

Linux-Einführungskurs

C.Möller (cm@iam.de)

Inhaltsverzeichnis

1. Was ist Linux ?

- 1.1. Allgemeines
- 1.2. Hardwareunterstützung
- 1.3. Stabilität
- 1.4. Verfügbare Anwendungssoftware
- 1.5. Anwendungsgebiete
- 1.6. Technische Eigenschaften von Linux

2. Die Installation

- 2.1. Wie startet man die Installation ?
- 2.2. Benötigter Plattenplatz
- 2.3. Partitionierung für ein „Linux-only“-System
- 2.4. Partitionierung bei mehreren installierten Betriebssystemen
- 2.5. Bootmenü bei mehreren installierten Betriebssystemen
- 2.6. Abfrage der Hardware-Informationen
- 2.7. Wichtige Schritte nach erfolgter Installation

3. Die Konfiguration des Systems

- 3.1. X-Window
- 3.2. TCP/IP-Netzwerk
- 3.3. Druckereinrichtung
- 3.4. Kernel neu übersetzen
- 3.5. Sonstiges

4. Umstieg von DOS auf Linux – Befehlskurzübersicht

- 4.1. Kommandos zum Umgang mit Dateien und Verzeichnissen
 - 4.1.1. man – Manual Pages anzeigen
 - 4.1.2. ls – Verzeichnisinhalt ausgeben
 - 4.1.3. cd – Verzeichnis wechseln
 - 4.1.4. pwd – Ausgabe des aktuellen Verzeichnisses
 - 4.1.5. cp - kopieren
 - 4.1.6. mv – verschieben und umbenennen
 - 4.1.7. rm – Dateien löschen
 - 4.1.8. mkdir – Verzeichnisse anlegen
 - 4.1.9. rmdir – Verzeichnisse löschen
 - 4.1.10. cat – Ausgabe von Dateien
 - 4.1.11. less – komfortable Ausgabe von Textdateien
 - 4.1.12. vi – der Standard-Unix-Texteditor
 - 4.1.13. joe – der etwas einfacher zu bedienende Texteditor
 - 4.1.14. chmod – Dateisystemrechte vergeben
 - 4.1.15. chown – Eigentümer und Gruppenzugehörigkeit ändern
 - 4.1.16. mc – Midnight Commander
 - 4.1.17. mtools – DOS-Disketten benutzen
- 4.2. Kommandos zur Systemverwaltung
 - 4.2.1. User und Gruppen verwalten
 - 4.2.2. passwd – Passwort ändern
 - 4.2.3. mount – Festplattenpartitionen ins Dateisystem einbinden
 - 4.2.4. df – Festplattenauslastung anzeigen
 - 4.2.5. kdm – KDE-Windowmanager starten
 - 4.2.6. Alternativer Start des X-Window-Systems
 - 4.2.7. su – Superuser-Kommandos ausführen oder: wie werde ich im Terminal zum root ?
 - 4.2.8. ps, kill – Prozeßinformationen anzeigen und Programme gewaltsam beenden
 - 4.2.9. rpm, tar, gzip – Programmpakete installieren
 - 4.2.10. shutdown, halt, reboot – das System herunterfahren oder neu booten

Inhaltsverzeichnis

5. Weitergehende Informationen

5.1. Netzwerkkommandos

5.1.1. ping

5.1.2. telnet

5.1.3. finger

5.1.4. traceroute

5.1.5. tcpdump

5.1.6. netcat

5.2. Pipes, Umleitungen, Kommandoverkettungen in der bash

5.3. Erweiterte Prozeßverwaltung

6. Anhang

6.1. Copyright

6.2. Disclaimer

1. Was ist Linux?

1.1. Allgemeines

Linux ist eine vollständig frei verfügbare Reimplementation der POSIX-Spezifikation mit SYSV- und BSD-Erweiterungen (im Klartext: es gleicht Unix, stammt aber nicht vom selben Quellcode ab) und besteht im Wesentlichen aus dem Kernel (dem Systemkern), einer einheitlichen Verzeichnisstruktur und diversen Systemprogrammen. Linux ist sowohl als Quellcode als auch in Binärforn erhältlich. Das Copyright liegt bei Linus Torvalds (dem „Ur“-Entwickler) und anderen. Das System ist frei erhältlich und unterliegt den Bestimmungen der GNU General Public License (GPL). Eine Kopie der GPL gehört zum Quellcode von Linux.

Verschiedene Distributoren (S.u.S.E., Red Hat, DLD, Slackware etc.) stellen diese Grundkomponenten zu einem lauffähigen und benutzbaren System zusammen.

Linux ist weder „Public Domain“ noch „Shareware“. Es ist „freie“ Software, sogenannte Freeware. Man kann Kopien weitergeben oder verkaufen, aber man muß den Quellcode mitliefern oder ihn genauso verfügbar machen wie jede Binärdatei, die man weitergibt oder verkauft. Falls man veränderten Code weitergibt, ist man rechtlich verpflichtet, den Quellcode für diese Änderungen mitzuliefern.

Linux v2.x.x ist noch immer frei verfügbar und wird es auch weiterhin bleiben. Aufgrund der GPL, der Linux unterliegt, wäre es illegal, Linux nicht frei auszuliefern. Wichtig: „frei“ bezieht sich auf den Zugang zum Quellcode und nicht auf die Kosten. Es ist absolut legal, Geld für Linux-Distributionen zu verlangen, so lange man auch den Sourcecode mitliefert. Dies ist jedoch nur eine Verallgemeinerung, für Feinheiten sollte man die GPL lesen.

1.2. Hardwareunterstützung

Linux läuft auf 386, 486, Pentium, PII und allen anderen x86-kompatiblen Rechnern mit ISA-, EISA-, PCI- oder VLB-Bus. Die Unterstützung für MCA (IBM Microchannel) ist bislang nur experimentell. Desweiteren sind Portierungen in unterschiedlichen Entwicklungsstufen für Motorola 680x0 (Amiga, Atari etc.), DEC Alpha, Sun SPARC, PowerPC, MIPS, ARM erhältlich.

Man kann davon ausgehen, daß darüber hinaus nahezu sämtliche gebräuchliche Zusatzhardware unterstützt wird. Detaillierte Listen der unterstützten Hardware und der Hardwareanforderungen erhält man in der Regel auf den Homepages der Linux-Distributoren.

1.3. Stabilität

Linux gilt längst nicht mehr als Programm in der Betaphase, da die Version 1.0 am 14. März 1994 herausgebracht wurde. Es gibt immer noch Fehler im System und im Laufe der Zeit werden neue auftauchen und beseitigt werden. Da Linux dem ``offenen Entwicklungsmodell'' folgt, werden alle neuen Versionen der Öffentlichkeit zur Verfügung gestellt, ob sie nun ``Serienreife'' haben oder nicht.

Um den Anwendern trotzdem die Unterscheidung zwischen einer stabilen und einer Beta-Version zu ermöglichen, bedient man sich des folgenden Schemas:

Die Versionen 1.x.y sind bei geradem x stabile Versionen; bei Beseitigung von Fehlern wird y erhöht. Beim Wechsel von Version 1.2.2 auf 1.2.3 wurden also nur Fehler bereinigt, es gab keine neuen Eigenschaften.

Die Versionen 1.x.y mit ungeradem y sind Beta-Versionen, die hauptsächlich für Entwickler geeignet sind. Sie können instabil sein oder abstürzen, und es werden ständig neue Eigenschaften hinzugefügt. Von Zeit zu Zeit, wenn der aktuelle Entwicklungs-Kernel stabil ist, wird er als neuer ``stabiler'' Kernel eingefroren und die Entwicklung wird mit einer neuen Entwicklungsversion des Kernels fortgesetzt.

Derzeit ist ca. 2.2.5 die aktuellste stabile Version (dies ändert sich sehr oft, z.B. wenn neue Gerätetreiber hinzugefügt oder Fehler entfernt werden). Man beachte, daß man für ein Kernelupdate auch einige Hilfsprogramme erneuern muß. Um diese Hilfsprogramme zu erhalten, sollte man auf die neuste Version seiner Linuxdistribution wechseln, um eine optimale Anpassung der einzelnen Pakete an das Gesamtsystem zu erhalten.

Den aktuellen Entwicklungsstand erfährt man aus erster Hand unter <http://www.kernel.org>.

1.4. Verfügbare Anwendungssoftware

Für Linux steht ein immenser Softwarepool zur Verfügung.

Standardprogramme für die tägliche Arbeit (Office, WWW, E-Mail, Grafikbearbeitung etc.) liegen ebenso vor wie Spezialsoftware (DTP, CAD, Sound- u. Videobearbeitung, diverse Netzwerkprogramme etc.).

Der Großteil der unter Linux erhältlichen Software unterliegt ebenfalls diversen Freeware-Lizenzen und wird in unentgeltlicher Freizeitarbeit von engagierten Hobbyprogrammierern entwickelt.

Mehr und mehr Firmen erkennen mittlerweile die Marktchance Linux und entwickeln auch kommerzielle Software, die vergleichbaren Produkten für kommerzielle Betriebssysteme in nichts nachsteht.

Einen guten Überblick über den Stand der Entwicklung im Bereich der Anwendungen erhält man bei sog. Linux-Portal-Sites, wie z.B. <http://www.linux.org>, [.de](http://www.linux.de), [.com](http://www.linux.com) usw., die aktuelle Berichte über Linux veröffentlichen.

1.5. Anwendungsgebiete

Linux-Systeme werden heutzutage in allen Bereichen der Computeranwendungen erfolgreich eingesetzt. Vom reinen Arbeitsplatz-PC über die Grafikworkstation, das Entwicklungssystem, den Internet-, Mail-, Fax-, Telefon- oder ftp-Server, das Internet-Gateway, Echtzeitanwendungen zu Automatisierungszwecken bis hin zum massiven Parallelrechnen – Linux hat so gut wie jede Domäne erobert.

Aufgrund seiner ausgezeichneten Netzwerkfähigkeiten liegt der Schwerpunkt des Einsatzes von Linux natürlich bei Internet-, Intranet- und Serverdiensten.

1.6. Technische Eigenschaften von Linux

- Multitasking: mehrere Programme laufen zur selben Zeit.
- Multiuser: mehrere Benutzer arbeiten gleichzeitig auf demselben.
- Multiplattform: läuft auf vielen verschiedenen CPUs, nicht nur Intel.
- Multiprozessor: SMP-Unterstützung steht auf Intel- und SPARC-Plattformen zur Verfügung (an anderen Plattformen wird gearbeitet). Linux wird für verschiedene offene Multiprocessing-Anwendungen benutzt, Beowulf-Systeme (Netzwerk-Supercomputing) und der SPARC-basierte Fujitsu AP1000+ Supercomputer eingeschlossen.
- hat Speicherschutz zwischen Prozessen, so daß ein Programm nicht das ganze System zum Absturz bringen kann
- Ausführbarer Code wird nach Bedarf geladen: Linux liest nur diejenigen Teile eines Programms von Platte, die tatsächlich benutzt werden.
- Virtueller Speicher benutzt Paging auf Festplatte: Es werden nicht komplette Prozesse ausgelagert, sondern nur soviele Speicherseiten, wie ein anderer Prozeß gerade anfordert. Dies geschieht auf eine separate Partition oder in eine Datei im Dateisystem oder beides, mit der Möglichkeit, während des Betriebs Auslagerungsbereiche hinzuzufügen (ja, man spricht immer noch von Auslagerungsbereichen). Insgesamt können 16 dieser 128 MB großen Bereiche auf einmal benutzt werden. Das ergibt theoretisch einen Gesamtwert von 2 GB an verwendbarem Auslagerungsbereich. Bei Bedarf kann dies leicht durch Ändern einiger Zeilen im Quellcode erweitert werden.
- Ein gemeinsamer Speicherpool für Anwender-Programme und Plattenpuffer, so daß der gesamte freie Speicher für den Cache verwendet werden kann, der Cache aber reduziert werden kann, wenn große Programme laufen.
- Dynamisch gelinkte Shared Libraries (DLL's), und natürlich auch statisch gelinkte Libraries.
- Führt „core dumps“ (Speicherabzüge) für Post-Mortem-Analysen aus. Dadurch wird die Anwendung eines Debuggers auf ein Programm nicht nur während dieses läuft, sondern auch nachdem es abgestürzt ist, möglich.
- Weitgehend kompatibel mit POSIX, System V und BSD auf der Ebene des Quellcodes.
- Der gesamte Quellcode ist verfügbar, den kompletten Kernel, alle Treiber, die Entwicklungswerkzeuge und alle Nutzerprogramme eingeschlossen. Alles ist auch frei verteilbar. Es gibt eine Vielzahl an kommerziellen Programmen für Linux, die ohne Sourcecode ausgeliefert werden, aber alles, was einmal frei war, das gesamte Betriebssystem eingeschlossen, ist immer noch frei erhältlich.
- Unterstützung für viele nationale oder speziell angepaßte Tastaturen.
- Mehrere virtuelle Konsolen: mehrere unabhängige Zugänge zugleich über die Konsole. Man kann dabei durch eine spezielle Tastenkombination hin- und herschalten (unabhängig von der

Grafikhardware). Die virtuellen Konsolen werden dynamisch angelegt. Bis zu 64 Stück kann man benutzen.

- Unterstützt verschiedene gängige Dateisysteme, wie z.B. Minix, Xenix und alle gängigen System-V-Dateisysteme; hat ein ausgefeiltes eigenes Dateisystemschemata, das Dateisysteme bis zu 4 TB mit Dateinamen von bis zu 255 Zeichen Länge bietet.
- Transparenter Zugriff auf MS-DOS Partitionen (oder OS/2 FAT-Partitionen) über ein spezielles Dateisystem: man braucht keine speziellen Befehle zur Benutzung der MS-DOS-Partition, sie präsentiert sich einfach wie ein normales UNIX-Dateisystem (abgesehen von Beschränkungen für Dateinamen, Berechtigungen, etc). Mit MS-DOS 6 komprimierte Partitionen funktionieren derzeit nicht ohne Patch (dmsdosfs). VFAT (Windows 95/98) wird ab Linux 2.0 unterstützt.
- Ein spezielles Dateisystem namens UMSDOS, das es ermöglicht, Linux in einem DOS-Dateisystem zu installieren.
- Lesezugriff für HPFS-2 von OS/2 2.1 wird unterstützt.
- HFS-Dateisystem-Unterstützung (Macintosh) ist separat als Modul erhältlich.
- CD-ROM-Dateisystem, das alle gängigen CD-ROM-Formate liest.
- TCP/IP Netzwerkunterstützung, einschließlich ftp, telnet, NFS etc.
- Appletalk Server
- Netware Client und Server
- Lan Manager (SMB) Client und Server
- Viele Netzwerkprotokolle: die neuen Kernel beinhalten TCP, IPv4, IPv6, AX.25, X.25, IPX, DDP (Appletalk), NetBEUI, u.a.

2. Die Installation

Um es vorweg zu nehmen, die beste Unterstützung bei der Installation eines Linux-Systems bietet meistens die Dokumentation der eingesetzten Linux-Distribution. Besonders zu empfehlen ist hier die S.u.S.E.-Distribution, deren Handbuch kaum Fragen zur Installation und Konfiguration offen läßt. Auch auf den schnellen Einstieg in ein laufendes Linux-System wird dort ausführlich eingegangen, sowohl mit einer Erklärung der wichtigsten Befehle unter Linux, als auch einer detaillierten Beschreibung der Konfiguration vieler mitgelieferter Anwendungspakete (KDE, Intranet-, Internet-, und Systemwerkzeuge, Emulatoren u.v.m.).

Daher folgen hier nur einige grundlegende Betrachtungen zur Linux-Installation:

2.1. Wie startet man die Installation ?

Das Linux-Installationsprogramm kann auf viele verschiedene Arten gestartet werden:

- unter DOS (bei S.u.S.E.): setup.exe auf der Linux-CD aufrufen
- per Systemstart von der Linux-CD: vollautomatisch; benötigt bootfähiges CD-Rom-Laufwerk und BIOS-Unterstützung
- per mitgelieferter Bootdiskette
- von einer FAT-Partition aus: wenn das eingesetzte CD-Rom-Laufwerk nicht von Linux unterstützt wird; Installationsdaten werden vorher unter DOS auf die Platte kopiert.

Weitere Möglichkeiten, die Installation zu starten, nennen die Distributoren meist in README-Dateien auf den Installationsmedien. Der weitere Verlauf der Installation ist – je nach Distribution – menügeführt und relativ selbsterklärend, oft auch schon im Grafikmodus.

Auch hier gilt, wie so oft unter Linux, daß Fehlermeldungen, Hinweise, Warnungen und Tips wirklich gelesen werden sollten, da sich dadurch schon die meisten der auftauchenden Fragen beantworten lassen :-)

Einigen Frust während der Installation kann man sich auch ersparen, indem man die Interna seines PCs vor dem Beginn ermittelt (Festplattenbelegung, Netz-, Grafik- und Soundkartentyp usw.).

2.2. Benötigter Plattenplatz

Ein durchschnittliches Linux-Basissystem mit den wichtigsten Anwendungsprogrammen (Netscape, StarOffice, gimp u.a.), dem X-Window-System und KDE (der Benutzeroberfläche), dem Kernel Quellcode, einem lauffähigen C/C++-Entwicklungssystem (für die Kernelübersetzung), Hardwaretreibern und gerätespezifischer Software (Scannerprogramme, Sound-/TV-Software o.ä.) benötigt einigen Festplattenspeicher, grob geschätzt 1GB für die Installationspartition.

Ein Netzwerkserver benötigt erheblich mehr Speicherplatz, je nach seiner Aufgabe.

Natürlich lassen sich für dieses Einsatzgebiet keine genauen Angaben machen. Ein Mailserver benötigt z.B. viel Platz für zu verarbeitende E-Mail, ein Fileserver braucht enorme Plattenkapazität für die zu speichernden Benutzerdaten, ein Webserver ist eher genügsam, während ein Internet-Cache (Proxy) in einem großen Netzwerk wiederum Unmengen an Plattenplatz beanspruchen kann. Beim Einsatz eines Linux-PCs als Netzwerkservice-Rechner ist also vor der Installation genau abzuwägen, mit welchen Spitzenbelastungen zu rechnen ist. Da Linux jedoch gut mit sehr großen Partitionen und Laufwerken umgehen kann, sind hier eher durch finanzielle Gründe Grenzen gesetzt.

Genauer zum Einsatz als Netzwerkserver entnimmt man am Besten den *HOWTO*s, die zu fast jedem erdenklichen Szenario erhältlich sind und bei den meisten Linux-Distributionen auch lokal (unter `/usr/doc/` o.ä.) zu finden sind.

Als Faustregel liegt man bei einer Serverinstallation mit einer Root-(Installations-)Partition von 1GB und weiteren Gbytes je für Daten-, Spool-(Verarbeitungs-) und Cache-Partitionen erst einmal auf der sicheren Seite :-)

Desweiteren benötigt ein Linux-System in der Regel eine Swap-(Auslagerungs-)Partition, die bei Bedarf als virtuelle Hauptspeichererweiterung genutzt wird. Diese sollte mindestens die doppelte Größe des Hauptspeichers in MB besitzen, jedoch macht es keinen Sinn, mehr als 128 MB pro Swap-Partition zu reservieren (mehr unterstützt der Linux-Kernel nicht). Jedoch können bis zu 16 Swap-Partitionen genutzt werden.

2.3. Partitionierung für ein „Linux only“-System

Mit dem Linux-Installationsprogramm wird eine primäre Partition für das Root-Filesystem (also die Installationspartition) in der gewünschten Größe angelegt. Weitere Partitionen für Homeverzeichnisse, Daten oder Backup (je nach Bedarf, zur Not ist eine großzügig bemessene Rootpartition ausreichend) und die Swap-Partition legt man am besten als logische Partitionen in einer erweiterten Partition an, da bei PC-Systemen nur maximal vier primäre Partitionen pro Festplatte verwaltet werden können. Alle Partitionen müssen dann (mit dem jeweiligen Installationsprogramm) in das Verzeichnissystem eingebunden (gemounted) werden, um benutzbar zu werden. Der Rest der Installation läuft strikt nach den Anweisungen der Installationsprogramme ab.

2.4. Partitionierung bei mehreren installierten Betriebssystemen

Linux „verträgt“ sich generell mit jedem anderen Betriebssystem und sollte daher auf einer eigenen Partition installiert werden, so daß mehrere Betriebssysteme im Wechselbetrieb auf dem PC gefahren werden können.

Prinzipiell läßt sich Linux auch auf einer DOS-Partition installieren, jedoch verliert man in diesem Fall etliche Vorteile des Systems (lange Dateinamen, Geschwindigkeit der Plattenzugriffe, Sicherheit, Kompatibilität). Von einer derartigen Installation ist also dringend abzuraten. Linux benutzt viel lieber ein eigenes Filesystem namens e2fs (Second Extended File System), das auf jeden Fall zu bevorzugen ist.

Grundsätzlich sollte darauf geachtet werden, Linux immer als letztes Betriebssystem auf dem Computer zu installieren, da einige andere Betriebssysteme (und deren Installationsprogramme) recht sauer auf die Konkurrenz alternativer Systeme reagieren. Windows z.B. überschreibt ohne Warnung Linux-Bootsektoren, d.h. ein zuvor installiertes Linux wird nach einer Windows-Installation erst einmal nicht mehr ohne weiteres starten. Sind schon alle anderen Betriebssysteme installiert, bringt die Linux-Installation keine größeren Schwierigkeiten mit sich.

Steht zur Installation nicht mehr genug freier Festplattenplatz bereit (z.B. weil große, teils ungenutzte Windows-Partitionen eingerichtet sind), läßt sich mit geeigneter Software (z.B. mit PQ-Magic von PowerQuest oder dem Freeware-Tool FIPS) Platz schaffen, ohne daß Daten verlorengehen. Derartige Programme sind in der Lage, die Größe von Partitionen zu ändern, ohne die darauf befindlichen Daten zu löschen. Natürlich sollte vor einer solchen Aktion trotzdem ein komplettes Backup des bestehenden Systems erfolgen :-)

Weitere Informationen entnimmt man am besten der Dokumentation des eingesetzten Partitionierungswerkzeugs.

Nach erfolgter Partitionsverkleinerung geht man wie folgt vor:

Steht genügend freier Plattenplatz zur Verfügung, legt man mit dem Linux-Installationsprogramm eine neue erweiterte Partition an, in der logische Partitionen für das Root-Filesystem und für die Swap-Partition angelegt werden, optional noch weitere für Daten, Homeverzeichnisse etc.

Diese logischen Partitionen werden dann sog. Mount Points zugeordnet (ihren Einbindungspositionen im Linux-Verzeichnisbaum).

Die Installation erfolgt dann in diese logischen Laufwerke. Laufwerksbuchstaben, z.B. unter Windows, geraten dabei nicht durcheinander, da Linux-Partitionen für die meisten anderen Betriebssysteme unsichtbar sind.

2.5. Bootmenü bei mehreren installierten Betriebssystemen

Sobald auf einem PC mehrere Betriebssysteme installiert sind, stellt sich die Frage, welches beim Hochfahren des System gestartet werden soll.

Linux bringt dazu den sog. Linux-Loader, kurz LiLo, mit, der in der Lage ist, so gut wie alle Betriebssysteme im Wechselbetrieb zu booten.

Setzt man den LiLo als einzigen Bootmanager ein, sollte er im Master-Boot-Record der Festplatte installiert und für den Start der anderen Betriebssysteme konfiguriert werden.

Soll dagegen ein externer Bootmanager (wie z.B. der OS/2-Bootmanager) zur Auswahl des zu startenden Systems eingesetzt werden, so ist zunächst dieser zu installieren und weiter darauf zu achten, den LiLo zusätzlich im Bootrecord der Linux-Startpartition zu installieren und dem externen Bootmanager diese Partition nach erfolgter Linux-Installation bekanntzumachen.

Anleitungen zur genauen Konfiguration des LiLo und des Bootmanagers entnimmt man der jeweiligen Dokumentation.

2.6. Abfrage der Hardware-Informationen

Die meisten Distributionen fragen nach erfolgtem Kopiervorgang die Hardware des eingesetzten PCs ab. Diese Informationen werden teilweise schon im Plug&Play-Verfahren ermittelt (z.B.

Grafikkartenerkennung bei S.u.S.E.), teilweise muß man jedoch auch selbst nachhelfen. Es empfiehlt sich daher, vor Beginn der Installation Daten über die vorhandene Hardware zusammenzutragen.

Wichtige Informationen sind:

- Typ, Bussystem, Einstellungen, Chipsätze der Steckkarten
- Typ, Größe, Organisation der Festplatte(n)
- Typ des CD-Rom-Laufwerks
- Technische Daten des Monitors (Bildwiederholfrequenz etc.)

Nach erfolgreicher Beantwortung der interaktiven Menüs des Installationsprogramms steht – mit ein bißchen Glück – ein lauffähiges Linux-Basissystem zur Verfügung.

2.7. Wichtige Schritte nach erfolgter Installation

Nachdem das so installierte System zum ersten Mal gestartet worden ist, wird in der Regel nach einem Paßwort für den Benutzer `root` gefragt. Dieser Benutzer ist der Systemadministrator und hat jegliche Berechtigung auf dem PC. Diese potentiell gefährlichen Rechte sollte man durch ein gut gewähltes Paßwort schützen.

Als `root` sollte man sich auch im normalen Betrieb des Rechners nur dann einloggen, wenn es absolut notwendig ist !

So früh wie möglich sollte also nach Abschluß der Konfiguration des Systems ein in seinen Rechten eingeschränkter „normaler“ User zur täglichen Arbeit eingerichtet werden.

Genauer dazu entnimmt man wiederum der Dokumentation der Linux-Distribution oder der in Kapitel 4 folgenden kurzen Befehlsübersicht.

3. Die Konfiguration des Systems

Nach erfolgter Erstinstallation stehen noch nicht alle Fähigkeiten des Linux-Systems zur Verfügung. Folgende Dienste sollten daher auf jeden Fall konfiguriert werden:

3.1. X-Window

Das X-Window-System XFree86 bildet die Grundlage aller grafischen Benutzeroberflächen unter Linux. Auf ihm setzen Desktop-Manager wie der fvwm (klassisch), Gnome (bunt) oder KDE (funktional) auf, die die eigentliche „Optik“ der Linux-Oberfläche bestimmen.

Um eine der grafischen Oberflächen zu nutzen, muß zunächst das X-Window-System an die Grafikkarten- und Monitorfähigkeiten angepaßt werden.

Die meisten Distributionen bringen daher menügeführte Konfigurationsprogramme mit (bei S.u.S.E. heißt das Programm `sax`). Auch hier ist teilweise wieder die Kenntnis der eigenen Hardware gefragt, wenn z.B. die Distribution keine automatische Grafikkartenerkennung vorsieht.

Heutzutage funktioniert die Konfiguration des Grafiksystems allerdings nahezu reibungslos und beschränkt sich meist auf Angaben über Anzahl der Maustasten, Tastaturlayout, Modell des Monitors und die gewünschte Bildschirmauflösung.

Nach erfolgter Konfiguration des X-Window-Systems testet man die korrekte Funktion der gewählten Grafikoberfläche als „normaler“ User mit dem Kommando `startx`. Alternativ kann nun das Linux-System auch so eingerichtet werden, daß beim Systemstart direkt das X-Window-System gestartet wird und alle Logins grafisch erfolgen.

Weitere Informationen hierzu gibt jedes Linux-Handbuch.

3.2. TCP/IP-Netzwerk

Linux basiert zu einem großen Teil auf seinen TCP/IP-Fähigkeiten. Für fast alle Funktionen wird auf irgendeine Weise ein (wenn auch nur virtuelles, auf den eigenen PC beschränktes) Netzwerk vorausgesetzt.

Ist der Rechner mit einer Ethernet-Karte ausgerüstet, sollte man den Hersteller, den Modellnamen, den Typ und die benötigten Einstellungen bereithalten und dem Netzwerkkonfigurationsprogramm der jeweiligen Distribution mitteilen (bei S.u.S.E. `yast`).

Das Programm wird nach einem Rechnernamen und einer Domain fragen. Diese kann man frei wählen (also z.B. Rechnername `erde` und Domainname `kosmos.all`).

Als IP-Adresse wählt man, sofern man keine offizielle Adresse zugewiesen bekommen hat, eine Adresse aus dem Bereich zwischen 192.168.0.1 bis 192.168.255.255. Damit vermeidet man Routingprobleme bei einer eventuellen Internetverbindung.

Fragen nach bestimmten Netzwerkdiensten folgen (je nach Distribution) – Informationen dazu im jeweiligen Handbuch.

Ob die Verbindung zu anderen TCP/IP-fähigen PCs funktioniert, läßt sich einfach überprüfen, indem man folgendes Kommando eingibt:

```
ping <IP-Adresse eines anderen PCs in unserem Netzwerk>
```

`ping` sendet kurze Datenpakete an andere Rechner und „lauscht“ auf der Netzwerkkarte nach Antwortquittierungen der angepingten PCs. Wenn also derartige Rückmeldungen erscheinen, ist alles in Ordnung und funktionsbereit:

```
erde:~ # ping 192.168.102.2
PING 192.168.102.2 (192.168.102.2): 56 data bytes
64 bytes from 192.168.102.2: icmp_seq=0 ttl=128 time=1.095 ms
64 bytes from 192.168.102.2: icmp_seq=1 ttl=128 time=0.942 ms
[...]
```

Nach Abbruch des Kommandos mit STRG-C folgt die Zusammenfassung inkl. einiger Statistik:

```
--- 192.168.102.2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.942/1.050/1.208 ms
```

3.3. Druckereinrichtung

Unter Linux wird jeder Drucker im PostScript-Verfahren angesprochen. Auch Drucker, die diese Seitenbeschreibungssprache nicht von sich aus beherrschen, werden über einen PostScript-Interpreter (`ghostscript`) bedient, der meistens über ein menügeführtes Setup-Tool konfiguriert wird, in dem nur der richtige Drucker auszuwählen ist. Bei S.u.S.E. geschieht dies auch über das Programm `yast`.

Der Test des Drucksystems kann mit dem Kommando `lpr <Textdatei>` durchgeführt werden.

3.4. Kernel neu übersetzen

Linux bietet noch viel mehr Konfigurationsmöglichkeiten.

Um bestimmte Funktionen, wie z.B. Fileserverfunktionen, Soundunterstützung, Unterstützung für weitere Hardware o.ä., zu erhalten, muß oft der Kernel des Linux-Systems mit anderen Optionen neu kompiliert werden. Ob dies der Fall ist, entnimmt man der Dokumentation.

Den Kernel neu zu kompilieren ist nur als Benutzer `root` möglich.

Folgende Kommandos sind zur Neukonfiguration der Kernelquellen nötig:

```
cd /usr/src/linux
make menuconfig
```

Es folgt ein Menü, in dem alle Parameter des Kernels eingestellt werden können (die meisten Punkte bieten einen kleinen Hilfetext an, weiteres wie immer im Handbuch).

Nach erfolgter Konfiguration setzt man folgendes Kommando ab:

```
make dep; clean; zImage; modules; modules_install
```

Nach einiger Wartezeit (und seitenlangen Ausgaben des C-Compilers) steht dann der neue Kernel im Verzeichnis `/usr/src/linux/arch/i386/boot/` unter dem Dateinamen `zImage` bereit.

Diese Datei sollte man wie folgt kopieren:

```
cp zImage /boot/zImage
```

Damit der neue Kernel auch gestartet werden kann, ist noch eine Neukonfiguration des LiLo fällig (mittels `yast` bei S.u.S.E.). LiLo muß mitgeteilt werden, daß ein neuer Kernel verwendet werden soll – am Besten legt man eine neue LiLo-Bootoption an, damit der alte Kernel noch alternativ startbar bleibt, falls mit dem neuen Kernel Probleme auftauchen.

Ob der neue Kernel wie gewünscht funktioniert, läßt sich mit einem kompletten Reboot des PCs feststellen (Bildschirmausgabe verfolgen!).

Hilfestellung zur Kernelcompilation erhält man im Kernel-Compilation.HOWTO, im S.u.S.E.-Hilfesystem und sehr ausführlich im Handbuch.

3.5. Sonstiges

Viele Programme unter Linux funktionieren ohne vorhergehende Einrichtung und Anpassung an die eigenen Bedürfnisse gar nicht oder nur sehr eingeschränkt. Die Vorgehensweise zur Konfiguration spezieller Programme ist sehr unterschiedlich und wird daher immer in entsprechender mitgelieferter Dokumentation beschrieben. Oft lohnt sich ein Blick in das Verzeichnis `/usr/doc/`, um detaillierte Tips zur etlichen Programmpaketen zu finden.

Mit dem Befehl `man <Programmname>` kann in vielen Fällen auch eine entsprechende Anleitung (Manual Page) aufgerufen werden.

Unter S.u.S.E. Linux steht noch der Befehl `hilfe` zur Verfügung, der ein Hypertext-basiertes Hilfesystem mit Suchfunktion aufruft.

4. Umstieg von DOS auf Linux – Kurzübersicht

Beim Umstieg von DOS auf Linux stehen viele Benutzer zunächst relativ hilflos da; nur ein auf den ersten Blick simpler Textprompt blinkt dem Neuling entgegen. Gewohnte Kommandos der DOS-Kommandoebene stehen nicht mehr zur Verfügung, und man weiß noch nicht einmal, wie Hilfe angefordert werden kann.

Erste *Überlebens-tips* für die Kommandozeile:

- Die meisten Linux-Kommandos und -Programme bieten einen Parameter `--help` oder `-h` an, der einen kurzen Hilfetext auf der Konsole ausgibt.
- Reicht diese Kurzhilfe nicht aus, sollte man die Manual Page (Anleitung) des Kommandos lesen; das geschieht über den Befehl `man <Kommando>`.
- Datei-/Verzeichnisnamen und Kommandos lassen sich durch einmaliges Drücken der `<TAB>`-Taste vervollständigen; das erspart Tipparbeit und hilft, wenn man die genaue Schreibweise mal nicht im Kopf hat.
- Eine Liste der für den aktuellen User verfügbaren Kommandos erhält man, indem man am leeren Prompt zweimal die `<TAB>`-Taste drückt.
- Mit `<SHIFT>-<Bild-auf/Bild-ab>` kann man immer nach oben und unten durch die Ausgabemeldungen blättern.

Um den Einstieg in die etwas trockene Welt der Linuxbefehle zu erleichtern, folgt hier eine Kurzübersicht der wichtigsten Befehle zur täglichen Arbeit mit dem Verzeichnissystem und Dateien, zur Systempflege und –administration und eine kurze Liste nützlicher Tools, die einem Linux-User viel Arbeit abnehmen können.

Wichtig: für eine erfolgreiche Anwendung dieser Übersicht wird vorausgesetzt, daß ein lauffähiges Linux-System in einer aktuellen Version (ab Kernel 2.0.35) vorliegt, die `bash` (Bourne-Again-Shell) als Kommandointerpreter eingesetzt wird und bei der Installation ein „normaler“ Benutzer angelegt worden ist.

4.1. Kommandos zum Umgang mit Dateien und Verzeichnissen

Dateien lassen sich unter Linux ähnlich wie unter DOS verwalten. Es können Verzeichnisse angelegt werden, um Daten besser zu organisieren. Allerdings sind einige weitere Punkte zu beachten.

So besitzen unter Linux alle Dateien und Verzeichnisse eine ausgeklügelte Rechtestruktur, um Daten vor unbefugten Usern zu schützen, für bestimmte Usergruppen beschreibbar zu machen oder sie zu verstecken.

Desweiteren wird zwischen Groß- und Kleinschreibung unterschieden, sowohl bei den Dateien im Filesystem, als auch bei jeder anderen Ein-/Ausgabesituation, Benutzernamen, Paßwörtern und bei Kommandonamen.

Im Gegensatz zu DOS benötigen Dateien im Linux-Filesystem keine Endungen wie `.bat`, `.exe` oder `.sys`, um eine bestimmte Funktion zu erfüllen. Solche Funktionen werden z.T. über die Dateirechte geregelt.

Eine ausführbare Datei hat daher unter Linux das `x(eXecute)`-Bit gesetzt (s. `chmod` – Rechtevergabe). Verstecken kann man Dateien über einen vorangestellten Punkt im Dateinamen.

4.1.1. `man` – Manual Pages anzeigen

Der Befehl `man` ist wahrscheinlich einer der wichtigsten überhaupt. Mit ihm lassen sich Hilfetexte zu *allen* installierten Programmen und Kommandos anzeigen. Die Bedienung des Programms `man` ist allerdings nicht auf den ersten Blick erkenntlich.

Scrollen ist mit den Cursortasten, `<Bild-auf>` und `<Bild-ab>`, `<Pos1>` und `<Ende>` möglich.

Suchen innerhalb einer `man` page ist mit `/`, gefolgt vom Suchwort, möglich. Sollte das Suchwort mehrmals im Text vorkommen, so läßt sich mit `n` die nächste Stelle suchen.

Beendet wird das Programm mit `q`.

4.1.2. ls – Verzeichnisinhalt ausgeben

Der Befehl `ls` ist das Äquivalent des DOS-Befehls `dir`.

`ls` ist jedoch um einiges leistungsfähiger, was allein schon die ca. 60 Zeilen lange Liste seiner möglichen Parameter belegt. Die wichtigsten Aufrufarten sind:

`ls -l` (oder `l`)

lange Liste des Verzeichnisinhalts inkl. Dateirechten, User-/Gruppenzugeh., Erstelldatum

`ls -al` (oder `la`)

wie oben, zeigt jedoch auch versteckte Dateien (Dateien, deren Name mit einem `.` beginnt) an

`ls -alR`

rekursive lange Ansicht eines Verzeichnisses inkl. aller enthaltenen Verzeichnisse und versteckten Dateien

4.1.3. cd – Verzeichnis wechseln

Mit `cd` wechselt man in andere Verzeichnisse. Die Syntax ist analog zu `cd` unter DOS, allerdings sollte man beachten, daß das Trennzeichen zwischen den einzelnen Verzeichnissen nicht der Backslash `\` ist, sondern der Slash `/`.

Mit `cd` ohne Verzeichnisangabe wechselt man ins Homeverzeichnis des aktuellen Users. Für den Befehl `cd ..` ist auf den meisten Linux-Systemen das Alias `..` angelegt, d.h. die Eingabe von `..` bewirkt den Sprung ins übergeordnete Verzeichnis.

4.1.4. pwd – Ausgabe des aktuellen Verzeichnisses

`pwd` gibt das Verzeichnis, in dem man sich gerade befindet, aus. Dieses Kommando ist vor allem nützlich, wenn man einmal nicht genau weiß, an welcher Stelle im Dateisystem man sich gerade aufhält.

4.1.5. cp – kopieren

Das Kommando `cp` verhält sich so wie der DOS-Befehl `copy`.

Die Syntax ist `cp [Optionen] <Quelle> <Ziel>`.

Beispiel:

```
cp -R /usr/doc/howto/ .
```

kopiert das Verzeichnis `/usr/doc/howto/` und alle Unterverzeichnisse (Option `-R`) ins aktuelle Verzeichnis `.`

4.1.6. mv – verschieben und umbenennen

`mv` wird zum Verschieben und Umbenennen von Dateien benutzt (`move` und `ren` unter DOS).

Die Syntax ist `mv [Optionen] <Quelle> <Ziel>`.

4.1.7. rm – Dateien löschen

`rm` entspricht dem DOS-Befehl `del`. Die Syntax ist ähnlich:

```
rm [Optionen] <Dateien>
```

Mögliche Optionen sind z.B. `-r` für rekursives Löschen oder `-f` für Force (Löschen erzwingen).

4.1.8. mkdir – Verzeichnisse anlegen

Mit `mkdir` (oder seinem Alias `md`) legt man neue Verzeichnisse an.

Syntax: `mkdir <Verzeichnisname>`

4.1.9. rmdir – Verzeichnisse löschen

Mit `rmdir` (oder seinem Alias `rd`) löscht man bestehende leere Verzeichnisse.

Syntax: `rmdir <Verzeichnisname>`

Komplette Verzeichnisse mit Inhalt löscht man mit `rm -rf <Verzeichnisname>`.

4.1.10. cat – Ausgabe von Dateien

`cat` gibt Dateien aller Art aus, vergleichbar mit `type` unter DOS.

4.1.11. less – komfortable Ausgabe von Textdateien

Gibt ASCII-Dateien aus. `less` erlaubt das Hoch- und Hinunterscrollen mit den Cursortasten sowie die Suche nach bestimmten Textstrings durch Eingabe des Slashes `/`, gefolgt von der Eingabe des zu suchenden Strings. Beendet wird `less` durch `q`.

4.1.12. vi – der Standard-Unix-Texteditor

`vi` ist in gewisser Hinsicht ein Relikt aus vergangenen Tagen der Informationstechnologie. Der `vi` hält sich an keinerlei Bedienungskonventionen, wie sie heutzutage bekannt sind, ist aber trotzdem ungemein wichtig, da er einer der wenigen auf *jedem* Unix-System verfügbaren Editoren ist. Darüberhinaus ist der `vi` natürlich auch extrem leistungsfähig – wenn man die Tastaturkommandos kennt.

Hier einige wichtige Kommandos:

```
vi <DATEI>      öffnet eine Datei zum Editieren.
i              Insert-Modus
R              Replace-Modus
A              Anfüge-Modus am Zeilenende
a              Anfüge-Modus am Punkt
ESC-Taste     Eingabemodi abbrechen
k              Zeichen vor
j              Zeichen zurück
h              Zeile aufwärts
l              Zeile abwärts
J              folgende Zeile an diese anhängen
dd            Zeile löschen
5dd           5 Zeilen löschen
D              Zeile bis zum Ende löschen
x              Zeichen löschen
Y              Zeile in Puffer kopieren
7Y           7 Zeilen in Puffer kopieren
P              Zeile(n) vor Cursor einfügen
p              Zeile(n) nach Cursor einfügen
:w            Datei speichern
:wq           Datei speichern, vi beenden
:q           vi beenden
:q!          vi beenden, ohne zu speichern (wenn mal was schiefgelaufen ist)
$            Ans Zeilenende
^            An den Zeilenanfang
```

usw...

Wer sich wirklich für den `vi` interessiert, sollte das Lernprogramm `vitutor` (gibt's im Internet) mehrere Male in Ruhe durchspielen. Wer mit dem `vi` sicher Dateien bearbeiten kann, hat keine Probleme mehr, sich auf Unix-System aller Art einigermaßen zurechtzufinden :-)

4.1.13. joe – der etwas einfacher zu bedienende Texteditor

joe gehört nicht zum Standardumfang von Linux und muß also bei der Paketauswahl gesondert installiert werden. joe bietet eine eingebaute Hilfefunktion und hält sich grob an die vom alten WordStar-Editor unter DOS bekannten Tastenkombis. Bei Problemen gibt man einfach <STRG>-K-H ein und erhält eine Befehlsübersicht in der oberen Hälfte des Editorfensters.

4.1.14. chmod – Dateisystemrechte vergeben

Auch unter Linux haben Dateien Attribute.

Das Prinzip der Dateiattribute unter Linux ist recht einfach: Jede Datei hat 3 mal 3, also 9 Attribute, die jeweils aktiviert oder deaktiviert sein können, und ein Attribut, das den Typ des Objekts angibt (Verzeichnis oder Datei).

Die Attribute sind r (readable, lesbar), w (writable, schreibbar) und x (executable, ausführbar).

Diese drei Rechte werden sowohl für den Besitzer einer Datei, für seine Benutzergruppe und für alle anderen Benutzer im System vergeben, wodurch es zu folgender Rechtemaske kommt:

Typ	Rechte des Eigentümers			Rechte der Gruppe			Rechte aller anderen User		
d bzw. -	r	w	x	r	w	x	r	w	x

Wenn man sich die Dateien im aktuellen Verzeichnis z.B. mit der Option -l (long) anzeigen läßt, sieht man die Rechtevergabe ganz links:

```
# ls -l
drwxrwxr-x 7 root root 1024 Jan 15 02:46 .
drwxrwxr-x 7 root root 1024 Jan 15 02:46 ..
-rwxrwxr-x 1 root root 7856 Jan 23 13:42 README
```

Danach folgen weitere Informationen über die Datei, wie z.B. Name des Besitzers, die Größe, Datum der letzten Änderung und ganz rechts der Dateiname.

Als Besitzer können Sie die Attribute mit dem Kommando chmod ("change mode") ändern. Rufen Sie chmod mit der folgenden Syntax auf:

```
chmod [Optionen] Modus <Dateien>
```

Optionen: z.B. -R für rekursive Operation (für ganze Verzeichnisbäume)

Der Modus läßt sich als numerischer Wert im oktalen Zahlensystem oder als Kombination von Anweisungen übergeben:

a) numerische Modusangabe; der Modus ist die Summe der nach folgendem Schema berechneten Oktalwerte:

	Rechte des Eigentümers			Rechte der Gruppe			Rechte aller anderen User		
Attribut	r	w	x	r	w	x	r	w	x
Dezimal	2 ⁸	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
Oktal	400	200	100	40	20	10	4	2	1

Beispiel:

```
# ls -l README
-rwxrwxr-x 1 root root 7856 Jan 23 13:42 README
```

Wir legen also die Rechte r(400), w(200) und x(100) für den Eigentümer der Datei fest, addieren deren Oktalwerte aus oben stehender Tabelle (400+200+100=700) und übergeben diesen Wert 700 an das Kommando `chmod`:

```
# chmod 700 README
# ls -l README
-rwx----- 1 root root 7856 Jan 23 13:42 README
```

b) Modusanweisungen; der gewünschte Modus wird durch Kombinationen folgender Syntax ausgedrückt:

```
chmod [ugoa] + bzw. - [rwx] <Dateien>
```

wobei

u für User	+ f. Recht vergeben	r für read (lesen)
g für Gruppe	- f. Recht entziehen	w für write (schreiben)
o für Others		x für execute (ausführen)
a für alle steht.		

Beispiel:

```
# ls -l README
-rwxrwxr-x 1 root root 7856 Jan 23 13:42 README
```

Nun entziehen wir der Gruppe und allen übrigen Benutzern sämtliche Rechte:

```
# chmod go-rwx README
-rwx----- 1 root root 7856 Jan 23 13:42 README
```

Bei Verzeichnissen bedeutet ein gesetztes x-Recht, daß User, Gruppen, Andere dorthin wechseln dürfen. Um allerdings den Inhalt eines Verzeichnisses anzeigen zu können (mit `ls`), braucht man darüber hinaus Leserecht.

4.1.15. `chown` – Eigentümer und Gruppenzugehörigkeit ändern

Mit `chown <neuer Eigentümername>:<neue Gruppenzugehörigkeit>` lassen sich Eigentumsverhältnisse der Dateien und Verzeichnisse angeben.

Soll die Datei `/home/willi/Install.txt` dem Benutzer `joe` aus der Gruppe `users` zugeordnet werden, ist dazu folgendes Kommando nötig:

```
chown joe:users /home/willi/Install.txt
```

4.1.16. `mc` – Midnight Commander

Der Midnight Commander `mc` ist eine nahezu 100%ige Nachbildung des bekannten Norton Commanders für DOS mit einigen Linux-spezifischen Erweiterungen. Mit ihm lassen sich Dateien, Verzeichnisse und Berechtigungen editieren und verwalten.

Wer sich also in der Kommandozeile nicht ganz sicher fühlt, kann trotzdem mit `mc [-c für Farbausgabe]` einfach losarbeiten.

4.1.17. `mtools` – DOS-Disketten benutzen

Sofern im System die `mtools` installiert sind [`rpm -qa | grep mtools` zur Überprüfung], läßt sich sehr einfach auf MS-DOS-formatierte Disketten zugreifen.

Mit `mmdir a:` [oder `b:`] erhält man das Inhaltsverzeichnis der Floppy, mit `mcopy a:* .` wird der Inhalt der Diskette ins aktuelle Verzeichnis kopiert, `mformat a:` formatiert die Diskette, `mdel a:text.doc` löscht die Datei `text.doc` auf der Diskette.

4.2. Kommandos zur Systemverwaltung

4.2.1. User und Gruppen verwalten

Unter Linux wird es oft anfallen, daß der Systemverwalter Accounts für andere User oder neue Benutzergruppen auf seinem System einrichten muß. Linux bietet dazu mehrere Möglichkeiten:

- a) mit einem menügeführten Tool (`yast` bei S.u.S.E.), das eine Eingabemaske für die user- und gruppenspezifischen Daten (Loginname, Paßwort, Homeverzeichnis, Kommandoshell, Gruppenzugehörigkeit, Bemerkungen etc.) bereitstellt. Die Bedienung erklärt sich in diesem Fall meistens selbst.

Im `yast` erreicht man die Userverwaltung über die Menüfolge

Administration des Systems -> Benutzerverwaltung

und die Gruppenverwaltung über

Administration des Systems -> Gruppenverwaltung.

Nähere Angaben zu diesen Programmmodulen erhält man in der Online-Hilfe.

- b) Mit den Kommandos `useradd`, `usermod` und `userdel` lassen sich alle Operationen für die Verwaltung von Useraccounts ausführen, Gruppen lassen sich mit `groupadd`, `groupmod` und `groupdel` anlegen, ändern und löschen. Diese Programme sind relativ einfach zu bedienen, trotzdem sollte man vorher die Hilfetexte mit der Option `--help` lesen, da es zu viele verschiedene Startparameter gibt, um sie hier alle zu erwähnen.

4.2.2. `passwd` – Passwort ändern

Mit der Eingabe von `passwd` ändert der aktuelle User sein Login-Paßwort. Es wird zunächst nach dem alten Paßwort gefragt, dann das neue erbeten und zur Sicherheit ist es noch einmal zu bestätigen.

Der einzige User im System, der Paßwörter anderer User ändern darf, ist der `root`.

`root` kann auch direkt die Paßwörter anderer User ändern, indem er `passwd <Username>` aufruft.

4.2.3. `mount` – Festplattenpartitionen ins Dateisystem einbinden

Festplattenpartitionen werden unter Linux nicht wie unter DOS als Laufwerksbuchstaben zur Verfügung gestellt, sondern einfach irgendwo als ein Verzeichnis in unseren Verzeichnisbaum eingebunden (gemountet).

Es ist also z.B. möglich, zusätzlichen Plattenplatz (etwa eine Partition einer nachträglich eingebauten Festplatte) etwa im Verzeichnis `/home/willi/Projekte/` einzubinden.

Die Syntax des `mount`-Befehls ist wie folgt:

```
mount -t <Typ der Einbindung> <Devicename> <Einbindungsposition im Dateisystem>
```

also z.B. das Mounten einer ISO9660-CD-Rom nach `/cdrom`:

```
mount -t iso9660 /dev/scd0 /cdrom
```

Ein `mount` ohne Parameter gibt eine Liste der eingebundenen Partitionen, Netzlaufwerke etc. aus.

Das Kommando `mount` ist potentiell gefährlich, also sollte man sich vor der Benutzung mit der Syntax und der Festplattenorganisation im eigenen System vertraut machen !

4.2.4. `df` – Festplattenauslastung anzeigen

`df` dient dazu, den verfügbaren und bereits genutzten Festplattenplatz anzuzeigen.

`df -h` (-h für „human readable“) ergibt eine Ausgabe im folgenden Format:

```
neuling@erde: > df -h
Filesystem      Size  Used Avail Capacity Mounted on
/dev/hda3        774M  587M  148M    80%   /
/dev/hdc1       1000M  795M  154M    84%  /disk1
/dev/hda5       1017M  507M  458M    53%  /usr/local
```

4.2.5. kdm – KDE-Windowmanager starten

Standardmäßig wird jedes Linux-System nach erfolgter Erstinstallation im Textmodus starten, da das X-Window-System noch nicht konfiguriert ist.

Möchte man aber schon den Loginvorgang mit der grafischen Oberfläche durchführen, sollte man im Konfigurationsprogramm der jeweiligen Distribution nach einer Einstellung für grafischen Login suchen.

Bei S.u.S.E. ist diese Option im `yast` unter Administration des Systems -> Loginverwaltung zu finden. Dort ist die Option Login auf Grafisch zu setzen, damit beim nächsten Systemboot direkt KDE gestartet wird.

Alternativ kann der Benutzer `root` den `kdm` (KDE Desktop Manager) starten. Dieser läuft dann auf Systemkonsole 7 (erreichbar mittels `<ALT>-<F7>`) und präsentiert wiederum ein Loginmenü.

Am `kdm`-Loginmenü sollte man sich aus Sicherheitsgründen stets als normaler User und NIEMALS als `root` anmelden! Bei Bedarf kann mit dem Kommando `su` - immer noch ein `root`-Account im Terminalfenster geöffnet werden. Diese Vorgehensweise schützt zu einem gewissen Grad vor Hackerangriffen aus dem Internet und nicht zuletzt auch vor eigenen Bedienungsfehlern.

Ein Umschalten in die verschiedenen Textkonsolen ist mit den Tastenkommandos `<STRG>-<ALT>-<F1...F6>` möglich. Zurückschalten in den Grafikmodus geschieht in der Textkonsole mittels `<ALT>-<F7>`.

4.2.6. Alternativer Start des X-Window-Systems

Will man nicht gleich das gesamte Loginwesen über den `kdm` ablaufen lassen, weil man z.B. nur schnell mit Netscape ins Internet möchte, kann man als normaler Benutzer eines der Kommandos `kde`, `startx` oder `xstart` absetzen und befindet sich ohne grafisches Login direkt in seiner gewohnten KDE-Umgebung.

Wird dieses System über „Abmelden“ im KDE-Startmenü verlassen, findet man sich in der Kommandozeile im Textmodus wieder und nicht, wie bei gestartetem `kdm`, wieder im grafischen Loginmenü.

4.2.7. su – Superuser-Kommandos ausführen oder: wie werde ich im Terminal zum root ?

Das Kommando `su` [Option - für echten Login inkl. Sprung ins Homeverzeichnis] dient zur Autorisierung als `root`. Ist es also irgendwann notwendig, `root` zu werden, um bestimmte Kommandos ausführen zu dürfen, gibt man in der Konsole oder im Terminalwindow den Befehl

```
su -
```

ein und nach Rootpaßworteingabe ist man der Benutzer `root`.

Als `root` stehen einem noch andere Optionen des Kommandos `su` zur Verfügung, z.B. die Möglichkeit, die Identität anderer Benutzer anzunehmen:

```
su wwwrun
```

z.B. macht unseren `root` zum Benutzer `wwwrun` (User-ID des Webservers).

4.2.8. ps, kill – Prozeßinformationen anzeigen und Programme gewaltsam beenden

ps gibt die aktuell laufenden Prozesse des Systems aus.

Ohne Parameter gestartet werden nur eigene Prozesse angezeigt, mit der Option axu werden alle laufenden Programme inkl. einer Menge an Informationen ausgegeben:

```
# ps axu
USER      PID %CPU %MEM  SIZE  RSS TTY  STAT  START   TIME  COMMAND
at        108  0.0  0.4   824   300 ?    S     Dec  3   0:00  /usr/sbin/atd
bin       97   0.0  0.5   840   336 ?    S     Dec  3   0:33  /usr/sbin/portmap
cm       14785  0.8  1.7  1772  1080 p0   S     16:37   0:00  bash
cm       14795  0.0  0.5   884   372 p0   R     16:37   0:00  ps axu
root      1    0.0  0.4   820   256 ?    S     Dec  3   0:07  init [2]
[...]
```

Wichtig ist hier vor allem die Prozeß-Nummer (PID), über die sich jedes Programm eindeutig identifizieren und auch beenden läßt.

Da auf einem durchschnittlichen Linux-System jedoch bereits beim Systemstart etliche kryptische Prozesse ausgeführt werden, ist diese Liste unübersichtlich. Mittels einer Umleitung des ps-Outputs in den Textfilter grep (mit dem sog. Pipe-Operator |) lassen sich bestimmte für uns interessante Prozesse finden.

Beispiel:

Netscape ist abgestürzt, jedoch verschwindet sein Fenster nicht von KDE-Desktop.

Wir wollen Netscape daher auf die unsanfte Art beenden.

```
# ps axu | grep netscape
cm       14785  0.8  1.7  7728  1080 ?    S     16:37   0:00  netscape
```

Diese Suche nach dem String „netscape“ liefert uns die betroffene Prozeß-ID, hier also 14785. Dieser Prozeß muß nun nur noch getötet werden (hier mit Parameter -9, für „auf jeden Fall beenden“):

```
# kill -9 14785
```

Fertig. Mit der soeben beschriebenen Methode wird man übrigens auch den xpinguin wieder los :-)

4.2.9. rpm, tar, gzip – Programmpakete installieren

Wer aus dem Internet oder von CDs Software für Linux installieren möchte, steht meistens erst einmal vor dem Problem, wie diese Pakete überhaupt auszupacken sind. Ein unzip – wie unter DOS – existiert in der Regel nicht und führt auch nicht zum gewünschten Erfolg.

Programmpakete unter Linux liegen meistens in einem der beiden Standard-Paketformate .rpm oder .tar.gz bzw. .tgz vor.

.rpm-Pakete sind Archive, die mit dem Redhat Packet Manager erzeugt worden sind, einem mächtigen Hilfsprogramm des amerikanischen Linux-Distributors Red Hat. Um den Bezug zu Windows zu schaffen: rpm ist das Install-Shield für Linux und liegt wegen seines Komforts unter fast allen Linux-Distributionen vor.

.rpm-Pakete lassen sich als root mittels rpm -i <paketname.rpm> installieren und werden automatisch dekomprimiert. Die enthaltenen Programme werden automatisch an die richtigen Stellen im Verzeichnisbaum kopiert und mit den richtigen Rechten versehen.

Derartige Pakete können unter S.u.S.E.-Linux auch über yast installiert werden.

Möchte man feststellen, ob bestimmte .rpm-Pakete schon im System installiert sind, benutzt man das Kommando rpm -qa | grep <paketname.rpm>.

Weitere syntaktische Tips zu rpm enthält seine manual page.

.tar.gz- oder auch .tgz-Pakete sind mit gzip komprimierte tar-Archive. Das hört sich kompliziert an, ist es in der Praxis aber nicht. tar, ehemals ausschließlich für die unkomprimierte Datensicherung

auf Magnetbänder benutzt, bietet die Möglichkeit, seinen Output direkt an `gzip` weiterzuleiten (Option `z`).

`.tar.gz`-Dateien lassen sich mit `tar -tvzf <paketname.tar.gz>` auf Fehler testen und mit der Option `-xvzf` im aktuellen Verzeichnis entpacken.

Enthaltene Dateien sind meistens mit Informationen über den benötigten Verzeichnisbaum eingepackt worden, daher wird beim Auspacken auch die richtige Verzeichnisstruktur erzeugt. Rechte etc. sind meistens auch korrekt gesetzt.

Generell ist zur Installation von fremden Programmpaketen zu bemerken, daß man, falls man vor dem Download die Möglichkeit dazu hat, immer zuerst nach speziell angepaßten Versionen für die eigene Distribution Ausschau halten sollte. So erspart man sich einiges an Handarbeit (Anpassen von bestimmten Pfaden etc.).

Zur Konfiguration der frisch installierten Pakete läßt sich nichts allgemeingültiges sagen, dazu bemüht man am besten die mitgelieferte Dokumentation.

4.2.10. shutdown, halt, reboot – das System herunterfahren oder neu booten

Wie jedes moderne Betriebssystem muß auch Linux vor dem Ausschalten des PCs heruntergefahren werden, z.B. um den Festplatten-Schreibcache korrekt auf die Platte zu schreiben. Wird das System einfach ausgeschaltet, droht massiver Datenverlust.

Um das System nun ordnungsgemäß zu beenden, gibt Linux uns mehrere Möglichkeiten. Der erste Schritt sollte immer die Eingabe des Befehls `sync` sein (als `root`). Dieser Befehl schreibt evtl. noch nicht endgültig gespeicherte Daten, die sich im Schreibcache befinden, auf die Festplatte.

Danach kann man folgendes tun:

- per `<STRG>-<ALT>-<Entf>` in der Textkonsole wird ein Reboot ausgelöst.
- als `root` können die Kommandos `reboot -n` (`-n` für `now`, `jetzt`), `shutdown -h now` oder `halt` eingegeben werden.

Ist das System gestoppt, kann der PC abgeschaltet werden.

5. Weitergehende Informationen

Wer diese Grundlagen der Linux-Benutzung verstanden hat und schon erste eigene Erfahrungen mit dem System sammeln konnte, wird, sofern er sein Linux-System ernsthaft einsetzen möchte, nach Möglichkeiten suchen, sich das Leben mit dem Betriebssystem einfacher zu gestalten. Oder man möchte komplexere Aufgaben angehen, denn dafür eignet sich Linux hervorragend.

Die nachfolgenden Informationen sind nur erste knappe Tips & Tricks, wie bestimmte Dinge unter Linux komfortabel erledigt werden können. Aber wie man weitergehende Hilfestellung unter Linux erhält, sollte man in diesem Stadium ja eh schon längst wissen :-)

5.1. Netzwerkkommandos

Linux bietet extrem gute Voraussetzungen zum Einsatz auf Netzwerk-Workstations. Netzwerkadministratoren in heterogenen Netzwerken wissen die Offenheit des Systems und die mächtigen Diagnose- und Servicetools zu schätzen. Mit den folgenden Befehlen lassen sich bei durchdachtem Einsatz Netzwerkprobleme gut diagnostizieren.

5.1.1. ping

`ping` schickt kleine TCP/IP-Pakete an definierte IP-Adressen und lauscht auf Antworten. Jeder TCP/IP-fähige Computer reflektiert solche Pakete und sendet eine Quittierung an den anfragenden Rechner zurück. Mit `ping` läßt sich überprüfen, ob bestimmte Rechner erreichbar sind oder die Verbindung aus irgendwelchen Gründen gestört ist (Kabel-, Netzkarten-, Routingprobleme)

Die Syntax lautet:

```
ping [optionen] <Rechnername oder IP-Adresse>
```

5.1.2. telnet

Mit `telnet` ist es möglich, sich auf anderen Rechnern einzuloggen, auf denen der `telnetd` (`telnet`-Daemon) läuft. Auf dem entfernten System kann dann ganz normal gearbeitet werden.

Syntax:

```
telnet <Rechnername oder IP-Adresse>
```

5.1.3. finger

Mit dem Befehl `finger` erhält man Informationen über auf entfernten Rechnern eingeloggt User. Damit der `finger`-Befehl funktionieren kann, muß auf dem zu überprüfenden Host ein Programm namens `fingerd` (`finger`-Daemon) im Hintergrund aktiv sein, das auf `finger`-Anfragen reagiert.

Syntax:

```
finger <User>@<Hostname>
```

Läßt man `<User>` weg, erhält man eine Übersicht aller auf dem entfernten Rechner eingeloggt User, ohne `<Hostnamen>` gibt das Programm Informationen über `<User>` auf dem eigenen PC aus.

Mit `finger` läßt sich z.B. auch feststellen, wann ein bestimmter User auf einem bestimmten PC zum letzten Mal Mail gelesen hat (siehe man `finger`).

5.1.4. traceroute

Wie ein Paket im Internet oder in lokalen Netzen den Weg zwischen zwei Punkten zurücklegt, sprich wie es geroutet wird, findet man mit dem Kommando `traceroute` heraus. Besonders nützlich ist dieses Kommando in großen Netzen zur Überprüfung des Routings oder zur Fehleranalyse bei Internetstau.

Auch statistische Daten, wie z.B. die Dauer der einzelnen Weiterleitungen etc., werden ausgegeben.

Die Syntax ist:

```
traceroute [optionen] <Hostname oder IP-Adresse>
```

5.1.5. tcpdump

Mit `tcpdump` kann man den Datenverkehr auf bestimmten Netzwerkinterfaces mitprotokollieren und auswerten. So ist es möglich, zu überprüfen, welche Protokollverhandlungen z.B. auf der Internetleitung ablaufen oder ob bestimmte Rechner von uns Daten empfangen.

Die Syntax lautet:

```
tcpdump -i <Interface>, wobei <Interface> z.B. eth0 für die erste Netzwerkkarte, ppp0 für das erste Modem usw. sein kann.
```

Beispiele:

```
# tcpdump -i eth0 > eth0-protokoll.txt
```

Protokolliert den Verkehr auf der Ethernetkarte in die Datei `eth0-protokoll.txt`.

```
# tcpdump -i isdn0 >> werbetraffic.txt | grep ad.doubleclick.net
```

Protokolliert alle Verbindungen auf der ersten ISDN-Leitung zu `ad.doubleclick.net` in die Datei `webetraffic.txt`.

5.1.6. netcat

Mit `netcat` lassen sich direkte Verbindungen zu bestimmten Ports entfernter Rechner aufbauen. Wer also schon immer mal wissen wollte, wie `sendmail` von anderen Mailservern die Email annimmt, kann z.B. folgendes probieren:

```
netcat mail.ibm.de 25
```

Die genaue Syntax und weitere Parameter entnimmt man der `manual page`.

5.2. Pipes, Umleitungen, Kommandoverkettungen in der bash

Ein paar Beispiele sollen die Möglichkeiten hier einmal ansatzweise vorstellen:

```
ls | grep xyz
```

Die Ausgabe von `ls` wird als Eingabe für den Filter `grep` benutzt. `grep` sucht alle Zeilen heraus, die die Zeichenkette `xyz` beinhalten.

```
ps -aux | grep netscape | cut -c9-16 > nc_proc_nr.txt
```

`ps -aux` gibt alle Prozesse mit allerlei Zusatzinformation aus. Diese Ausgabe wird nach `grep` geleitet. `grep` sucht die Zeilen heraus, in denen `netscape` vorkommt. Diese Zeilen (die aus der Standardausgabe von `grep` kommen) werden dem Filter `cut` mittels einer Pipe in die Standardeingabe geleitet. Dieser Filter lässt nun von jeder Zeile nur die Zeichen von Position 9 bis 16 übrig (das sind bei `ps -aux` zufällig die Prozessnummern) und gibt das Resultat durch die Standardausgabe aus. Das was aus `cut` herauskommt, wird mittels `>` in die Datei `nc_proc_nr.txt` gespeichert.

```
grep $LOGNAME < /etc/passwd | cut -d: -f7
```

Diese Befehle suchen heraus, welche Shell als Loginshell in `/etc/passwd` eingetragen ist.

Mit `<` wird aus der Datei `/etc/passwd` der Inhalt ausgelesen. Dieser Inhalt wird in `grep` geleitet. `grep` sucht nach dem Inhalt der Variable `LOGNAME` und gibt die Zeilen, die passen, durch die Standardausgabe aus. Diese wird nach `cut` umgeleitet. `cut` sucht nun das 7. Feld aus, wobei die Felder mit `:` getrennt werden.

```
sync; sleep 60s; halt
```

Leert den Schreibcache, wartet 60s und fährt das System herunter. Mit `;` als Trennzeichen lassen sich also Befehlsfolgen in einer Zeile eingeben.

5.3. Erweiterte Prozeßverwaltung

Programme oder Kommandos lassen sich auch im Hintergrund starten, wieder in den Vordergrund zurückholen etc.

<i>Befehl</i>	<i>Bedeutung</i>	<i>Beispiel</i>
<Kommando> & STRG-Z	Im Hintergrund laufen lassen Stoppt den Vordergrundprozeß	netscape &
jobs	Listet die Prozesse der bash auf	jobs [1] Running netscape & (wd: ~) [2]- Stopped gimp & [3]+ Stopped html2index
%n	Bezeichnung des Prozesses n	
bg	Bringt einen Prozeß in den Hintergrund	bg %2
fg	Bringt einen Prozeß in den Fordergrund	fg %2

6. ANHANG

6.1. Copyright

Das vorliegende Dokument wurde von Christoph Möller als kursbegleitendes Skript zum Arbeitsamts-Fortbildungskurs "Netzwerkadministrator/-in" erstellt.

Teile des Dokuments sind an öffentlich vorliegende Quellen (z.B. HOWTOs, Doku zu spez. Software, etc.) angelehnt - daher gebührt der Dank erst einmal den Autoren der hervorragenden Linux-Dokumentation!!

Vervielfältigung, Benutzung zu kommerziellen Zwecken und Veröffentlichung in jeglicher Form sind nur mit vorheriger schriftlicher Einwilligung von Christoph Möller gestattet.

Ausdrücklich erlaubt dagegen ist die nichtkommerzielle Nutzung im Rahmen der beruflichen Ausbildung/Studium/Schule.

6.2. Disclaimer

Dieses Dokument dient lediglich als Einführung in die Materie "Linux".

Was mit Eurem Rechner passiert, wenn ihr die hier vorgestellten Techniken anwendet, geschieht einzig und allein auf Eure Gefahr. Der Autor ist in keinem Fall haftbar für eventuelle Schäden an Hard- und/oder Software.