System ersetzt mit den Sektionen 2 und 3 die Hilfe, die in DOS/Windows traditionell in der Oberfläche des jeweiligen Compilers integriert ist.

4 **Geräte** enthält Zusatzinformationen zu verschiedenen Gerätetreibern - z.B. wie bestimmte Sonderfunktionen aktiviert werden können etc.

5 **Dateiformate** Das Verhalten von UNIX/Linux wird mit Hilfe vieler Konfigurationsdateien gesteuert. Die Formate (Aufbau, erlaubte Befehle, Optionen) dieser Dateien sind hier dokumentiert.

6 **Spiele** Programme mit weniger ernster Anwendung sind hier beschrieben. Es handelt sich nicht nur um Spiele, sondern auch oft um "Spielereien" wie Bildschirmschoner o.ä.

7 **Diverses** Alles, was nicht so recht in eine der anderen Kategorien paßt.

8 **Administrationskommandos** Befehle, die nur dem Administrator zur Verfügung stehen bzw. fast ausschließlich vom ihm genutzt werden.

n **Neues** Solle eigentlich leer sein. In der Praxis findet man aber Einträge für so ziemlich alles dort. Insofern ähnelt es etwas Sektion 7.

Diese Sektionen kann man beim Aufruf von `man` direkt mit angeben. Vergleichen Sie z.B. einmal die Ausgabe von `man 1 printf` mit der von `man 3 printf`.

4.4.4 /usr/doc

In diesem Verzeichnis und dessen Unterverzeichnissen finden Sie eine ganze Reihe nützlicher Hilfetexte, die zumeist sehr ausführlich auf ein Thema eingehen.

Um diese Texte lesen zu können benötigen Sie einige grundlegende Kenntnisse darüber, wie man sich im Unix-Verzeichnisbaum bewegt.

UNIX kennt keine Laufwerksbuchstaben wie z.B. DOS. Es organisiert sich in einem einzigen sogenannten Verzeichnisbaum. Dieser hat sein Wurzel im Verzeichnis "/", in das man mit `cd /` gelangt.

Von dort aus gelangt man mit `cd [Verzeichnisname]` in die höheren Ebenen. Um eine Stufe zurückzugehen, benutzt man `cd ..`.

Beispiel:

Um nach `/usr/doc` zu gelangen können sie eingeben:

```
cd /
cd usr
cd doc
```

oder auch direkt `cd /usr/doc` . Um zurück ins Wurzelverzeichnis zu gelangen, benutzen sie z.B. entweder zweimal `cd ..` oder auch `cd ../..` oder direkt `cd /`. Wenn Sie einen "Pfad" (so nennt man eine solche Verzeichnisangabe, da man daran den "Weg" ablesen kann, der genommen werden soll) eingaben, der mit `/` beginnt, so geht die Suche vom Wurzelverzeichnis aus, ansonsten vom aktuellen Verzeichnis (das Sie mit dem Kommando `pwd` abfragen können).

Wollen Sie nun eine Datei betrachten, so haben Sie mehrere Möglichkeiten:

cat Dieser Befehl zeigt eine Datei auf dem Bildschirm an. Allerdings eignet er sich nur für sehr kurze Dateien, da er nicht seitenweise anhält.

less Dieses Programm ist der schon von den Manpages her bekannte Pager (es wird von man intern verwendet). Andere UNIXe enthalten evtl. den `less` Befehl nicht, sondern nur den weniger mächtigen Vorgänger `more`

mc Wer den Norton Commander von DOS her kennt, dem wird evtl. der Midnight Commander gefallen. Auch dieser besitzt einen "View"-Modus, mit dem man gut Dateien betrachten kann. Gerade dem unerfahrenen Administrator sei der `mc` auch zum Löschen, Umbenennen und Kopieren von Dateien empfohlen, da durch die grafische (Textmode) Oberfläche einige gängige Fehler bei der Verwendung von Wildcards vermieden werden, da man die Auswirkungen noch *rechtzeitig* sieht.

Editoren Natürlich kann man auch seinen gewohnten Texteditor verwenden, wenn man sich bereits mit einem auskennt. UNIX besitzt eine Reihe sehr mächtiger Editoren, die allerdings teilweise etwas eigenwillig in der Bedienung sind - wir werden später einige der wichtigsten Editoren kurz besprechen.

Doch nach diesem Exkurs in die Dateibehandlung unter UNIX zurück zum `/usr/doc`-Verzeichnis und seinen Hilfedateien. Sie werden hier vermutlich nicht direkt Texte finden, sondern einige Unterverzeichnisse, die die Hilfen weiter untergliedern:

howto enthält Schritt-für-Schritt-Anweisungen für viele Themen, oft auch in mehreren Sprachen. Beachten Sie aber, daß viele Dokumente nicht übersetzt wurden - wenn Sie also eine Datei nicht auf Deutsch finden, werfen Sie einen Blick in die englischen Howtos.

packages enthält jeweils ein Directory für jedes Programmpaket, das auf dem Rechner installiert ist und für das weitere Dokumentation außer den Man- bzw. Info-Pages geliefert werden soll. Zumeist sind diese Zusatzinformationen von eher technischer Natur und beschäftigen sich mit "Advanced Topics" die für den einfachen Umgang mit dem jeweiligen Paket nicht unbedingt notwendig sind. Sie sind aber insbesondere interessant, wenn man ein Programm auf die eigenen persönlichen Vorlieben anpassen will.

Dateiendung	**Format**	**Betrachtungsprogramm**
.txt .doc .asc	Text	`less`,`more`,`cat`, Editoren
.ps	Postscript	`gv` oder `ghostview` (X11)
.html .htm	HTML (WWW-Format)	`lynx` (Textmode) oder `netscape` (X11)
.tex	TeX/LaTeX	`latex` übersetzt nach .dvi meist finden Sie aber bereits fertige .dvi-Dateien im selben Verzeichnis.
.dvi	TeX-DeVice-Independent	xdvi (X11)
.man .1 9	Manual pages	`nroff -man [Dateiname] |less`

Tabelle 4.1: Dateiendungen und Betrachterprogramme für Hilfetexte

Je nach Distribution finden Sie noch weitere Unterverzeichnisse, die distributionsspezifische Dokumente (wie z.B. das Handbuch) enthalten.

Die meisten Dokumente sind einfache Textdateien, einige sind allerdings auch in anderen Formaten gespeichert, siehe Tabelle 4.1.

Dort sind einige Programme erwähnt, die die graphische Benutzeroberfläche X11 benötigen. Wie diese konfiguriert, gestartet und benutzt wird, erfahren wir noch in späteren Kapiteln.

4.4.5 Weitere Quellen für Hilfe

Der Distributor

Die namhaften Distributoren bieten oft eine Hotline oder zumindest Hilfedatenbanken auf dem Internet an. Das Handbuch Ihrer Distribution sollte Hinweise auf entsprechende Angebote enthalten.

Besondere Erwähnung empfiehlt die SuSe Support Datenbank, die sehr viele gängige Fragen eingehend behandelt und recht gut gepflegt wird: `http://sdb.suse.de/sdb/de/html/index.html`

Andere Benutzer

Linuxbenutzer sind zumeist sehr erfreut, andere Linuxbenutzer zu treffen und können sich im allgemeinen stundenlang mit ihnen über die gemeinsame Leidenschaft unterhalten. Wenn Sie also jemanden kennen, der schon länger mit Linux arbeitet, wird er Ihnen oft weiterhelfen können oder evtl. jemanden kennen, der es kann.

User Groups/Benutzergruppen

In vielen Städten haben sich Linux User Groups (LUGs) gebildet, wie z.B. die DLUG (`http://www.dlug.de/`) in Düsseldorf. Diese unterhalten oft elektronische Diskussionsforen (Mailinglisten) oder auch Stammtische, bei denen man oft wertvolles Wissen, mit etwas Glück sogar aus "erster Hand", also vom jeweiligen Programmierer der fraglichen Software erwerben kann.

Außerdem veranstalten viele LUGs regelmäßig sogenannte Installparties, wo man das "Abenteuer Linuxinstallation" unter fachkundiger Anleitung hinter sich bringen kann, oder Kurse und Workshops zu einzelnen Themen.

Professioneller Support

Es gibt inzwischen auch auf Linux spezialisierte DV-Serviceunternehmen, die professionellen Linux-Support leisten. Für den Privatanwender werden allerdings die Preise oft eher abschreckend sein. Für die kommerzielle Anwendung erhalten Sie allerdings im allgemeinen sehr kompetente Unterstützung bei der Lösung Ihrer Probleme. In der Linux Consultants HOWTO (`http://www.linuxports.com/howto/consultants/Consultants-HOWTO.html`) finden Sie viele entsprechende Firmen.

4.5 Der Verzeichnisbaum

4.5.1 Einfache Dateien

Einzelne Dokumente werden in Rechnern als sogenannte "Dateien (engl. Files) verwaltet. Dabei handelt es sich um benannte Speicherobjekte, die eine (fast) beliebige Anzahl Bytes umfassen dürfen.

Den Namen des Objektes vergibt der Benutzer. Um nun zu vermeiden, daß es zu Überschneidungen bei beliebten Dateinamen (wie z.B. `test`) kommt, wird das Konzept des Verzeichnisbaumes eingeführt.

4.5.2 Directories

Zusätzlich zu den herkömmlichen Dateien (regular files) gibt es dort Verzeichnisse (directories), die als Behälter (container) für andere Dateien (und damit auch weiteren Verzeichnissen) dienen. Dies erlaubt

- mehrere gleichnamige Dateien zu haben, solange sie sich nicht im selben Verzeichnis befinden.

- Dateien nach ihrem Zweck bzw. nach Zusammengehörigkeit zu sortieren.

- Zugriffsrechte für ganze Dateibereiche festzulegen (indem der Zugriff auf das Directory, das diese Bereiche enthält, beschränkt wird).

4.5.3 Links

Manchmal ist es sinnvoll, dasselbe Speicherobjekt unter verschiedenen Namen ansprechen zu können. Zum Beispiel, wenn eine Software pro Benutzer konfiguriert werden kann, die meisten Benutzer aber eine vom Systemverwalter eingerichtete Standardkonfiguration benutzen.

Man könnte nun diese Standardkonfiguration für jeden Benutzer kopieren, aber das wäre weder ökonomisch (man bräuchte ja Speicherplatz für jede Kopie), noch praktisch, da diese Kopien bei jeder Änderung aufgefrischt werden müssen.

Daher bietet Unix die Möglichkeit, sogenannte Links (Verweise) anzulegen. Es handelt sich dabei um Verzeichniseinträge, die auf andere Dateien verweisen. Nach außen verhalten sich diese (fast) genau wie die Dateien, auf die sie verweisen.

Solche Links werden mit `ln` angelegt - wie Sie Informationen über die Benutzung dieses Kommandos erhalten, haben Sie ja bereits erfahren.

Sie werden in der Manpage feststellen, daß zwischen sogenannten "hard- und "soft-links unterscheiden wird.

Der Unterschied ist eher technischer Natur und Sie werden in den meisten Fällen Softlinks verwenden wollen. Softlinks verweisen über den Dateinamen auf das andere Objekt - das heißt, wenn Sie das ursprüngliche Objekt umbenennen "bricht der Link (er zeigt dann ins "Leere). Hardlinks zeigen dieses Verhalten nicht, haben aber andere Einschränkungen (sie können keine Verzeichnisse verbinden und der Link kann keine Dateisystem-Grenzen überschreiten - dazu später mehr).

4.5.4 Gerätedateien

Das Konzept, daß Speicherobjekte im Verzeichnisbaum abgelegt werden, ist bei Unix sehr weit gefaßt. Unter Unix wird so ziemlich alles als Datei dargestellt - insbesondere auch die angeschlossenen Geräte.

Dazu ist ein spezieller Pfad (`/dev`) im Verzeichnisbaum reserviert, in dem diese speziellen Gerätedateien angelegt werden (sie können natürlich überall angelegt werden, nur ist dies selten sinnvoll).

Diese "Dateien erlauben einen relativ direkten (allerdings vom Betriebssystem überwachten) Zugriff auf die damit verbundenen Geräte. Wenn Sie z.B. eine Soundkarte im Rechner haben, und diese korrekt konfiguriert ist, können Sie ihr z.B. mit `cat /etc/passwd >/dev/audio` ein kurzes Zischen entlocken.

Mit den Gerätedateien müssen Sie relativ selten direkt arbeiten. Sie benötigen sie manchmal zu Testzwecken oder administrativen Aufgaben. Als normaler Nutzer greifen Sie fast nie explizit auf diese Dateien zu (obwohl viele Programme dies implizit für Sie tun werden, um z.B. Audioausgaben zu erzeugen).

4.5.5 Named Pipes und Sockets

Diese Sonderdateien sollen hier nur kurz erwähnt werden - sie werden zum Transport von Daten zwischen Programmen benutzt.

4.5.6 Der Aufbau des Verzeichnisbaumes

Unix kennt im Gegensatz zu DOS kein "Laufwerkskonzept". Der Verzeichnisbaum startet im Wurzelverzeichnis (root directory), über dem alle Zweige angeordnet werden.

Wenn Sie verschiedene Partitionen angelegt haben, werden diese nicht wie bei DOS verschiedenen Laufwerksbuchstaben zugeordnet, sondern sie werden an verschiedenen Stellen im Verzeichnisbaum "eingehängt (gemountet).

Diese Zuordnung ist dynamisch und kann (sofern dadurch keine laufenden Programme beeinträchtigt werden) im laufenden Betrieb geändert werden. Eine Ausnahme hiervon bildet das root-Verzeichnis, das immer von mindestens einem laufenden Programm (init) benutzt wird, und daher nicht umgemountet werden kann.

Im Hauptverzeichnis sollten sich folgende Unterverzeichnisse befinden:

/bin enthält alle Dateien, die beim Hochfahren des Systems benötigt werden (z.B. der `mount`-Befehl, der gebraucht wird, um die anderen Dateisysteme einzuhängen), und auch später von normalen Benutzern noch gebraucht werden könnten.

/boot enthält für den Bootvorgang des Kernels wichtige Dateien wie die Loadmap, der Kernel selbst, ...

/cdrom ist in der SuSe-Distribution als sog. Mountpoint für das CDROM vorgesehen. Wenn Sie mit `mount /cdrom` eine CD einhängen finden Sie die Dateien auf der CDROM anschließend in diesem Verzeichnis. Bei der RedHat finden Sie diesen Mountpoint unter `/mnt/cdrom`.

/dev enthält wie oben erklärt die Gerätedateien.

/etc Konfigurationsdateien für diesen Rechner.

/floppy siehe `/cdrom`, nur für die Floppy.

/home beinhaltet die "Heimatverzeichnisse (home directories) der Benutzer. Jeder Benutzer erhält ein Unterverzeichnis in `/home`, in dem er seine eigenen Dateien ablegen darf.

/lib zum Systemstart wichtige Bibliotheken (Hilfsdateien, die häufig von Programmen gebrauchte Befehlssequenzen enthalten).

/lost+found ist nicht wirklich ein absichtlich angelegtes Verzeichnis, sondern ein internes Hilfsverzeichnis des Dateisystems. Falls es zu Beschädigungen am Dateisystem kommt (z.B. nach einem Stromausfall), legen die Reparaturtools wiedergefundene Dateien dort ab.

/mnt ist vorgesehen, um Mountpoints, insbesondere für wechselbare medien anzulegen. Einige Distributionen verwenden `/mnt` selbst als "Universalmountpoint (SuSE), andere legen darin Unterverzeichnisse für einzelne Geräte (CDROM, Floppy, etc.) an.

/opt ist nonstandard und reserviert für **opt**ionale Pakete. Einige kommerzielle Pakete wie die Applixware installieren sich in diesen Pfad.

/proc ist ein Pseudo-Dateisystem. Es enthält "Dateien, die den aktuellen Systemstatus wiederspiegeln. Sie können oftmals sehr nützliche Informationen aus diesen Dateien erhalten.

/root ist das Heitmatverzeichnis des Superusers. Es liegt nicht auf `/home`, da `/home` u.U. während Wartungsarbeiten etc. nicht zur Verfügung stehen könnte.

/sbin entspricht /bin, allerdings enthält es nur Kommandos, die der Normalbenutzer zumeist nicht benötigt (**s**uperuser **bin**aries).

/tmp Platz für temporäre Dateien. Obwohl normale Nutzer dort Dateien anlegen dürfen, sollten Sie dort nichts "wertvolles lagern, da dieses Verzeichnis vom Administrator oder auch automatisch gelegentlich gelöscht wird.

/usr enthält alle Benutzerprogramme und deren Grunddaten (Hilfeseiten, Tabellen etc.). Der Inhalt dieses Verzeichnisses ist im wesentlichen statisch, d.h. er ändert sich mit der Zeit nur wenig.

/var wird für veränderliche (**var**iable) Daten verwendet, wie den Inhalt der Benutzer-Mailboxen, Newsartikel, ausstehende Printjobs etc..

4.5.7 Die Dateiattribute

Abgesehen vom Namen einer Datei enthält ein Verzeichniseintrag noch einige weitere Attribute:

size Länge der Datei in Bytes. Bitte beachten Sie, daß die Datei auf der Platte immer ganze "Cluster" belegt (so daß insbesondere sehr kleine Dateien mehr Platz belegen, als man zunächst anhand des Längeneintrages glauben würde) und andererseits Unix in der Lage ist, "leere" Dateibereiche (die nie beschrieben wurden und somit per default nur Nullbytes enthalten) effizienter zu speichern. Die Längenangabe stimmt somit nicht unbedingt mit dem verbrauchten Plattenplatz (den Sie mit dem Kommando `du` (**d**isk **u**sage) ermitteln können) überein.

owner Besitzer der Datei. Wenn eine neue Datei angelegt wird, geschieht das mit dem anlegenden Benutzer als Besitzer. Dies erlaubt, dem Besitzer andere Rechte einzuraumen, als den anderen Benutzern.

group Gruppe zu der die Datei gehört. Einer Gruppe gehören i.a. mehrere Benutzer an, was es erlaubt gezielt bestimmten Benutzern (Mitgliedern derselben Gruppe) besondere Rechte einzuräumen.

link count Anzahl von per Hardlink angelegten Referenzen. Nur von internem Interesse.

mtime Modification time, Änderungszeit. Gibt den Zeitpunkt der letzten Änderung der Datei an.

ctime Change time, Statusänderungszeit. Gibt denZeitpunkt der letzten Änderung des Dateistatus (Attribute) an.

atime Access time, Zeit des letzten Zugriffs auf die Datei.

rights Zugriffsrechte. Getrennt für Benutzer (**u**ser), Gruppe (**g**roup) und alle anderen (**o**ther) läßt sich hier einstellen, ob eine Datei gelesen (**r**ead), beschrieben (**w**rite) und ausgeführt (e**x**ecute) werden darf.

Sie können diese Attribute mit `ls` sichtbar machen. Wenn Sie sich für die Dateiattribute interessieren werden sie zumeist `ls -l` verwenden:

```
-rw-r--r--    1 becka    cv          37438 Apr 28 14:21 nutzen.tex
Rechte       LZ Nutzer   Gruppe     Laenge MTime        Dateiname
```

Dies ist die Ausgabe von `ls -l`, angewendet auf die Datei, die diesen Text enthält. Es wird nur die mtime angezeigt, da diese meist die interessante Zeit ist. Der Besitzer der Datei ist der Benutzer `becka`, die Gruppe `cv` Die Länge der Datei ist zur Zeit 23894 Bytes und es gibt keine Hardlinks auf die Datei (der Linkzähler Lz ist 1).

Der Rechte-Eintrag verdient etwas genauere Betrachtung:

Der erste `-` sagt aus, daß es sich um eine geöhnliche Datei handelt. Directories werden durch ein `d` an dieser Stelle angezeigt, symbolische Links durch ein `l` etc.

Danach folgen drei Dreiergruppen, die die Rechte für den Benutzer, die Gruppe und alle anderen (in dieser Reihenfolge) regeln.

Der Benutzer darf die Datei lesen (**r**ead) und beschreiben (**w**rite), die Gruppe und die anderen lediglich lesen. Niemand darf die Datei ausführen (execute), da es sich um eine Textdatei und nicht um ein Programm handelt. Im Gegensatz zu DOS erkennt UNIX ausführbare Dateien nicht an einer "magischen" Endung wie .EXE, .COM oder .BAT, sondern daran, daß das execute-bit gesetzt ist, wie hier:

```
becka@ecce:~/doc/vorl_linux > ls -l /bin/ls
-rwxr-xr-x    1 root     root        52292 Nov  8 20:51 /bin/ls
```

Diese Datei enthält das `ls` Programm. Programme gehören i.a. root und sind auch nur für ihn schreibbar. Alle anderen Benutzer dürfen sie nur ausführen.

Diese Eigenschaft schützt Unix weitgehend vor Viren. Für Linux sind nur zwei Viren bekannt, die jedoch beide lediglich als "Proof of Concept", also als "Beweis, daß es geht", gedacht sind. Bei beiden wird kein bleibender Schaden angerichtet und sie können sich nur einnisten, wenn man als root unvorsichtig fremde Programme startet.

Tut man dies als normaler Nutzer, ist der Schaden von vorneherein begrenzt, da eben nur die eigenen Dateien überhaupt beschrieben werden dürfen und somit dem Virus auch sein Verbreitungsmechanismus (das sich-an-andere-Programme-anhängen) genommen ist.

Mode oktal	Mode rwx	typ. Verwendung
600	`rw-------`	private Dateien
640	`rw-r-----`	für die Gruppe lesbare Datei
660	`rw-rw----`	gemeinsam von der Gruppe bearbeitbare Datei
644	`rw-r--r--`	öffentlich lesbare Datei
700	`rwx------`	privates Directory/Executable
750	`rwxr-x---`	für Gruppe zugängliches Directory/Executable
770	`rwxrwx---`	von der Gruppe bearbeitbares Directory/Executable
755	`rwxr-xr-x`	öffentlich zugängliches Directory/Executable

Tabelle 4.2: Gängige Kombinationen von Dateirechten

Bitte beachten Sie, daß die Bedeutung der Rechte bei Directories etwas anders interpretiert wird:

write bedeutet hier, daß Dateien angelegt und gelöscht werden dürfen.

read erlaubt das Durchsuchen des Verzeichnisses, also das Anzeigen der darin enthaltenen Dateien.

e**x**ecute erlaubt das "Betreten" dieses Verzeichnisses und damit den Zugriff auf darin befindliche Dateien.

4.5.8 Ändern von Dateiattributen

Insbesondere den Benutzer, dir Gruppe und die Rechte von Dateien möchte man zuweilen ändern. Dazu dienen die Kommandos `chown`, `chgrp` und `chmod`. Bitte lesen Sie die zu diesen Kommandos gehörigen Manpages.

Seien Sie vorsichtig beim Gebrauch dieser Befehle. Falsch gesetzte Rechte können sehr schnell das System unbrauchbar machen. Prüfen sie ein solches Kommando lieber einmal zu viel als zu wenig.

Einige kurze Bemerkungen zu chmod:

Die normale Sytax für chmod lautet: `chmod [u|g|o|a][+|-][r|w|x] [Dateiname]`. Dabei gibt man die gewünschte Veränderung an: `[u|g|o|a]` bestimmt, wessen Rechte geändert werden (**u**ser, **g**roup, **o**ther oder für **a**lle gleichzeitig), `[+|-]` ob das Recht verliehen (+) oder entzogen (-) werden soll und `[r|w|x]`, welches Recht verändert werden soll.

Alternativ gibt es eine schnellere Methode bestimmte Rechte zu setzen. Sie hat die Syntax: `chmod [0-7][0-7][0-7] [Dateiname]`

Dabei entspricht jede der drei Zahlen von 0-7 jeweils den Rechten einer Rechteklasse, in der Reihenfolge Benutzer, Gruppe, Andere. Die einzelnen Rechte ergeben sich durch Addition der Werte 4 für read, 2 für write und 1 für execute.

Besonders gängige Kombinationen, die oft verwendet werden (siehe Tabelle 4.2), setzt man meist auf diese Weise, da man nichts vergessen kann und das Ergebnis nicht vom vorherigen Zustand abhängt.

4.6 Editoren

Bei der Benutzung von Linux werden Sie recht oft mit Editoren arbeiten. Sei es um Mail einzugeben, die Oberfläche an Ihre Wünsche anzupassen, Programme und Shellscripte zu schreiben oder administrative Einstellungen zu treffen.

Da traditionell unter Unix fast alle Systemeinstellungen in menschenlesbaren Textdateien erfolgen (und nicht in einer obskuren, eher für die Lesbarkeit durch die Maschine geeigneten Registry), ist ein Texteditor auch das wichtigste Werkzeug bei der Systemadministration, wenn man sich nicht auf die Administrationstools verlassen will oder kann.

Im folgenden werden die drei wichtigsten und beliebtesten Editoren unter Linux kurz besprochen. Lesen Sie bitte alle drei Abschnitte, auch wenn Sie nur mit einem Editor arbeiten wollen. Einige Programme starten automatisch bestimmte Editoren, wenn Sie längere Benutzereingaben erwarten ... es kann in diesem Fall (besonders vom `vi`) ausgesprochen schwierig sein, den Editor wieder zu verlassen, wenn man nicht die entsprechende Tastenkombination kennt.

4.6.1 vi

Vi ist der "Dinosaurier" unter den unixtypischen Editoren, was sein Alter betrifft. Er ist sehr klein und schnell, aber durchaus sehr mächtig.

Allerdings wurde der vi ursprünglich für sehr eingeschränkte Terminals entwickelt, die oft noch nicht einmal einen Cursorblock, geschweige denn Editiertasten wie Insert, Delete etc. hatten.

Daher steuert der vi diese Funktionen mit Escape-Sequenzen an. Auf modernen Linuxsystemen werden aber die vorhandenen Tasten unterstützt, so daß man einigermaßen normal arbeiten kann, wenn man einmal im sog. "Insert-Mode" ist. Dazu bewegt man den Cursor an die Stelle, an der man Veränderungen vornehmen will und drückt [Esc] [i]. Danach verält sich `vi` weitgehend wie ein "normaler" Texteditor.

Sie werden sich ggf. wundern, wie man etwas am Ende einer Zeile einfügen kann, denn der Insert-Mode stellt den Cursor immer vor das letzte Zeichen in der Zeile. `Vi` hat hierfür einen eigenen Mode, den **a**ppend mode, der mit Esc a aktiviert wird.

Häufig brauchen werden Sie wahrscheinlich auch das Löschen von Zeilen mit Esc d d oder von Zeichen (obwohl dies bei modernen `vi`- Varianten zumeist auch im Insert-Modus mit den Tasten Del oder ⟵ geht) mit Esc d Space.

Etwas wundern werden Sie sich auch, wenn Sie versuchen, zwei Zeilen zu verbinden. Die geht nicht, wie von anderen Editoren gewohnt durch Löschen des (unsichtbaren) Zeilenumbruchzeichens am Ende der Zeile, sondern mit Esc J (**J**oin, Achtung: großes J !).

Zum Verlassen von `vi` haben Sie die Alternativen:

- Esc : w q Enter - Speichern der aktuellen Datei und verlassen des `vi`
- Esc : q ! Enter - Verlassen des `vi` ohne zu speichern.

4.6.2 emacs

Emacs ist der "Dinosaurier" unter den Unix-Editoren, was seinen Umfang betrifft. Er ist sehr sehr mächtig, allerdings auch nicht gerade klein und braucht damit so einige Zeit zum Laden.

Auf heutigen Rechnern ist dies nicht mehr so kritisch, aber zu Zeiten in denen 4MB Hauptspeicher üblich waren, wurde emacs auch gerne übersetzt als **e**ight **m**egabytes **a**lways **c**onstantly **s**wapping.

Emacs ist in LISP programmierbar (böse Zugen behaupten, Emacs wäre kein Editor, sondern ein LISP-Interpreter mit Beispielapplikation Editor) und kann damit auf sehr viele Einsatzgebiete angepaßt werden. Er bietet spezielle Modi zur Bearbeitung gängiger Dateitypen, wie C-Quelltexten, LaTeX-Dateien etc.

Dieses Script entstand vorwiegend unter XEmacs (einer Variante des Emacs, das aufgrund seiner Integration mit der grafischen Oberfläche X-Windows noch einige zusätzliche Effekte beherrscht) wobei durch die Integration der LaTeX-Tools ein sehr effektives Arbeiten möglich ist.

Auch hier seien die wichtigsten Tastenkombinationen kurz erwähnt: Im Grundmodus verhält sich emacs wie sie es von einem modernen Editor erwarten würden - sie bewegen sich mit den Cursortasten durch den Text, fügen Zeichen einfach durch Eingeben ein und löschen Zeichen mit Del oder ⟵. Ganze Zeilen können mit Ctrl+k (**k**ill) gelöscht werden. Diese Zeilen landen im sogenannten "kill-ring", aus dem sie mit ctrl+y (**y**ank) wieder eingefügt werden können.

Sie können die aktuelle Datei mit Ctrl+x Ctrl+s abspeichern und den Editor mit Ctrl+x Ctrl+c verlassen. Haben Sie die Datei zuvor nicht gespeichert, erfolgt eine

Sicherheitsabfrage.

4.6.3 joe

Joe ist ein relativ neuer Editor, der klein und leicht angelegt ist, wie der vi, aber eine etwas zeitgemäßere Kommandostruktur hat.

Seine Kommandos sind an WordStar angelehnt, und dürften somit vielen Anwendern aus anderen Editoren bekannt sein.

Auch hier landen Sie nach dem Aufruf direkt im Editiermodus und können Ihren Text ähnlich wie beim emacs mit den üblichen Tasten manipulieren.

Interessant beim joe sind insbesondere die Blockkommandos:

- Den Anfang des markierten Blockes legen Sie mit Ctrl+k Ctrl+b (**b**egin) fest.
- Das Blockende bestimmen Sie mit Ctrl+k Ctrl+k.
- Mit Ctrl+k Ctrl+c (**c**opy) können Sie den markierten Block an die aktuelle Cursorposition kopieren.
- Mit Ctrl+k Ctrl+m (**m**ove) verschieben Sie den Block an die aktuelle Cursorposition.
- Mit Ctrl+k Ctrl+y wird der aktuelle Block gelöscht.

Außerdem werden Sie die Kommandos zum Verlassen des Editors benötigen:

- Zum Verlassen von joe ohne zu speichern benutzen Sie Ctrl+c (**c**ancel) und beantworten sie darauf ggf. erscheinende Sicherheitsabfrage.
- Zum Verlassen mit Speichern der Änderungen verwenden Sie Ctrl+k Ctrl+x (e**x**it).

Joe ist trotz seiner geringen Größe ein sehr mächtiger Editor, der "teach-in"-Makros, Suchfunktionen etc. beherrscht. Sie erhalten eine Kurzhilfe im Editor über Ctrl+k Ctrl+h (**h**elp) und eine ausführliche Hilfe in den oben beschriebenen Hilfe- und Dokumentationssystemen.

4.7 Übungen

1. Loggen Sie sich als normaler Benutzer neu ein und finden Sie heraus, in welchem Verzeichnis Sie sich befinden.

2. Begeben Sie sich in die Wurzel des Dateibaumes. Wieviele verschiedene Arten, dies ohne Umwege zu erreichen kennen Sie?

3. Geben Sie einfach nur `cd` als Kommando ein. Was passiert? In welchem Verzeichnis sind Sie anschließend?

4. Im Verzeichnis `/usr/bin` befinden sich alle normalen Benutzerkommandos. Lassen Sie sich dieses Verzeichnis anzeigen und lesen Sie die Manpages zu einigen Kommandos. Überprüfen Sie auch, ob zu diesen Kommandos auch im Info-System Einträge vorhanden sind.

5. Mit dem Pipe-Operator `|` kann man die Ausgabe eines Kommandos zur Eingabe des nächsten Kommandos machen. Das `/usr/bin`-Verzeichnis ist sehr lang. Um es seitenweise zu betrachten, können Sie z.B. `ls -l | less` benutzen.

6. Wem gehören die Dateien in `/usr/bin`? Gibt es Ausnahmen?

7. Welchen Gruppen gehören diese Dateien an?

8. Welchen Rechtekombinationen tauchen auf? Gibt es nur normale Dateien oder auch Links, Directories etc.?

9. Gibt es zusätzliche Rechte, die noch nicht behandelt wurden ?

10. Wie viele Kommandos enthält `/usr/bin` ? Hinweis: Verwenden Sie `less` oder `wc` (Beschreibung siehe Manpages !).

11. Erstellen Sie nun mit einem Editor Ihrer Wahl eine Datei names “test.txt”. In welchem Verzeichnis sollten Sie diese datei erstellen?

12. Tragen Sie einie Zeilen in diese Datei ein, speichern Sie sie ab und beenden Sie den Editor.

13. Benutzen Sie `ls -l`, um sich die Dateirechte anzeigen zu lassen und anschliessend den Editor, `less` und `cat`, um sich den Dateiinhalt noch einmal anzusehen.

14. Wenden Sie das Kommando `chmod 444 test.txt` an. Was passiert? Wiederholen Sie die Schritte der vorherigen Aufgabe. Wie können Sie diese Änderung rückgängig machen?

15. Versuchen Sie, sich die Datei `/etc/shadow` mit less anzeigen zu lassen. Was geschieht? Warum?

16. Benutzen sie `apropos` und `man`, um mehr über den Zweck und Aufbau der Datei `/etc/shadow` zu erfahren. Erklärt das das Verhalten der Datei bei der vorherigen Frage?

17. Die Manpage shadow in Sektion 5 verweist auf eine weitere Datei, die Informationen über Benutzeraccounts enthält. Wiederholen Sie die beiden vorherigen Fragen mit dieser Datei.

4.8 Zusammenfassung

Sie sollten nun in der Lage sein, sich in Linux einzuloggen, Dateien zu manipulieren (`rm, mv, cp, ln`) und zu editieren und gezielt nach gewünschten Befehlen und deren Dokumentation zu suchen.

Üben Sie diese Fähigkeiten mit Hilfe von Kapitel 4.7 und durch eingenständiges "Erforschen" des installierten Systems. Führen Sie dies aber **unbedingt** unter einer "normalen" Benutzerkennung durch, **niemals als root**.

Das nächste Kapitel beschäftigt sich mit der Konfiguration von Systemparametern und macht von diesen Methoden Gebrauch. Da dort aber oft als **root** gearbeitet werden muß, ist es wichtig, daß Sie im Umgang mit den Kommandos und Programmen sicher sind, da eine Fehlbedienung weitreichende Konsequenzen haben kann.

Kapitel 5

Konfiguration und Administration

Dieses Kapitel beschäftigt sich mit der Konfiguration von Linux (und anderen Unixen, da die Grundkonfiguration zumeist sehr ähnlich realisiert ist).

Wir werden dabei bewußt nicht mit den distributionseigenen Hilfsprogrammen arbeiten, da diese sich von Distribution zu Distribution unterscheiden.

5.1 Benutzerkonfiguration

Das Benutzerkonzept von Unix ordnet jedem Benutzer eine numerische Benutzer-Identifikation (User-ID, UID) zu. Diese ist maßgeblich für alle Rechtebeschränkungen dieses Benutzers. D.h. wenn Sie einen Nutzer löschen und später unter der gleichen ID einen neuen Benutzer anlegen, "erbt" dieser Zugang zu allen Dateien, die dem alten Benutzer gehört haben. In gewissen Situationen mag dies sinnvoll sein, doch zumeist sollte man Sorge tragen, daß solche alten Dateien zuvor gelöscht werden bzw. man besser eine neue UID vergibt. Es ist möglich, wenn auch nur in Ausnahmefällen sinnvoll, mehrere Benutzernamen mit der gleichen UID anzulegen.

Die UID 0 genießt eine Sonderstellung: Sie gehört dem Benutzer "root" (der Benutzername ist nicht wirklich wichtig, der Administratoraccount wird aber traditionell so benannt) und erlaubt das Übergehen fast aller Sicherheitsprüfungen. Daher sollte man so wenig wie möglich unter dieser UID arbeiten (benutzen Sie ggf. das Kommando `su`, um für kurze Zeit Administrator zu werden).

Die Liste der Benutzer finden Sie in der Datei `/etc/passwd`.

Zu jedem Benutzer werden dort bzw. in der optionalen `/etc/shadow` Datei folgende Informationen abgelegt:

- Login-Name. Der Name des Benutzers, wie er bei der Anmeldung am System eingegeben werden muß.

- Verschlüsseltes Paßwort. Das Paßwort des Benutzers, das bei der Anmeldung am System eingegeben werden muß, wird mit einer sog. Einwegfunktion verschlüsselt. Es ist also nicht möglich, aus dem verschlüsselten Paßwort wieder das Originalpaßwort zu berechnen. Zur Kontrolle des Logins wird das eingegebene Paßwort erneut verschlüsselt, und das Ergebnis mit dem gespeicherten verglichen. Allerdings ist es möglich durch Wörterbuch- basierte Attacken oder sog. "brute-force" (brutale Gewalt)-Attacken, bei denen alle Möglichekiten durchprobiert werden, das Originalpaßwort ggf. zu finden. Verwenden Sie daher keine allzu einfachen oder zu kurzen Paßwörter. Der Rechenaufwand für eine brute-force-Attacke steigt exponentiell mit der Länge des Paßwortes. 8 Zeichen sollten heute eine ausreichende Sicherheit bieten. Dieses Feld sollte auch möglichst in der gegen unbefugtes lesen geschützten `/etc/shadow` abgelegt sein.

- Numerische Benutzer ID

- Nummer der Default-Gruppe (siehe unten)

- Kommentarfeld - zumeist genutzt, um den vollen Benutzernamen zu speichern.

- Heimatverzeichnis (home directory). Nach dem Einloggen wird man automatisch in dieses Verzeichnis versetzt, und dort werden benutzerspezifische Anpassungen von Programmen gespeichert.

- Default-Shell. Das Programm, das nach dem Einloggen des Benutzers automatisch gestartet wird.

- Passwortwechselinformationen. Datumsinformationen, ob und wann das Paßwort gewechselt werden muß und ob und wann der Account ggf. abläuft.

Abgesehen von der Benutzerliste gibt es auch eine Gruppenliste in `/etc/group`. Auch hier gibt es jeweils einen lesbaren Gruppennamen, wohingegen intern wieder die Gruppen-Nummer (GID) verwendet wird.

Jeder Benutzer gehört mindestens einer Gruppe, seiner "default group" an. Diese ist als Feld Nummer 4 in `/etc/passwd` eingetragen.

Darüberhinaus kann ein Benutzer zu weiteren Gruppen (auxiliary groups) gehören. Diese Zugehörigkeit wird in `/etc/group` eingetragen.

Mit den Befehlen `newgrp` und `sg` kann die aktuelle Gruppe gewechselt werden.

5.1.1 Benutzerverwaltung

Mit Hilfsprogrammen

In gängigen Linuxsystemen stehen die Tools `useradd`, `userdel` und `usermod` zur Verfügung, die das Anlegen, Löschen und Ändern eines Benutzers vereinfachen. Oft ist zusätzlich eine menügeführte Verwaltung über das Konfigurationstool der Distribution möglich.

Die normale Vorgehensweise, um einen Benutzer mit useradd anzulegen ist:

```
bash# useradd -m [username]
bash# passwd [username]
```

Mit dieser "Einfachvariante" werden alle Werte wie Standardgruppe, Passwortwechseldaten etc. automatisch entsprechend der Voreinstellungen des Systems vergeben. "-m" bewirkt die Erzeugung eines Heimatverzeichnisses mit vorgegebenen Dateien (**m**ake home directory).

Eine wichtige Option ist "-s shell" um die Startshell des Benutzers zu setzen. Unter Linux ist dies üblicherweise die Bash (**B**ourne **A**gain **SH**ell), einige User bevorzugen aber z.B. dir Tcsh, die man dann z.B. mit "-s /bin/tcsh" angeben kann.

[Eine kurze Zwischenbemerkung zu Shells: Die Shell ist das Programm, das beim Einloggen des Benutzers automatisch gestartet wird. Normalerweise handelt es sich dabei um einen Befehlsinterpreter (entsprechend dem DOS-Programm COMMAND.COM), der Benutzerkommandos entgegennimmt und ausführt. Theoretisch ist es möglich völlig andersartige Shells einzustellen, z.B. direkt in ein Menüsystem zu verzweigen.]

Wenn Sie ein Netzwerk aus mehreren Rechnern aufbauen wollen, ist es u.U. sinnvoll, die numerischen User-IDs auf den einzelnen Rechnern synchron zu halten, da diese bei der Benutzung von Dateien über NFS gleich sein sollten. Dazu hat useradd die Option "-u uid" vorgesehen.

Der Account wird bei der Einrichtung mit einem "unmöglichen" Passwort versehen, und dadurch gesperrt.

Daher muß der Account erst mit einem gültigen Paßwort versehen werden, bevor er benutzt werden kann. Der Superuser darf Paßwörter für fremde Accounts ändern, und er muß dazu auch das alte Paßwort nicht kennen. Auf diese Weise ist es möglich, einem Benutzer zu helfen, der sein Paßwort vergessen hat. Es ist dagegen **nicht** möglich, das vergessene Paßwort zu rekonstruieren.

Bei allen Änderungen von Paßwörtern oder anderen kritischen Systemparametern, empfiehlt es sich dringend, zunächst einen Test auf einer freien virtuellen Konsole durchzuführen, bevor man sich ausloggt. So kann man im Falle eines Fehlers zumindest noch mit Hilfe der noch laufenden Sitzung eventuelle Probleme noch korrigieren.

Per Hand

Stehen solche Tools nicht zur Verfügung, oder sind z.B. die Passwortdateien beschädigt, ist es zuweilen nötig, manuelle Eingriffe vorzunehmen, um sich wieder in das System einloggen zu können.

Dazu öffnet man mit einem Editor die Datei `/etc/passwd` und erzeugt einen gültigen Eintrag. Sofern noch andere Einträge existieren, kopiert man am einfachsten einen der existierenden Einträge.

Bei "Rettungsaktionen" legt man i.a. einfach einen neuen passwortlosen Rootaccount an. Dazu läßt man einfach das zweite Feld frei.

Bei Systemen mit shadow-passwords muß auch noc `/etc/shadow` entsprechend angepaßt werden. Auch hier ist das zweite Feld freizulassen, um einen passwortlosen Account anzulegen.

Natürlich sollte man diesem Account dann schnellstmöglich ein Passwort verpassen, sobald man wieder normalen Zugriff auf das System hat.

Ein zumeist per Hand ausgeführte Tätigkeit ist die Aufnahme von Benutzern in Gruppen. Dazu trägt man den jeweiligen Benutzer einfach mit Komma getrennt in die Liste im letzten Feld der `/etc/group` ein.

5.1.2 Die Benutzerrechte

Alle Benutzerrechte leiten sich wie bereits erwähnt von der numerischen User ID (UID) ab. Dabei haben die einzelnen Zahlen selbst keine Bedeutung — mit einer Ausnahme: Die ID 0 ist für root, den Administrator reserviert.

Prozesse die mit dieser User-ID laufen (Prozesse [also Programme] erhalten die Rechte des Benutzers, der sie gestartet hat), haben stark erweiterte Rechte. Sie können sich über fast jede Zugriffsbeschränkung hinwegsetzen. Sie können alle Dateien lesen und schreiben (egal, ob ihnen dafür Berechtigung erteilt wurde oder nicht) und Systemparameter verändern (Netzwerkinterfaces starten und stoppen, Firewall-Regeln ändern, belibige Dateisysteme einhängen).

Nun ist es für einige Tätigkeiten erforderlich, daß man für einen ganz bestimmten Zweck spezielle Zusatzrechte erhält, die man normalerweise nicht hat. Ein Beispiel hierfür ist das Wechseln des eigenen Paßwortes.

Dazu muß auf `/etc/passwd` geschrieben werden, was nur root darf.

Es gibt nun zwei prinzipielle Möglichkeiten, wie dieses Problem unter Unix gelöst werden kann.

Die eine ist es, die Änderung z.B. über eine (ggf. lokale) Netzwerkverbindung von einem Programm anzufordern, das die erforderlichen Rechte besitzt. Dazu erfahren wir später noch mehr.

Die andere, die wir zunächst betrachten wollen, besteht in sog. SetUID (SUID) bzw. SetGID (SGID) Programmen.

Das Dateisystem besitzt noch ein viertes "Rechteoktett", mit dessen Hilfe Programme geschaffen werden können, die beim Aufruf die UID bzw. GID annehmen, der sie "gehören".

Die entsprechenden `chmod`-Befehle dafür lauten `chmos u+s Datei` bzw. `chmod g+s Datei` bzw haben die oktalen Masken 04000 (SUID) und 02000 (SGID).

Betrachten sie z.B.

```
ls -l /usr/bin/passwd
-rwsr-xr-x   1 root     shadow      27920 Mar 11 11:50 /usr/bin/passwd
```

Diese Datei hat das SUID Attribut gesetzt (erkennbar am "s" am Platz des "x" in der Benutzersektion). Führt man sie aus, so erlangt der Prozess root-Privilegien und ist daher in der Lage, `/etc/passwd` zu ändern.

Solche Programme bedürfen der besonderen Umsicht des Programmierers, damit sie nicht mißbraucht werden können. Die Programme sollten sehr sehr strikt prüfen, ob die Aktion, die der Benutzer anfordert keine Übertretung der Sicherheitsrichtlinien bedeutet.

Warnung: Setzen Sie **niemals** das SUID-Bit an einem Programm, wenn Sie nicht zu mehr als 100% überzeugt sind, daß das Programm dafür ausgelegt ist. Insbesondere nicht, wenn der Besitzer root ist.

Ähnliches gilt für SGID. Allerdings ist hier der mögliche Schaden ein wenig geringer, da es keine "besondere" GID mit Sonderrechten auf Betriebssystemebene gibt.

Allerdings ermöglicht es ein unsicheres SGID-Programm unter Umständen kritische Systemdateien zu schreiben, was wiederum die Gewinnung von root-Rechten zur Folge haben kann.

5.2 Dateisystem

Linux verwaltet alle verfügbaren physischen und logischen Geräte in einem einzigen Verzeichnisbaum.

In diesen können an beliebigen Stellen vom Administrator Inhalte von Festplatten, Wechselmedien (Diskette, CD-ROM), Netzwerkservern etc. eingebunden werden.

5.2.1 Einhängen (mounten) von Geräten in den Dateibaum

Der Administrator kann mit den Befehlen `mount` und `umount` einzelne Geräte in den Dateibaum einfügen oder daraus entfernen.

Die Sytax von `mount` lautet:

`mount [-t type] [-o optionen] Ger"atedatei Verzeichnis`

Dabei gibt Gerätedatei einen Eintrag in `/dev/` an, der mit dem gewünschten physikalischen oder logischen Gerät verbunden ist. z.B. Ist `/dev/hda1` die erste Partition (1) auf der IDE-Festplatte (hd, harddisk), die auf dem primären IDE-Controller (hda/hdb) als Master (hda) konfiguriert ist, oder `/dev/sdb5` die erste erweiterte Partition (erweiterte Partitionen werden ab 5 numeriert) auf der zweiten (b) SCSI-Disk (sd).

Verzeichnis zeigt an, wo das auf der Partition enthaltene Dateisystem im Verzeichnisbaum eingefügt werden soll. (Heinweis: Falls dieses Verzeichnis nicht leer war, so istWas immer sich unterhalb des hier angegebenen Verzeichnisses befand, bis zum umount nicht mehr sichtbar.)

Manche Geräte, z.B. Disketten und CDROMs besitzen keine Aufteilung in Partitionen. In diesem Fall mounted man das gesamte Geräte (also z.B. `/dev/hdc, /dev/scd0, /dev/fd0`.

Nun ist aber die Struktur der auf den verschiedenen Medien vorhandenen Daten oft sehr unterschiedlich. Der innere Aufbau von DOS-Disketten ist ein ganz anderer wie der eines CDROMs und wieder anders als der einer Linuxpartition.

`mount` versucht zwar, sich automatisch um diese problematik zu kümmern, ist dabei aber nicht immer erfolgreich. Daher kann das "Format" des Gerätes explizit mit dem "-t" (filesystem **t**ype, Dateisystem**t**yp) angegeben werden und ggf. das Verhalten mit den in "-o" gegebenen **O**ptionen noch weiter modifiziert werden.

5.2.2 Wichtige Dateisystemtypen

Es gibt eine Vielzahl von möglichen Dateisystemen, von denen wir nur die wichtigsten kurz beschreiben wollen.

ext2

Ist heute das Standard-Dateisystem für Linux. Es handelt sich um das sogenannte "second extended Filesystem", einer Weiterentwicklung des "extended Filesystem", das seinerseits auf dem Minix-Dateisystem basiert.

Es bietet alle unixtypischen Eigenschaften eines Dateisystems (Rechte, Unterstützung für "special files", etc).

minix

Der Vorgänger. Es besteht — außer um Daten von einem Fremdsystem zu übernehmen, das nur dieses Format beherrscht — heute wenig Veranlassung dieses System zu verwenden. Es hat stärkere Restriktionen bezüglich der möglichen Dateinamen und ist langsamer als ext2.

reiserfs

Ist ein etwas modernerer Ansatz als ext2. Dieses Dateisystem ist noch recht neu, aber angeblich bereits recht stabil. Es ist etwas schneller als ext2 und spart dabei auch noch etwas Platz. Allerdings eignet es sich nur begrenzt für das Dateisystem, das `/boot/` enthält, da LILO nicht mit der Platzsparmaßnahme harmoniert. Ggf. muß das `/boot/`-Dateisystem mit der Option "notail" gemountet werden.

msdos,umsdos,vfat

Diese Dateisysteme erlauben den Zugriff auf MS-DOS/Windows9x-Laufwerke. Die Dateisysteme haben allerdings keine volle Unix-Funktionalität, da ihnen z.B. das Benutzersystem völlig fremd ist. Beim umsdos System werden diese Zusatzattribute "emuliert", indem in jedem DOS-Verzeichnis eine Datei mit diesen Informationen angelegt wird. Bei msdos und vfat gehören alle Dateien einem bestimmten (einstellbaren, `man mount`) Benutzer und einer bestimmten Gruppe und haben alle dieselben Rechte.

ntfs

Das Analogon zum Zugriff auf Windows-NT-Dateisysteme. Zur Zeit ist der Zugriff standardmäßig read-only, man kann also keine Daten auf solche Laufwerke schreiben. Dies ist zwar grundsätzlich einschaltbar, es birgt jedoch dann die Gefahr, die NT-Partition zu beschädigen.

iso9660

Das Dateisystem auf CD-ROMS. Erweiterungen wie Unix-Rock-Ridge oder Microsoft Joliet werden automatisch erkannt und aktiviert.

Entsprechend der Präsenz solcher Erweiterungen ergibt sich mehr oder minder volle Unix-Funktionalität.

coda, ncpfs, nfs, smbfs

Zum beiziehen von Dateien von Netzwerkservern. NFS ist das Standardprotokoll unter Unix, CODA ein ausgeklügeltes verteiltes Dateisystem, ncpfs wird zur Anbindung von Novell-Servern benutzt und smbfs für Windows-Server.

proc

Ist kein Dateisystem im eigentlichen Sinn, sondern eher eine Schnittstelle zum Betriebssystem. Es wird gewöhnlich nach `/proc/` gemountet und stellt dort einige Dateien mit administrativen Daten zur Verfügung. Auch können dort Kernelparameter eingestellt werden. Es benötigt kein "device", da es seinen Inhalt selbst enthält.

5.2.3 /etc/fstab

In dieser Datei können gängige `mount`-Zuordnungen gespeichert werden, so daß sie leichter aufzurufen sind (man braucht nur noch `mount device` oder `mount Verzeichnis` eingeben) bzw. damit die Zuordnungen vom System automatisch beim Hochfahren hergestellt werden können.

Das Format sind einzelne Zeilen mit Einträgen wie in folgendem Beispiel:

```
# Geraet        Verzeichnis                Typ           Optionen dump fsck_pass
/dev/sda2       swap                       swap          defaults   0    0

/dev/sda1       /                          ext2          defaults   1    1
/dev/sda5       /var                       ext2          defaults   1    2
/dev/sda6       /home                      ext2          defaults   1    2

/dev/scd1       /mnt/cdrom                 iso9660       ro,noauto,user,exec 0   0
/dev/scd0       /mnt/writer                iso9660       ro,noauto,user,exec 0 0
/dev/fd0        /mnt/floppy                auto          noauto,user 0   0

none            /proc                      proc          defaults   0    0

sartre2:/home2  /sartre2/home2             nfs           defaults   0    0
```

Die beiden nicht offensichtlichen Einträge dump und fsck_pass haben folgende Bedeutung:

Das Unixeigene Backup-Utility "dump" sichert per default alle Partitionen, die hier einen Eintrag von dump=1 haben.

Beim Systemstart werden alle Dateisysteme überprüft, ob das Dateisystem korrekt heruntergefahren wurde (und damit fehlerfrei sein sollte) und falls nicht ausführlich getestet. Der Eintrag fsck_pass gibt an, in welcher Reihenfolge dies geschehen soll, 0 bedeutet, daß kein Test stattfinden soll. Üblicherweise trägt man hier 1 für das Root-Dateisystem (das bei / gemountet wird) ein, und 2 für alle anderen.

In der Spalte Optionen gibt es drei zusätzliche Einträge, die auf der Kommandozeile keinen Sinn machen:

noauto Das Dateisystem wird nicht automatisch beim Systemstart eingehängt. Diese Option verwendet man gewöhnlich für wechselbare Medien.

user Das Dateisystem darf von nicht-Root Benutzern eigehängt werden. Dieser Eintrag bewirkt auch automatisch einige Restriktionen für den Mount, sofern diese nicht explizit aufgehoben werden, wie z.B. die Option nosuid. Auch diese option wird oft für Wechselmedien verwendet.

defaults ist ein Platzhalter für “keine Optionen”. Er wird benötigt, damit die Feldzählung nicht durcheinanderkommt.

Außerdem sind in obigem Listing noch zwei “besondere” Einträge zu finden. Der allererste ist für eine sogenannte Swap-Partition. Diese wird benutzt, wenn das System nicht mehr genügend Hauptspeicher hat, um alle laufenden Programme im RAM zu halten. Es lagert dann momentan ungenutzte Beriche auf diese Swappartitionen aus. Und der letzte Eintrag ist ein NFS-Mount. Das Dateisystem befindet sich nicht auf einem lokalen Gerät, sondern auf dem Server sartre2 in dessen Verzeichnis /home2.

5.3 Netzwerk

Auch wenn Sie an kein lokales Netzwerk angeschlossen sind, werden Sie unter Linux zumeist trotzdem zumindest rudimentären Netzwerksupport aktivieren, da viele Dienste darauf ausgelegt sind, über das Netzwerk zu funktionieren und sein es nur auf einer Verbindung “innerhalb” einer Maschine.

Diese Konstruktion erlaubt es, Dienste von entfernten Rechnern aus genauso einfach zu nutzen wie von der lokalen Maschine.

5.3.1 Grundlagen TCP/IP

Das Standardprotokoll auf allen Unix-Maschinen ist TCP/IP. Wir werden eine kurzen Blick auf dessen wichtigste Eigenschaften werfen, soweit sie für die Konfiguration von Bedeutung sind.

IP-Adressen

Unter TCP/IP erhält jeder am Netz teilnehmende Rechner eine sog. IP-Adresse. Das ist in IPv4 (dem momentag gängigen Standard, IPv6 ist zwar verfügbar, aber noch nicht verbreitet) ein 4-Tupel von Bytes in durch Punkte getrennter dezimaler Notation.

So ist z.B. die IP-Adresse von "zeus.rz.uni-duesseldorf.de" 134.99.128.99 .

Mit diesem Adressraum lassen sich ca.4 Milliarden Adressen abdecken. Diese Zahl wird in Zukunft vermutlich nicht mehr ausreichen, insbesondere, da der Trend dahin zu gehen scheint, für so ziemlich jedes kommerziell verfügbare Produkt mindestens 3 Domainnamen und damit evtl. auch IPs zu reservieren.

Die IP-Adresse muß weltweit eindeutig sein, und wird daher auf Antrag und gegen Gebühr von einer zentralen Institution vergeben. In Deutschland ist hierfür die DeNIC eG zuständig `http://www.denic.de/` .

Für private Netze verwendet man zumeist einen der drei in RFC 1597 (RFC heißt Request for Comment, die bescheidene Bezeichnung für die Internetstandards) beschriebenen freien Netzbereiche:

```
10.x.x.x          Class-A Net
172.[16-31].x.x   Class-B Net
192.168.x.x       Class-C Net
```

Der Unterscheid zwischen den einzelnen Adressen liegt in der Anzahl adressierbarer Rechner. In aller Regel sollte eines oder einige wenige der C-Klasse Netze ausreichen.

Ein weiterer reservierter Adressraum ist `127.x.x.x`, auch `localnet` genannt. Der Rechner `127.0.0.1` wird `localhost` genannt und verweist stets auf die eigene Maschine. Diese Adresse erlaubt es, Verbindungen innerhalb des eigenen Rechners aufzubauen.

Die Zuordnung von Namen zu Rechnern erfolgt über einen externen Mechanismus (DNS, Domain Name Service), der für das Verständnis von TCP/IP nicht weiter wichtig ist.

Jede gültige IP-Adresse ist innerhalb eines Netzes eindeutig einem Rechner zuzuordnen (Achtung - die Umkehrung gilt nicht, ein Rechner kann mehrere IP-Adressen haben).

Damit ist es möglich, ein IP-Paket (das unter anderem auch die Zieladresse im Kopf beinhaltet) eindeutig zu seinem Ziel zu fhren.

Ports

Das Ziel wird noch mit Hilfe eines "Ports" genauer spezifiziert. Der Port — eine 16-bit Zahl, also ein Wert von 0-65535 — dient dazu, einzelne TCP/IP-Verbindungen eindeutig unterscheiden zu können. Zwei verschiedene TCP/IP-Verbindungen unterscheiden sich immer in mindestens einem der Parameter

- Quell-Adresse (Source-IP)
- Quell-Port (Source-port)
- Ziel-Adresse (Destination-IP)
- Ziel-Port (Destination-port)

Bestimmte Ports (die sogenanneten reservierten Ports mit Nummern kleiner als 1024) sind privilegiert, und sind unter Unix nur für Root benutzbar.

Auf diesen Ports warten normalerweise von Daemonen angelegte "listening sockets" darauf, daß Verbindungen angefordert werden.

Der jeweils verwendete Port korrespondiert mit dem gewünschten Dienst. Diese Zuordnung finden Sie in `/etc/services`. So ist z.B. 25 der Port für die Übertragung von EMail per SMTP (simple mail transfer protocol).

Sie können, wenn auf Ihrer Maschine der SMTP-Dienst aktiviert ist, z.B. einmal "per Hand" mit ihm kommunizieren (eigene Eingaben wurden mit einem * markiert):

```
* bash-2.03> telnet localhost smtp
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 mein.rechner.de -- Server ESMTP (Sun Internet Mail Server sims.4.0.1999.06.13.00.20
* HELO test
250 mein.rechner.de OK, mein.rechner.de [127.0.0.1].
* VRFY becka
250 2.5.0 Local user <becka@mein.rechner.de>
* QUIT
221 2.3.0 Bye received. Goodbye.
```

Wie sie sehen, wird ein einigermassen "menschenlesbares" Protokoll verwendet. Eine sehr angenehme Eigenschaft der meisten TCP/IP-basierten Protokolle, insbesondere in der Unix-Welt, da man damit in der Lage ist, Problemen ohne spezielle Tools auf den Grund gehen zu können.

Netzmasken

Bei der Konfiguration des Netzwerkes werden Sie öfter dem Begriff Netzmaske begegnen. Es handelt sich dabei ebenfalls um ein 4-tupel, wie eine IP-Adresse, das typischerweise von der Form `255.255.255.0` oder ähnlich ist.

Netzmasken werden verwendet, um zu entscheiden, ob eine Zielmaschine direkt mit Hilfe eines lokalen Interfaces (einer Netzwerkkarte oder ähnlichem) erreicht werden kann, oder ob eine Zwischenstation, ein sogenanntes Gateway benutzt werden muß.

Die Netzmaske gibt diejenigen Bits an, die bei Quell- und Zieladresse bereinstimmen müssen, damit es einen direkten Pfad gibt. Konkret wird der Ausdruck
`(Quelladresse & Netzmaske) == (Zieladresse & Netzmaske)`
ausgewertet. Liefert er true, so wird der angegebene Pfad gewählt.

Ein Beispiel hierzu folgt gleich im nächsten Abschnitt.

5.3.2 Interfaces

Ein Interface ist die abstrakte Darstellung einer Netzwerkkarte im Rechner.

Unter Linux werden Interfaces mit Hilfe des Befehls ifconfig eingerichtet.

Lassen Sie sich als root einmal die aktuelle Konfiguration ausgeben, indem Sie ifconfig ohne Parameter aufrufen. Sie sollten zumindest den Eintrag für das "local loopback" interface bekommen:

```
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:3924  Metric:1
          RX packets:59857 errors:0 dropped:0 overruns:0 frame:0
          TX packets:59857 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

Der Name des Interfaces lautet also "lo", die Verbindung wird mit Hilfe des "Hardware"-formates "Local Loopback" aufgebaut, die lokale Internet-Adresse ist 127.0.0.1 und die Netzmaske ist 255.0.0.0.

Das heißt, mit diesem Interface sind theoretisch alle Rechner mit einer IP der Form 127.x.x.x erreichbar. Typischerweise reagiert der eigene Rechner auf alle diese Adressen.

Weiter wird festgestellt, daß das Interface "up" (also aktiviert) ist und es eine MTU (maximum transmission unit, maximale Sendepaketlänge) von 3924 Bytes hat.

Auf dem Interface wurden bisher 59857 Pakete versendet und ebensoviele empfangen (klar, da es ja quasi mit sich selbst verbunden ist), wobei es zu keinen Fehlern kam.

Als weiteres Beispiel wird noch kurz die Ausgabe bei einer "echten" Netzwerkkarte erklärt:

```
eth0      Link encap:Ethernet  HWaddr 00:50:BA:C1:17:3C
          inet addr:134.99.112.39  Bcast:134.99.112.255  Mask:255.255.255.0
```

```
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:1177893 errors:0 dropped:0 overruns:0 frame:0
TX packets:118367 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
Interrupt:10 Base address:0xb800
```

Linux benennt Ethernetkarten als "eth[x]", wobei x in der Reihenfolge des auffindens vergeben wird.

Ethernetkarten besitzen eine interne "Hardwareadresse", die ebenfalls weltweit eindeutig ist. Diese Adresse ist für TCP/IP nicht von Bedeutung, sie wird über ein automatisches Verfahren (ARP, address resolution protocol) der zugehörigen IP-Adresse zugeordnet.

Dieses Interface hat die IP-Adresse `134.99.112.39` zugeordnet bekommen und besitzt eine Netzmaske von `255.255.255.0`, kann also direkt alle Adressen der Form `134.99.112.x` erreichen.

Ethernetkarten beherrschen sogenannte "Broadcast-Packets", die an alle auf dem lokalen Netzstrang angeschlossenen Rechner geschickt werden. Diese Pakete werden insbesondere verwendet, um Dienste "anzupreisen", oder wenn das Ziel noch ermittelt werden muss (z.B. beim ARP Protokoll).

Zu diesem Zweck wird auch auf IP-Ebene eine sogenannte "Broadcast-Adresse" definiert. Pakete die an diese Adresse geschickt werden, erreichen alle am Subnetz angeschlossenen Rechner.

Man beachte, daß die MTU hier kleiner ist, 1500 Byte. Durch unterschiedlich große MTUs innerhalb eines Netzes kann es zu sog. IP-Fragmentation kommen, d.h. ein Host ist gezwungen, ein Ip-Paket auf mehrere kleinere zu verteilen.

Die weiteren Felder erklären sich wie beim loopback-interface.

Die letzte Zeile gibt die Hardwareconfiguration der verwendeten Netzkarte aus. Sie dient nur der Information.

Das Interface bestimmt aber nicht alleine, wohin Daten geschickt werden. Zuständig für den Pfad, den die Daten nehmen ist das "Routing".

5.3.3 Routing

Hier werden spezielle Einstellungen getroffen, wie Pakete an einzelne Ziele gebracht werden sollen.

Geben Sie einmal (als root) "route -n" ein.

```
bash-2.03> /sbin/route -n
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
134.99.112.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
127.0.0.0	0.0.0.0	255.0.0.0	U	0	0	0	lo
0.0.0.0	134.99.112.1	0.0.0.0	UG	0	0	0	eth0

Sie sehen hier drei Einträge. Der mittlere Eintrag repräsentiert wieder das schon bekannte Loopback-Interface.

Der erste Eintrag bestimmt das Routing innerhalb des auf dem lokalen Ethernet (Netzwerkkarte am Interface eth0) verfgbaren IP-Bereiches. Alle Pakete mit Zielen der Form 134.99.112.x werden direkt über das Gerät eth0 an ihr Ziel geschickt.

Alle anderen Pakete (eine Netzmaske von `0.0.0.0` bedeutet "alles", da das Ergebnis der `&` Operation immer 0 ist, und damit der Vergleich immer gelingt) werden an das Gateway, den Rechner mit der IP 134.99.112.1 geschickt.

Dieses Gateway muß natürlich auf irgendeinem anderen Weg (in diesem Fall über den ersten Eintrag der Routingtabelle) erreichbar sein.

Per Konvention findet sich das Gateway typischerweise immer an einem Ende des Adressbereiches, also z.B. auf 1 oder 254. Die Werte 0 und 255 sind für sogenannte Broadcasts, also "Sendungen an alle" reserviert.

5.3.4 Dienste

Wie bereits erwähnt, werden unter Unix viele verschiedenen Netzwerk**dienste** angeboten. Eine (keineswegs vollständige) Liste möglicher Dienste finden Sie in `/etc/services`.

Unter Unix werden solche Dienste von Hintergrundprozessen, sogenannten "Daemonen" verwaltet. Wird ein Dienst angefordert, werden diese Programme aktiv und führen die Kommunikation mit dem anfordernden System und die eigentliche Dienstleistung durch.

Es gibt zwei grundsätzliche Möglichkeiten, einen solchen Dienst zu starten:

- über den "Über-Daemon" inetd oder
- durch direktes Starten des zugehörigen Daemons

Um den Unterschied und damit die unterschiedlichen Einsatzgebiete beider Verfahren zu verstehen, zunächst ein paar kurze Worte zum inetd.

inetd

Inetd ist ein Netzwerkdaemon, der in der Lage ist sehr viele Netzwerkprotokolle zu verwalten. Die zugehörige Konfiguration findet sich in der Datei `/etc/inetd`conf..

Diese enthält Zeilen der Form

```
<service_name> <sock_type> <proto> <flags> <user> <server_path> <args>
```

also z.B.

```
daytime    stream  tcp     nowait  root    internal
```

Diese Zeile beschreibt einen einfachen Dienst, der lediglich die aktuelle Uhrzeit des Servers ausgibt. Der Dienst läuft auf dem Port "daytime", laut /etc/services port 13 (eine genaue Beschreibung dieses Dienstes finden Sie in RFC 867), und ist vom Typ "stream", Protokoll "tcp".

Unter flags ist "nowait" angegeben, das heißt, daß der inetd nachdem er eine Verbindung angenommen hat nicht darauf wartet, daß das behandelnde Programm terminiert, sondern sofort wieder ansprechbar ist und ggf. weitere Kopien des behandelnden Programmes startet um weitere Verbindungen zu behandeln.

Der "user" gibt an, mit wessen Privilegien dieses Programm gestartet werden soll. Viele Dienste benötigen Zugriff auf geschützte Systemressourcen, so daß dort oftmals root eingetragen werden muß. Vergewissern Sie sich aber **unbedingt**, ob ein Programm für diesen Betriebsfall ausgelegt ist, bevor Sie es hier eintragen und als root laufen lassen. ungeeignet geschriebene Netzwerkdaemonen sind die häufigste Quelle für ungebetene Gäste (Cracker) auf dem Rechner.

In diesem einfachen Fall ist der Dienst "internal", d.h. es wird kein externes Programm gestartet, um die Anfrage zu beantworten, sondern der inetd kennt einige triviale Protokolle selbst und behandelt sie intern.

Das ist auch einer der Gründe für die Existenz des inetd. Einige Protokolle sind sehr trivial, und es wäre glatte Ressourcenverschwendung, jedesmal das gesamte Netzwerkhandling neu zu implementieren. Deshalb wurden einige zusammengefaßt und werden vom inetd intern implementiert.

Der zweite Grund liegt ebenfalls im Bereich Ressourcenschonung. würde man für jeden Dienst einen eigenen Serverprozess laufen lassen, so würde man relativ viel RAM (bzw. swap) mit diesen Daemonen belasten, obwohl sie nur relativ selten benutzt werden. Mit Hilfe des inetd werden die eigentlichen Programme nur geladen, wenn sie auch angesprochen werden.

Solche nicht-internen Dienste werden z.B. wie folgt eingetragen:

```
telnet  stream  tcp     nowait  root    /usr/sbin/tcpd  in.telnetd
```

Dieser Eintrag ist für telnet, also den entfernten Shell-Zugriff auf die Maschine. Das Argument nach dem "user", also "server-path" ist der Pfad zum auszuführenden Programm, in diesem Fall /usr/sbin/tcpd.

Danach kommen die an dieses Programm zu übergebenden Argumente. Achtung - hier wird auch `argv[0]` übergeben, also der Name des aufgerufenen Programmes. Im Regelfall wird dieser mit dem Dateinamenanteil des "server-path" übereinstimmen — das muß aber nicht sein. Diese Technik erlaubt wiederum, Programme zu nutzen, die sich abhängig von `argv[0]` verschieden verhalten.

In unserem Beispiel wird dieses Verhalten genutzt, um die sog. TCP-Wrappers anzusteuern. Dabei handelt es sich um ein Sicherheitsprogramm, das zunächst die angeforderte Verbindung auf Zulässigkeit prüft und dann das in `argv[0]` angegebene Programm aufruft oder die Verbindung beendet.

Bitte beachten Sie auch, daß ältere Versionen des inetd nicht in der Lage sind sehr viele oder sehr lange Parameter an das aufgerufene Programm zu übergeben. Wenn sich also ein Daemon seltsam verhält, ist dies eine mögliche Fehlerursache, die Sie einfach mit einem kleinen Script umgehen können.

Die Faustregel für eine sichere Konfiguration lautet:

Alles Abschalten, was nicht unbedingt notwendig ist.

Das gilt insbesondere für die "Trivialservices" echo, discard und chargen, da diese durch gefälschte Pakete leicht dazu benutzt werden können, sinnlos systemlast zu erzeugen.

Die Protokolle telnet und ftp sollten vermieden werden, da diese Passwörter im Klartext über das Netz schicken, was in den meisten Fällen leicht abgehört werden kann. In abgeschlossenen Privaten Netzen sind diese Dienste u.U. akzeptabel, bei an das Internet angeschlossenen Maschinen sollten sie aus sein. Ggf. kann man mit TCP-Wrappers ihre Nutzung auf das interne Netz beschränken.

NNTP und SMTP sollte man nach Bedarf einschalten, wobei hier allerdings meist ein dezidierter Start des zuständigen Daemonen vorzuziehen ist. Achten Sie bei diesen Systemen (insbesondere bei SMTP) unbedingt auf aktuelle Versionen, das Einspielen aller verfügbaren Sicherheitspatches und korrekte Konfiguration wenn Ihr System mit dem Internet verbunden ist.

SMTP, bzw. der beliebte Server "sendmail" war ein sehr gern genutzter Angriffspunkt. Insofern ist es oft nützlich, nonstandard-Programme einzusetzen, da deren Schwachpunkt zumindest nicht ganz so bekannt sind, wie die der am häufigsten verwendeten Programme. Allerdings hilft das auch nur, sofern deren Programmierer sich ausführliche Sicherheitsgedanken gemacht haben — ein "richtig schlecht" programmiertes Ersatztool ist natürlich eher noch gefährlicher.

Sofern Sie nicht eine zwingende Anwendung dafür haben, schalten Sie **ALLE** r-Dienste (rlogin, rsh, rexec) **unbedingt aus**. Diese Dienste gelten als hochgradig unsicher.

Die Post-Office Protokolle pop2 (veraltet), pop3 und imap schalten Sie bitte auch nur bei Bedarf (auf dem Incoming Mailserver) ein.

Gleiches gilt für bootp und tftp, die zum booten laufwerksloser Clients über das Netz verwendet werden.

Von den Informationsdiensten systat, netstat und finger ist allenfalls finger vertretbar, wobei auch dieser Dienst von Crackern gerne als Informationsquelle bezüglich selten benutzter Accounts etc. benutzt wird. Verwenden Sie zumindest einen etwas weniger "geschwätzigen" Finger-Daemon.

Für alle weiteren Dienste in der inetd.conf, mit denen Sie nichts anfangen können gilt die obige Grundregel: Ausschalten.

Stellen Sie fest, daß Sie die Dienste doch benötigen, schalten Sie sie eben später wieder ein - am besten erst nachdem Sie sich über die Sicherheitsaspekte des jeweiligen Dienstes ausführlich informiert haben.

Nach dem Editieren der `/etc/inetd.conf` werden diese Änderungen nicht sofort wirksam. Benutzen sie das Kommando `killall -HUP inetd`, um den Inet-Daemon dazu zu bewegen, seine Konfigurationsdatei neu einzulesen.

Sichern Sie soweit irgend möglich alle Dienste mit TCP-Wrappers.

TCP-Wrappers

Wie bereits erwähnt, ist es möglich, die Verfügbarkeit von Diensten weiter einzuschränken, indem man statt dem eigentlichen Daemon /usr/sbin/tcpd aufruft, der dann mit Hilfe von `argv[0]` den eigentlichen Prozess startet, nachdem er die Zugangsberechtigung des jeweiligen Hosts geprüft hat.

Dazu stellt der tcpd die IP-Adresse des anfragenden Systems fest und vergleicht sie mit einem Regelsatz in `/etc/hosts.allow` und `/etc/hosts.deny`.

Bitte lesen Sie die Manpages zu `tcpd(8)` und `hosts_access(5)`, um einen Einblick in dieses System zu erhalten.

Als empfohlene Grundkonfiguration tragen Sie in `/etc/hosts.allow` nichts ein, und in `/etc/hosts.deny`:

```
# See tcpd(8) and hosts_access(5) for a description.
ALL: ALL
```

Diese Konfiguration verweigert den Zugriff auf **alle Dienste** (erstes ALL), von **allen IP-Adressen**.

Wenn Sie zum lokalen Test allen Hosts aus der eigenen Domain (alle die keinen . im vollständigen Hostnamen haben) erlauben wollen, können sie den Ausdruck `ALL EXCEPT LOCAL` verwenden bzw. in die hosts.allow `ALL: LOCAL` eintragen.

Beachten Sie, dass einige Clients diese Funktionalität auch bieten, wenn sie nicht über den tcpd gestartet werden.

Insbesondere, wenn Sie nur remote (via telnet o.ä.) mit dem System verbunden sind, sollten Sie Änderungen an der tcpd-Konfiguration immer nur mit Vorsicht durchführen, damit Sie nicht versehentlich sich oder anderen legitimen Nutzer den Zugang zum System verwehren oder durch zu grosszügige Rechtevergabe Sicherheitslöcher schaffen.

Kapitel 6

Fehlersuche

6.1 Konfigurationsänderungen

6.1.1 Vorsichtsmaßnahmen

Konfigurationsänderungen an sich in Betrieb befindlichen Systemen (also Systemen, die gerade anderen Nutzern irgendwelche Dienste erbringen) sollte man möglichst vermeiden bzw. man sollte zumindest die Nutzer auf einen möglichen Ausfall des Systems während der Wartungsarbeiten hinweisen.

Wenn Sie nicht 100%ig sicher sind, wie eine Konfigurationsänderung korrekt durchgeführt wird, "üben" Sie sie zunächst an einem System dessen Ausfall nicht kritisch ist - am bestem auf einer eigens für diesen Zweck bzw. nur von Ihnen benutzten Maschine.

Bei Änderungen, die potentiell den vorhandenen Datenbestand gefährden, wie der Repartitionierung einzelner Platten, der Neuformatierung von Dateisystemen, der Rekonfiguration von RAID-Arrays etc. stellen sie **unbedingt** sicher, daß aktuelle Backups existieren. Am besten fahren Sie das System in den Single-User-Mode herunter und machen noch einmal direkt vor der Änderung ein Backup.

Stellen Sie auch regelmässig sicher, daß alle gemachten Backups auch wirklich funktionstüchtig und rekonstruierbar sind. Nicht mehr lesbare Backups durch abgenutzte Bänder oder antike Konfigurationen des Backup- Programmes, die viele neu hinzugekommene Dateibäume ignorieren führen oft zu irreparablen Fehlern.

Bevor Sie Dateien editieren, legen Sie ein Backup dieser Dateien an, z.B. durch `cp config.dat config.dat.old`. Verlassen Sie sich **nicht** auf die integrierte Backupfunktion des Editors. Dieser legt in der Regel nur ein Backup der letzten Version der Datei an — müssen Sie dann noch nach der ersten groesseren Änderung z.B. einen kleinen Tippfehler korrigieren, ist die Originaldatei bereits verschwunden. Durch explizites Sichern der Datei im funktionsfähigen Zustand haben Sie jederzeit die Möglichkeit dorthin zurückzukehren.

Bei kleineren Änderungen ist es bei allen gängigen Dateiformaten natürlich auch möglich das "Backup" in der Datei selbst abzulegen, z.B. so:

```
# ge"andert am 06.06.2000 von
# /dev/sda6       /home                     ext2            defaults   1   2
# Grund: Neue Platte f"ur /home eingeh"angt.

/dev/sdb1       /home                     ext2            defaults   1   2
```

Man sichert also den ursprünglichen Eintrag indem man ihn kopiert, die eine Kopie mit einem Kommentarzeichen (und ggf. einem Kommentar) versieht, und die andere editiert. Diese Vorgehensweise verhindert oft auch Formatfehler, da man sich ja an dem vorhandenen Eintrag orientieren kann.

6.1.2 Aktivierung von Änderungen

Im Gegensatz zu anderen Betriebssystemen ist es unter Unix höchst unüblich Konfigurationsänderungen durch einen Reboot zu aktivieren. Bis auf das Laden eines neuen Kerns, sollte es in der Regel nicht nötig sein, ein Unix zu rebooten, um eine neue Konfiguration zu aktivieren.

Lediglich, wenn Sie bestimmte Änderungen zunächst zu Testzwecken "zu Fuß" gemacht haben, und sie erst anschließend in die entsprechenden Konfigurationsdateien eingetragen haben, ist es u.U. sinnvoll probehalber einen Reboot auszuführen, um die Korrektheit und Vollständigkeit dieser Änderungen zu testen. Aufgrund der in der Regel langen "uptimes" (Zeit zwischen zwei reboots) von Unix Systemen besteht sonst die Chance, daß der Fehler erst beim nächsten Strom- oder Hardwareausfall bemerkt wird, wobei sich in der Regel niemand mehr daran erinnern kann, welche Änderung es gewesen sein könnte.

Je nach Subsystem gibt es verschiedene Methoden, dem laufenden Prozess Konfigurationsänderungen klar zu machen:

- Einige Programme erkennen selsttätig Änderungen an ihrer Konfigurationsdatei. Ein Beispiel dafür ist i.a. `cron`, der in regelmässigen Abständen das Änderungsdatum seiner Konfigurationsdatei überwacht. Das gleiche gilt für Programme, die regelmässig neu gestartet werden, wie z.B. die vom inetd aufgerufenen Programme, sofern diese eigene Konfigurationsdateien haben. Diese werden dann logischerweise beim Start gelesen, so daß sich Änderungen quasi "sofort" auswirken.

- Andere Programme werden mit einem speziellen "Signal" auf die Konfigurationsänderung aufmerksam gemacht. Dazu sendet man ihnen mit Hilfe von `kill`, `killall` oder `killproc` (siehe Manpages) das jeweils in der Manpage verzeichnete Signal. Der `inetd` ist ein Beispiel fur ein solches Programm. Sendet man ihm SIGHUP, so liest er seine Konfiguration neu ein.

- Manche Programme müssen beendet und neu gestartet werden. dazu sendet man Ihnen mit einem der `kill*`-Befehle ein `TERM-` (in einigen hartnäckigen Fällen, wie dem pppd auch ein `QUIT-`) Signal, woraufhin sich diese Prozesse beenden. Sollte der Prozess danach immer noch laufen (mit `ps` überprüfen), bleibt noch die "Vorschlaghammermethode" mit dem "KILL" Signal. Letzteres kann vom Programm nicht abgefangen werden und führt somit **immer** zur Terminierung. Allerdings hat das Programm dann auch keine Chance, sich sauber zu beenden ...

Bei dem auf Linux gängigen SysV-Init-Schema, gibt es allerdings eine elegante Methode, die es einem erspart in den Manpages nach den einzelnen Methoden für den gewünschten Dienst zu suchen:

In `/etc/rc`d/init.d finden Sie die sogenannten Startup-Scripte, die vom System benutzt werden, um einzelne Dienste beim Systemstart hochzufahren, bzw. beim Shutdown wieder sauber herunterzufahren.

Diese Scripte nehmen jeweils einen Parameter, mit dem ihr Verhalten gesteuert wird:

start Starte den zugehörigen Dienst.

stop Halte den zugehörigen Dienst an.

restart Stoppe den zugehörigen Dienst, dann starte ihn neu.

reload Lade die Konfigurationsdateien des Dienstes neu.

status Zeige an, ob dieser Dienst laeuft.

probe zeige an, ob der Dienst einen Reload braucht.

Sie können also z.B. mit

```
bash$ cd /etc/rc.d/init.d
bash$ ./inetd reload
```

Den inetd dazu veranlassen, seine Konfigurationsdateien neu zu lesen, egal wie das bei Ihrer konkreten Version des inetd geregelt ist.

In einigen Fällen werden Sie evtl. gezwungen sein auf “restart” auszuweichen, wenn der Daemon reload nicht korrekt implementiert oder durch eine vorherige Fehlkonfiguration so “verwirrt” wurde, daß dieser Mechanismus gestört ist.

Haben Sie den Eindruck, daß auch “restart” nicht den gewünschten Erfolg zeigt, so geben Sie mehrfach das “stop” Kommando und überprüfen Sie den Erfolg mit ps. Benutzen Sie ggf. selbst das `kill` Kommando, um eventuelle Fehlfunktionen in diesem Bereich zu übersteuern und starten Sie das Subsystem anschliessend mit “start” neu.

6.1.3 Automatische Konfigurationstools

Diese Hilfprogramme sind in zwei Kategorien einzuteilen:

1. Programme, die die Original-Konfigurationsdateien der entsprechenden Subsysteme einlesen, analysieren und verändern. Hierunter fällt z.B. Linuxconf, das RedHat printtool, etc. .

2. Programme, die eine eigene Konfigurationsdatei verwalten und daraus die eigentlichen Konfigurationsdateien der Subsysteme generieren (YAST).

Besonders die zweite Kategorie erfordert Umsicht bei manuellen Änderungen. Während bei der ersten nur die Gefahr besteht, daß das Konfigurationstool die Datei nicht mehr versteht und den Dienst verweigert, so daß man weitere Änderungen auch in Zukunft nur noch per Hand erledigen kann, kann es bei der zweiten passieren, daß ein Aufruf des automatischen Konfigurationstools mühevoll per Handarbeit erledigte Änderungen mit der aus der internen Konfigurationsdatei resultierenden Version überschreibt.

Die oben erwähnten Vorsichtsmaßnahmen gelten bei der Benutzung automatischer Konfigurationstools doppelt, da diese in der Regel **kein** Backup anlegen.

6.2 Programminstallation

6.2.1 Vorsichtsmaßnahmen

Denken Sie daran, daß ein installiertes Programm alles tun kann, was der Benutzer, der es aufruft tun kann.

Installieren Sie daher **niemals** Programme aus zweifelhaften Quellen - schon gar nicht irgendwelche von Unbekannten per EMail verschickten Programme und lassen Sie auch bei bekannten Absendern ein gehöriges Maß an Mißtrauen walten. EMail-Würmer sind auch für Linux denkbar !

Es gab auch bereits Fälle, wo auf angesehenen Servern bekannte Programmpakete von Crackern durch manipulierte Versionen ersetzt wurden, die Hintertüren in das System eingeführt haben.

Verwenden Sie wenn irgend möglich ein krytographisches Verfahren (Signatur), um die Integrität des Pakets zu überprüfen (z.B. mit `rpm --checksig`).

6.2.2 Mit Packaging-Tools

Die Installation von Programmen mit Hilfe von Packaging-Tools ist in der Regel problemlos. Geben Sie einfach den entsprechenden Befehl an den Package- Manager (z.B. `rpm -i packetname.rpm` bei RPM-basierten Systemen).

Die Fehlermöglichkeiten in diesem Fall sind:

Nicht aufgelöste Abhängigkeiten

Das heißt, daß vor der Installation des gewählten Paketes andere Pakete installiert werden müssen, damit das aktuelle Paket korrekt arbeiten kann.

Installieren Sie zunächst die in der Fehlermeldung genanneten Pakete.

Es besteht zwar die Möglichkeit, diesen Fehler zu übersteuern (–force*), was aber zumeist nicht den gewünschten Erfolg haben wird, da ja tatsächlich Funktionen fehlen, die das zu installierende Programm benötigt. Verwenden Sie diese Option nur in Fällen, in denen Sie ganz sicher sind, dass die gewünschte Funktionalität zur Verfügung steht, und dies lediglich dem Package-Management nicht bekannt ist.

Falsche Distribution

Obwohl durch den Linux Filesystem Standard eine weitgehende Standardisierung der Lage einzelner Dateien im Verzeichnisbaum erreicht wurde, gibt es immer noch kleine Unterschiede, die unter Umständen dazu führen, daß insbesondere Startup-Scripte nicht gefunden werden. Ältere SuSE Distributionen zeichneten sich durch die Unart aus das oben beschriebene init.d-Verzeichnis in /sbin statt in /etc/rc.d zu haben.

Werfen Sie am besten mit Hilfe des `mc` einen Blick **in** das RPM-File und vergleichen Sie die Pfade. Ggf. hilft ein Symlink weiter.

Falsche Version

Wenn Ihre Distribution schon etwas älter ist, kann es sein, daß neue Pakete sich nicht korrekt einspielen lassen. Das liegt zumeist an inkompatiblen Änderungen an Libraries etc. (der umgekehrte Fall ist auch möglich, aber recht selten, da zumeist sogenannte "compatibility-libs" mitgeliefert werden).

In diesem Fall müssen Sie entweder die Distribution updaten (zumeist sowieso eine gute Idee) oder nach einer älteren Version des Softwarepackages suchen. Mißlingt auch dies, bleibt die Installation aus dem Source (via einem SRPM oder einen tar.gz).

Da bei der Compilierung des Source die jeweils installierten Libraries verwendet werden, gibt es keine Probleme mit internen Änderungen. Nur in Ausnahmefällen (bei systemnahen Programmen bzw. wenn sich die externen Interfaces von Libraries geändert haben) kann es zu Komplikationen kommen.

6.2.3 Aus dem Source

Source-RPMs

Verwenden Sie einfach die "rpm –rebuild" bzw. "rpm –recompile" Optionen (siehe auch die Manpage). Leider haben Sie hierbei wenig Möglichkeiten in den Verlauf einzugreifen, so daß Sie im Fehlerfall (ausser bei den bereits oben erwähnten Fehlern) lediglich ein neues (hoffentlich funktionierendes) srpm suchen können, oder sich tiefer mit den "Innereien" von RPM beschäftigen müssen.

.tar.gz

Lesen Sie als erstes **unbedingt** die Inhalte der Hilfedateien, insbesondere von Dateien wie `README`, `README.FIRST`, `INSTALL` etc.

Manche Programme benötigen spezielle Installationsprozeduren, und "blindes" Installieren nach Standardprozeduren führt zu unbrauchbaren Ergebnissen.

Insbesondere kann es sein, daß Sie einige systemspezifische Voreinstellungen in Dateien treffen müssen, Dateien mit Beispielkonfigurationen editieren und/oder umbenennen müssen, etc..

Clever gemachte tgzs sorgen dafuer, daß durch Unterlassen der entsprechenden Schritte eine Fehlermeldung ausgeloest wird, aber in vielen Faellen wird einfach mit einer Standardkonfiguration compiliert, was dann erst spaeter zu Problemen fuehrt.

.tar.gz mit autoconf

Viele GNU-Programme verwenden autoconf, um die Programme automatisch an verschiedene Unix-Plattformen anzupassen.

Sie erkennen diese Programme an einer Datei namens "configure" im Hauptverzeichnis.

Auch hier gilt: Erst die Dokumentation lesen!

`./configure --help` zeigt Ihnen die möglichen Konfigurationsoptionen an. Es kann aber sein, daß auch hier zunächst bestimmte Dateien editiert werden müssen. Ist dies geschehen, so ist die Standardprozedur zur Installation

```
./configure [--options, falls erforderlich]
make
[ggf. make check, falls das Paket Selbsttests beinhaltet]
su -c "make install"
```

Beachten Sie, daß alle Schritte als **normaler User** stattfinden, bis auf "make install". Bei diesem letzten Schritt werden die Programme und Daten installiert, wozu Schreibzugriff auf Systemverzeichnisse erforderlich ist.

Bitte beachten Sie, daß nicht aus der Distribution stammende Programme nach `/usr/local` gehören. Dort bleiben sie vor Kollisionen mit eventuell später aufgespielten Updates der Distribution verschont und können leichter gezielt gesichert werden (z.B. zum Wiederaufspielen nach einer Neuinstallation). Den Installationspfad sollte man mit der configure-option `--prefix=/usr/local` entsprechend einstellen können, und dies sollte auch der Defaultwert sein.

Entstehen Fehler in der ./configure-Phase, so liegt dies zumeist an nicht vorhandenen Komponenten, die Voraussetzung für das aktuelle Paket sind. Sie sollten eine dahingehende Fehlermeldung erhalten. Erhalten Sie eine andere Meldung, die Sie sich nicht erklären können, lesen Sie noch einmal genau die READMEs (oft gibt es dort eine Troubleshooting-Sektion mit den häufigsten Problemen), und wenn das nicht hilft kontaktieren Sie den Verwalter (Maintainer) des Paketes. Geben Sie ihm eine exakte Beschreibung des Fehlers (am besten kopieren Sie die Ausgabe des configure-Scriptes) und der Schritte, die Sie vor dem Eingeben von "./configure" unternommen haben, sowie eventuell schon versuchte Lösungsideen.

Ggf. wird der Maintainer weitere Informationen brauchen, wie die Files `config.cache` oder `config.log`. Entsprechende Hinweise finden sich in der Regel in den READMEs unter "reporting problems" bzw. der Maintainer wird ggf. nachfragen.

Einige Pakete besitzen auch Diagnose- oder Bugreport-Scripte. Auch diese sollten in den READMEs erwähnt sein.

Fehler in der make-Phase erfordern meist Kenntnisse in der verwendeten Programmiersprache. Sind diese nicht in ausreichendem Maße vorhanden, bleibt wieder nur die Frage an den Maintainer. Auch hier ist es wichtig, daß man diesem alle wichtigen Details mitteilt. Insbesondere sollten die Fehlermeldungen mit denen abgebrochen wird komplett mitgeschickt werden.

Fehlende Bestandteile sollten bereits vom configure-Lauf erkannt worden sein, so daß diese Fehlermöglichkeit zumeist ausgeschlossen ist (bzw. der eigentliche fehler dann im configure-script liegt).

Beim "make install" kommt es nur selten zu Problemen. Zumeist, sind es Pfade, die nicht existieren, das Installationsprogramm aber als gegeben voraussetzt, oder Rechte, die nicht ausreichen (z.B. bei via NFS gemountetem `/usr/local`).

6.3 Netzwerk

Bei Problemen mit TCP/IP-Netzwerken gibt es eine ganze Reihe von möglichen Fehlerquellen, aber auch gute Möglichkeiten der Diagnostik.

Der erste Schritt besteht in der Überprüfung der Hardware, sofern dies einfach möglich ist. Nicht eingesteckte Kabel, gebrochene Adern etc. sind nicht selten der Grund für Totalausfälle der Netzanbindung.

Gute Hinweise auf solche Probleme liefern auch die Link-Status Anzeigen von guten Netzwerkkarten, Hubs und Switches.

Ergibt diese Überprüfung keine Hinweise auf Probleme, so beginnt die Fehlersuche auf dem Host.

Dazu verwendet man als erste Stufe `ping`.

Abbildungsverzeichnis

Tabellenverzeichnis