

Crash-Kurs LINUX

basierend auf:

Crash-Kurs "LINUX", Version 1.4.3

entstanden aus den Unterlagen zum Kurs "Einführung in LINUX" des
RRZ/MI der Universität zu Köln

Copyright (1994,1995) Winfried Trümper <truemper@MI.Uni-Koeln.DE>

Überarbeitet 1999 – 2001 Dirk Schneider <d.schneider@wemelug.de>

Nutzungslizenz:

Dieses Skript darf im ganzen oder teilweise vervielfältigt und
vertrieben werden, sowohl in gedruckter als auch in elektronischer
Form, solange die Nutzungslizenz unverändert weitergegeben wird.

Kommerzieller Vertrieb ist in diesem Sinne ausdrücklich erlaubt, solange nur (anteilige) Kosten für den Datenträger (Papier, Disketten, CD-ROM) berechnet werden.

WARNUNG:

Trotz Sorgfalt bei der Zusammenstellung der Kursunterlagen kann ich nicht deren Richtigkeit und Brauchbarkeit garantieren.

Insbesondere kann das Ausprobieren verschiedener Beispiele zu Datenverlust führen oder den Rechner in einen fliegenden Toaster verwandeln. Daher:

BENUTZUNG DES SKRIPTS AUF EIGENE GEFAHR

Inhaltsverzeichnis

- 1 Allgemeines zu LINUX und UNIX
- 2 Dokumentation
- 3 Die Dateiverwaltung von LINUX
- 4 Benutzer und Benutzergruppen
- 5 Bildschirm und Tastatur

6 Die Kommandozeile

7 Drucken

8 Prozesse

9 Netzwerk und Internet

10 X–Windows–System

11 Editoren zur Systemadministration

12 Systemadministration

13 Beschaffung und Installation von Software

Was dieses Skript NICHT leistet

- Beschreibung der Installation von LINUX: die Installation sollte menügeführt und selbsterklärend sein (dennoch ein Tip in 13.3)
- Besprechung von irgendwelchen Anwendungsprogrammen
- eine Bewertung der käuflichen Produkte (CD–ROMs, Bücher)
- ein Leitfaden für Systemverwalter zu sein

1. Allgemeines zu LINUX und UNIX

Glossar:

- mit "Server" werden Maschinen bezeichnet, die anderen Maschinen Daten und Dienstleistungen zur Verfügung stellen
- "POSIX" bezeichnet einen Programmierstandard in der UNIX–Welt
- ein "Betriebssystem" (eng.: "operating system", kurz "OS") ist ein "Programm" wie etwa DOS, WINDOWS oder OS/2

1.1 Was ist LINUX?

LINUX ist ein UNIX-artiges Betriebssystem für den PC, d.h.

- es unterscheidet die Benutzer des Computers ("multi-user")
[anhand ihres Namens und eines (geheimen) Passwortes]
- mehrere Programme können quasi gleichzeitig ausgeführt werden ("multi-tasking")
- es ist kostenlos, zuverlässig, schnell und POSIX-konform
- LINUX wird von einer "Fangemeinde" aus Programmieren entwickelt, die LINUX selbst einsetzen, daher werden Fehler schnell erkannt und behoben; die Entwicklung ist sehr rasant
- LINUX ist in der Programmiersprache C geschrieben
- im allgemeinen Sprachgebrauch wird zwischen dem LINUX-Betriebssystem(-kern) und den LINUX-Distributionen nicht unterschieden (was der Ausgangspunkt für viele Verwirrungen und überflüssige Diskussionen ist; deshalb bei Fragen und Fehlerberichten immer **** BEIDES ***** angeben)
- der LINUX-Betriebssystemkern ("kernel") ist für die Ansteuerung der Hardware und den Betrieb der Programme zuständig
- die sogenannten LINUX-Distributionen sind leicht zu installierende Softwaresammlungen "für den Betriebssystemkern"
- es gibt keine "offizielle" LINUX-Distribution und somit läßt sich von "LINUX" nicht sprechen wie von DOS oder WINDOWS
- technisch gesehen ist LINUX eine gut gelungene Umsetzung von (mehr oder weniger) bekannten Konzepten (und damit veraltet ?)

1.2 Worin unterscheiden sich LINUX-Programme von denen der gängigen Betriebssysteme?

- UNIX-Programme sind sehr zuverlässig
- sie machen nicht bei einer etwas größeren Aufgabe sofort schlapp
- sie besitzen viele Kommandozeilenschalter (zur Konfiguration), lesen ihre Daten vom vorhergehenden Programm und geben es an das folgende Programm weiter (damit lassen sich prima neue Programme entwickeln; aus der Sicht eines "Endbenutzers" aber katastrophal)
- Menüs werden auf die Original-Programme "nachträglich" aufgesetzt – wenn überhaupt !
- Vorgänge lassen sich einfach automatisieren (z.B. die Konvertierung von 1000 jpeg-icons in 1000 gif-icons)
- insbesondere kommen "LINUX-Programme" dem Benutzer meist NICHT entgegen, sondern verlangen eine Auseinandersetzung mit ihren Konzepten (z.B. "TeX"); dies ist für den Durchschnitts-Benutzer von Windows nicht akzeptabel!
- Programme liegen i.d.R im Quelltext vor
- es gibt noch relativ wenig kommerzielle Programme für LINUX

1.3 Stärken und Schwächen von LINUX

- LINUX eignet sich sehr gut zum Programmieren, TeXen, als Server und zum Betrieb der im Internet herumliegenden UNIX-Programme
- durch die hohe Zuverlässigkeit kann man sich auf seine Arbeit konzentrieren und ist nicht mit dem "tuning" des PCs belastet
- LINUX bietet einen relativ sicheren Ausgangspunkt (z.B. nahezu keine Viren) für einen PC, den sich mehrere Benutzer teilen müssen
- Installation, Konfiguration und Betrieb erfordern ein gewisses Maß an UNIX-Kenntnissen
- LINUX ist nicht besser als die herkömmlichen Betriebssysteme für den PC, sondern hat eine andere Zielgruppe, deswegen
 - eignen sich die unter LINUX verfügbaren Programme nicht für jederman (dies wird meist ignoriert)
 - lohnt sich der Aufwand der Installation nicht für jeden Anwender (= Systemverwalter ?)
 - zahlt sich die Einarbeitungszeit nicht in jedem Fall aus

1.4 Welche Hardwarevoraussetzungen benötigt LINUX?

Theoretisches Minimum: 386 SX, 2MB RAM

Akzeptables Minimum ohne GUI: 386 DX mit Coprozessor (!), 8 MB RAM

Akzeptables System ohne GUI: 486/33, 8MB RAM, 500 MB HDD

Akzeptables System mit GUI: Pentium 200 MMX, 64MB RAM, 2 GB HDD

Schnelles System ohne GUI: Pentium 200 MMX, 64MB RAM, 1 GB HDD

Schnelles System mit GUI: Pentium II 500, 128MB RAM, 3 GB HDD

Andere Plattformen: LINUX für Atari, Amiga, MAC, Alpha, Sparc

1.5 Rechtlicher Status

- das Copyright an LINUX liegt bei Linus Torvalds und vielen anderen Programmierern (s. Quelltexte des Betriebssystemkerns)
- die Nutzung von LINUX und vieler Anwendungen unterliegt der GNU Public License:
 - LINUX darf kostenlos weitergegeben werden

- es dürfen bei Verkauf lediglich Kosten für Datenträger berechnet werden (CD-ROM, Disketten, Papier, Verpackung)
- es werden keine Lizenz-, Registrations- oder Benutzungsgebühren fällig
- die Nutzungslizenz muß unverändert an Dritte weitergegeben werden
- LINUX darf geändert werden, aber bei der Weitergabe von geänderten Versionen muß die Originalversion beiliegen
- insbesondere müssen stets die Quelltexte beiliegen

1.6 Verfügbarkeit

- LINUX-Distributionen:
 - "Slackware" ist die Standarddistribution gewesen
 - es existieren "deutsche" Versionen der Slackware
 - "DEBIAN" ist mit Vorkenntnissen einfach zu installieren.
 - „S.u.S.E.“ mit YAST einfach (grafisch) zu installieren und zu konfigurieren
 - wenn irgendwie möglich, sollte man von einer CD, DVD oder via NFS installieren
 - Mittlerweile sehr viele kleine Distributionen, die einfach zu installieren sind
z.B Corel-Linux, Mandrake

1.7 Geschichtliches

- 1991 kauft sich der finnische Student LINUS TORVALDS einen 386er PC (auf Ratenzahlung), um mit den Möglichkeiten des 80386 Prozessors zu experimentieren
- aus den ersten Versuchen wird schnell ein einfaches, aber brauchbares multitasking-Betriebssystem
- zunächst wurde LINUX unter MINIX, einem Lehrbetriebssystem von A. Tannenbaum ("LINUX is obsolete") entwickelt
- später konnte LINUX unter LINUX weiterentwickelt werden
- LINUX wurde von Grund auf neu programmiert, hat also auf der Ebene des Programmcodes keine Gemeinsamkeiten mit MINIX oder anderen Betriebssystemen; auf der Ebene der Funktionalität gibt es wegen der POSIX-Konformität allerdings Gemeinsamkeiten
- die Weiterentwicklung fand (und findet) unter Beteiligung von vielen interessierten Programmierern im Internet statt
- "Updates" heißen "Patches" (-> "Patchwork") und werden offiziell von Linus herausgegeben (bisher einige hundert) SIEHE 12.6

2. Dokumentation

Glossar:

- FAQs: Frequently Asked Questions (with answers !)
- HOWTOs: überarbeitete FAQs
- Guides: überarbeitete HOWTOs
- LDP: LINUX Documentation Project
- "manual": Handbuch

Wichtig für die folgenden Seiten:

- Es wird i.d.R. zwischen Gross- und Kleinschreibung unterschieden.
- Umlaute und Sonderzeichen sollten vermieden werden.
- Befehle zur Systemverwaltung sind meist nur von RUTH ausführbar.

Es kann nicht oft genug betont werden, daß die FAQs und HOWTOs den größten Teil der "Anfängerfragen" beantworten; schliesslich bestehen sie aus "Anfängerfragen" (natürlich samt den Antworten). Wer diese Informationsquellen ignoriert, der macht sich selbst und anderen Menschen unnötige Mehrarbeit. In diesem Sinne: RTFM (SIEHE 9.3.1)

Zur effektiven Stichwortsuche in langen Dokumenten stehen in jedem Editor oder Dateibetrachter ("less", "zless") Suchfunktionen zur Verfügung (folgenden Editoren sieht man diese Funktion nicht an):

- "/" (slash) in "less", "more", "vi"
- Tastenkombination ctrl-s oder strg-s im "emacs"

Außerdem gilt: ist etwas unverständlich oder unvollständig beschrieben, so sollte man versuchen, eine bessere Beschreibung zu entwerfen und diese dem Autor des Originaldokuments zukommen lassen.

2.1 Dokumentation in elektronischer Form

2.1.1 READMEs, FAQs und HOWTOs

- zu finden in den Verzeichnissen `"/usr/doc"`, `"/usr/doc/faq"`, `"/usr/doc/howtos"`, `"/usr/src"`, `"/usr/share/doc"`, `"/usr/share/doc/faq"` und `"/usr/lib"`
- Dateien mit der Endung `".gz"` können mit dem Programm `"zless"` oder durch `"zcat German-HowTo.gz | less"` angezeigt werden
- den Programmtexten der Software liegt Dokumentation bei (in den wenigsten Fällen hat man auf der eigenen Festplatte jedoch Platz für alle Programmtexte)
- Internet:
 - <ftp://ftp.uni-paderborn.de/pub/linux/sunsite/doc>
 - <ftp://ftp.uni-paderborn.de/faq>
- CD-ROMs:
 - Verzeichnisse, die auf den sunsite oder tsx-11 ftp-Server verweisen (z.B. `"sunsite-mirror/doc"` o.ä.), bzw. Verzeichnisse mit Namen `"howto"`

2.1.2 manual-pages:

- `"man-pages"` sind per `man`-Befehl abrufbare Programmbeschreibungen
Bsp.: `"man 1 mtools"` (SIEHE 3.2.3)
- `"man -k"` zur Stichwortsuche in den Titeln der `man-pages`
Bsp.: `"man -k user"`
- am Schluss einer `"man-page"` findet man oft Beispiele und Tips, sowie die Namen der Konfigurationsdateien
- thematische Gliederung der `man-pages`:

- 1: Benutzerprogramme
- 2: Systemaufrufe
- 3: C Bibliotheksfunktionen
- 4: Geräte
- 5: Beschreibung (Format) der Konfigurationsdateien
- 6: Spiele
- 7: Vermischtes
- 8: Systemverwaltung

Insbesondere die Details zu den in diesem Skript mehr beiläufig erwähnten Programmnamen kann/soll man in den `man-pages` nachlesen!

2.1.3 Programmparameter, die eine Kurzbeschreibung anzeigen

- `"-h"`, `"-?"` sind die verbreitetsten Optionen
- `"-v"` oder `"-V"` spucken meist die Versionsnummer aus
Bsp.: `"gcc -v"`
- `"--help"` für GNU-Programme (mit vielen Ausnahmen)
Bsp.: `"ls -help"`
- manche Programme zeigen die Kurzbeschreibung bei Fehlbedienung automatisch an

2.1.4 GNU–TeXinfo

- textbasierte Dokumentation in hypertext–Manier
- nur die X–Windows Version ist für Anfänger geeignet:
"info"–mode des emacs–Editors ("Alt–x info")

2.2 Bücher:

- für Anfänger:
 - LINUX Installation and getting started (Welsh)
 - Running LINUX (Welsh/Kaufmann)
 - LINUX – Installation, Konfiguration, Anwendung (Kofler)
 - LINUX. Unix für jedermann, der PC als Workstation (Uhl)
- Nachschlagewerke, Systemverwaltung:
 - LINUX Anwenderhandbuch und Leitfaden für die Systemverwaltung (Müller/Hetze u.a.)
 - Linux–Companion zur Systemadministration (Hein/Schabacker/Rade/Eichwalder/Matthias)
 - The LINUX Bible (gedruckte HOWTOs)
 - Dr. LINUX
- Netzwerk:
 - Network Administration Guide (Kirch/Uhl)

3. Die Dateiverwaltung von LINUX

Glossar:

- "2" und "4" werden als "to" und "for" gebrauch
- ASCII–Text umfasst die Zeichen 0–9,a–z,A–Z und die wichtigsten Sonderzeichen, aber keine Umlaute und Rahmen
- „/“ ist der „slash“, „\“ der „backslash“
- EOF bedeutet "end of file" und entspricht control–D

3.1 Das Konzept der Dateiverwaltung unter LINUX

3.1.1 Dateien

- Dateien sind die Träger der Information (Texte, Bilder, usw.)
- zulässige Dateinamen:
 - Länge: 1–255 Zeichen
 - Format: kein spez. Format erforderlich (z.B. "DOS 8+3")
 - erlaubte Zeichen: "/" ist kein zulässiges Zeichen für Dateien, die Beschränkung auf a–z,A–Z,0–9,..,_ ist nicht notwendig, aber sinnvoll

3.1.2 Verzeichnisse (engl: "directories")

- Verzeichnisse sind Ordner für Dateien
- zulässige Namen: SIEHE 3.1.1
- Pfade zu Dateien werden mit "/" gebildet, Bsp.: "/home/truemper/devel/kurs/"
- Im Gegensatz zu DOS dient der Backslash unter UNIX zum Entwerten (engl: "Quoten"), d.h. befreit ein nachfolgendes Zeichen von seiner speziellen Bedeutung
Bsp.: "touch Dateiname\ mit\;spez.\Zeichen" (richtig)
"touch Dateiname mit;spez.\Zeichen" (falsch)

3.1.3 Umgang mit Verzeichnissen

- Inhalt anzeigen: "ls"
Bsp.: "ls -l" (list long)
"ls -la" (list long, all; -> Konfigurationsdateien (beginnen mit einem Punkt) werden angezeigt)
- Verzeichnis wechseln: "cd" (change directory)
oder "pushd", "popd" ("cd" mit Merkfunktion)
- löschen: "rmdir" oder: "rm -rf"
(letzteres ohne Nachfrage und rekursiv, d.h. mit Unterverzeichnissen)
- umbenennen, verschieben: "mv"
- erzeugen: "mkdir" oder "mkdirhier"

3.1.4 Handhabung von Dateien

- Ausgabe auf dem Bildschirm: "cat" (concatenate files)
Bsp.: "cat /etc/hosts"
- betrachten: "less", "zless", "more" und "most",
Bsp.: "less /etc/hosts"
- edieren: SIEHE 11.1.3
- kopieren: "cp" (copy) und "rcp" (remote-copy; SIEHE 9.2.2)
Bsp.: "cp /etc/profile .bash_profile"
- Ausgabe auf dem Drucker: "lpr" (line printer)
Bsp.: "lpr /etc/profile"
- berühren: "touch"
Bsp.: "touch Dies_ist_ein_sehr_langer_Dateiname"
- verschieben: "mv" (move)
Bsp.: "mv Dies_ist_ein_sehr_langer_Dateiname DiesID"
- suchen nach: "find"
Bsp.: "find /etc -name 'rc.*'"
- Suche nach Programmen: "type"
Bsp.: "type ls" oder "type pushd"
- suchen in: "grep"
Bsp.: "grep INIT /etc/rc.config"
- Dateityp feststellen: "file"
Bsp.: "file DiesID"
- löschen: "rm" (remove)
Bsp.: "rm -i DiesID .bash_profile"
- vergleichen: "diff"

3.1.5 Gebräuchliche Namenskonventionen für Dateien

- keine vorgeschriebenen Dateiendungen wie unter DOS
("exe" oder "com" für Programme, bzw. "bat" für batch-files)
Es gibt unter verschiedenen WindowManagern eine interne Zuordnung von
Dateiendungen zu bestimmten Programmen, z.B. KDE -> .jpg - kvierer
Typische Dateienden unter Linux sind:

Dateiendung Dateityp

rc Konfigurationsdatei (beginnen i.d.R. mit einem Punkt)

.c C-Programmtext

.cc oder .C C++-Programmtext

.h C-header-Dateien

.o Objekt-Dateien

.f Fortran-Programme

.p Pascal-Programme

.tex TeX oder LaTeX

.dvi DVI-Dateien (TeX)

.tar tape-archive (Sammlung von Dateien)

.shar shell-archive

.gz mit gzip komprimierte Dateien

.tmp temporäre Dateien

3.1.6 Format der Textdateien

- Endung einer Textzeile:
 - UNIX: LF ("linefeed" = Zeilenvorschub, "^J")
 - DOS: CR LF ("carriage return" = Wagenvorlauf, "^M")
 - MAC: CR
 - (Umwandlung mit "fromdos" und "todos" oder automatisch durch "mount" möglich)
- Zeichensätze:
 - UNIX: ASCII + ISO-LATIN1 (f. die Umlaute und Rahmen)
 - DOS: ASCII + IBMPC
 - (Umwandlung mit "dos2unix" und "unix2dos"
 - oder "recode ibmpc:lat1" und "recode lat1:ibmpc")

Vorsicht: ausführbare Programme und Daten in nicht-Text Form werden durch
Formatumwandlungen unbrauchbar!

3.1.7 Betriebssysteminterna

- das proc-Verzeichnis enthält imaginäre Dateien (read- und cat-only) mit wichtigen Informationen über die Hardware, das Betriebssystem und die laufenden Programme
 - Bsp.: "cat /proc/cpuinfo" (Hardware)
 - "cat /proc/version" (Betriebssystem)
 - "cd /proc/1" (Programm "init")

3.1.8 Verweise

- Softlinks: "ln -s" werden zur Laufzeit ausgewertet ,
d.h. beim Zugriff auf den link wird überprüft, ob

- die Datei (bzw. das Verzeichnis), auf die der link zeigt überhaupt existiert
- der Benutzer die notwendigen Rechte besitzt
- Bsp.: "ln -s mail Mail"
(legt einen Verweis "Mail-->mail" an)
- Hardlinks: "ln" sind mit Original IDENTISCH; nur für Dateien möglich; SIEHE 3.1.10

3.1.9 Gerätedateien

- Zugriff auf Hardware erfolgt (theoretisch) nur über Dateien
Bsp.: "cat /dev/mouse" (zeigt Mausbewegungen an; Abbruch mit strg-C)
- diese speziellen Dateien heissen "Gerätedateien" und liegen im Verzeichnis "/dev"
Bsp.:

"/dev/null" (Müllschlucker)

"/dev/mouse", "/dev/ttyS0" – "/dev/ttyS3" (com1–4)

"/dev/modem", "/dev/cua0" – "/dev/cua3" (com1–4)

"/dev/printer", "/dev/lp0" – "/dev/lp2" (lpt1–3)

"/dev/cdrom", "/dev/mcd", "/dev/cdu31a", "/dev/sbpcd", "/dev/scd0"

"/dev/audio", "/dev/dsp", "/dev/mixer"

3.1.10 Zugriffs- und Ausführungsrechte

Erläuterung der Ausgabe von "ls -l"

```
Typ Rechte Hardlinks Eigentümer Gruppe Grösse Datum Zeit Name
d rwxr-x--- 9 truemper studi 1024 Jul 8 13:40 spiele
```

- Typ:
 - file Datei
 - d directory Verzeichnis
 - l link Verweis
- Rechte:
 - kein Recht
 - r read 4 Leserecht
 - w write 2 Schreibrecht
 - x execute 1 Recht zum Ausführen des Programmes (Typ: file)
 - Recht zum Betreten des Verzeichnisses (Typ: directory)

s UID-Bit oder SIEHE 4.3
GID-Bit

- Rechtegruppen:
rwx ??? ??? u Rechte des Eigentümers (user)
??? rwx ??? g Rechte der Gruppe (group)
??? ??? rwx o Rechte aller anderen (others)
- Eigentümer / Gruppe
Name des Eigentümers bzw. der Gruppe ODER UID bzw. GID
- Ändern von Rechten – chmod (change mode; Zugriffsrechte setzen)
Bsp.: "chmod u=rwx,go=rx spiele" (u=user,g=group,o=others)
"chmod 755 spiele" (7=4+2+1, 5=4+1)
– chown (change owner; Eigentümer ändern) (SIEHE 4.1)
– chgrp (change group; Gruppenzugehörigkeit ändern) (SIEHE 4.2)

3.2 Speichermedien

3.2.1 Das "virtual file system (VFS)"

- das VFS ist "das einzige" Dateisystem für LINUX; zumindest erscheint dies dem Benutzer so
- der Zugriff auf tatsächlich existierende Dateisystemtypen (für LINUX, DOS, OS/2) geschieht über Treiber (im Betriebssystemkern)
- tatsächlich nicht vorhandene Eigenschaften einiger Dateisysteme werden durch den Treiber und das VFS vorgegaukelt (z.B. "Eigentümer" und "Zugriffsrechte" für DOS-Dateien)
- Standard-Dateisystemtyp für LINUX-Partitionen ist "extended-2", neu „reiserfs“

3.2.2 Der Verzeichnisbaum von LINUX

- keine Laufwerksbuchstaben, stattdessen ein einziger, aus den verschiedenen Massenspeichern zusammengesetzter Verzeichnisbaum
- FSSTND: File System Standard; Richtlinien zum Aufbau des Dateibaums
- der Befehl "mount" setzt den Verzeichnisbaum zusammen
- Konfiguration des "mount"-Befehls in "/etc/fstab"
- das mount-Kommando kann gewöhnlichen Benutzern in der Konfigurationsdatei erlaubt werden (z.B. sinnvoll für CD-ROMS)
Bsp.: "/dev/cdrom /cdrom iso9660 user,ro,noexec,nosuid"

Beispiel für einen Verzeichnisbaum:

/ Wurzel des Verzeichnisbaums

| (engl. "root-directory")

|

- +-- bin zum Starten benötigte Programme
- +-- boot Kopien der Boot-Konfiguration, Betriebssystemkern
- +-- dev Gerätedateien ("devices")
- +-- etc Konfigurationsdateien ("et cetera")
- +-- home Privatverzeichnisse der Benutzer
- +-- proc Statistiken
- +-- root Privatverzeichnis der Systemverwalterin
- +-- sbin Programme zur Systemverwaltung
- +-- tmp Verzeichnis für temporäre Dateien
- +-- usr nicht zum Booten benötigte Programme,
| | die sich nicht ändern
- | +-- TeX LaTeX/TeX-Satzsystem inkl. Zubehör
- | +-- X11 X Window System, Version 11
- | +-- bin Programme
- | +-- lib fertig übersetzte C-Bibliotheken
- | +-- include Variablendeklarationen f. C-Bibliotheken
- | +-- local nachträglich installierte Software
- | +-- man elektronische Handbücher ("manual pages")
- | +-- src Quelltexte ("sources")
- |
- +-- var veränderliche Dateien ("variable")

3.2.3 Diskettenlaufwerke

- "mtools" ermöglichen einfache Handhabung von DOS-Disketten (Laufwerke a: und b:)
Bsp.: "mdir a:"

- `/dev/fd0` ist das erste Diskettenlaufwerk (autom. Erkennung des Formats),
`/dev/fd1` das zweite
- formatieren von DOS-Disketten mit `fdformat /dev/fd0` ("low-level format")
danach `mformat a:` (fügt DOS-Dateisystem hinzu)

3.2.4 CD-ROM Laufwerke

- Gerätedatei `/dev/cdrom` ist ein link auf den entsprechenden Gerätetreiber (siehe oben)
- Zugriff: `mount -t iso9660 /dev/cdrom /cdrom`
- vor dem Wechseln unbedingt demontieren mit `umount /cdrom`

3.2.5 Festplatten

- die Festplatte wird zur Aufnahme bestimmter Äste des Verzeichnisbaums in Bereiche unterteilt; diese nennt man Partitionen (SIEHE 12.8)
- die Partitionierung der Festplatte wird mit dem Programm `fdisk` (menügeführt: `cdisk`) vorgenommen
- die Partition, die die Wurzel des Verzeichnisbaums `/` enthält, wird als `root-partition` bezeichnet
- `/dev/hda` ist die Gerätedatei der ersten IDE-Festplatte
`/dev/hda1`–`/dev/hda16` sind die Partitionen der ersten IDE-Festplatte
`/dev/hdb` ist zweite IDE-Festplatte
`/dev/sda` ist erste SCSI-Festplatte (usw.)
- Enhanced-IDE Festplatten werden unterstützt, ebenso UDMA
- Festplatten mit mehr als 1024 Zylindern werden unterstützt
(Einschränkung: `root`-Partition unterhalb dieser Grenze bei einigen LILO-Versionen)
- max. Partitionsgröße liegt bei 2 Tera-Byte
max. Dateigröße liegt bisher z.T. noch bei 2 GB, da einige Anwendungen noch kein 64-Bit Dateizugriff unterstützen
- Bsp: `mount -t msdos /dev/hda1 /dos`
`mount -t ext2 /dev/hda2 /var/tmp`

4. Benutzer und Benutzergruppen

4.1 Benutzer ("user")

- * die Daten eines Benutzers sind mit dessen Namen versehen und können so von den Daten anderer Benutzer unterschieden werden
- * vor dem Zugriff auf ein LINUX-System muß sich der Benutzer mit einem Namen (account) und einem Geheimwort (password) ausweisen
- * darüberhinaus bekommt jeder Benutzer zwei Nummern zugewiesen, die "user-identification" (uid) und die "group-id" (gid); programmintern werden stets diese Nummern benutzt, der Benutzer sieht jedoch die zugehörigen Namen (z.B. bei "ls -l")
- * Datenbank der zugelassenen Benutzer: "/etc/passwd" (zugelassene Benutzer -> Benutzer "mit account")
- * Password ändern: "passwd"
- * Shell ändern: "chsh"
- * Namen ändern: "chfn" (den vollen Benutzernamen)
- * die Superuserin RUTH darf ALLES (Benutzernummer 0); insbesondere die Konfiguration des Systems ist RUTH vorbehalten
- * jedem Benutzer ist ein privater Bereich im Verzeichnisbaum zugewiesen, das sogenannte HOME-Directory
- * Benutzerstatistik: "who", "w", "finger", "last -10"
- * Rechner verlassen mit "logout", "logoff" oder "exit"

NUR VON DER SYSTEMVERWALTERIN RUTH AUSFÜHRBAR:

- * "chown" (change owner; Eigentümer einer Datei ändern)
Bsp.: "chown -R mueller /home/mueller"
- * "adduser" oder "useradd" (neue Benutzer anlegen)

4.2 Gruppen

* Gruppen dienen dem gemeinsamen Zugriff mehrerer Benutzer auf dieselben Dateien (bzw. Geräte)

* die Programme "CVS" und "RCS" können (z.B.) konkurrierende Schreibzugriffe (!) auf "Textdateien" verwalten

* jeder Benutzer ist automatisch Mitglied der in der Datei "/etc/passwd" angegebenen Gruppe

* zusätzliche Mitgliedschaften werden in der Datei "/etc/groups" eingerichtet

* Konfiguration der Gruppen: "/etc/groups"

NUR VON MITGLIEDERN DER GRUPPE ODER DER SYSTEMVERWALTERIN

RUTH AUSFÜHRBAR

* "chgrp" (change group; Gruppenzugehörigkeit einer Datei ändern)

NUR VON DER SYSTEMVERWALTERIN RUTH AUSFÜHRBAR:

* "groupadd" (neue Gruppen anlegen)

4.3 Programme mit speziellen Rechten

* um Benutzern einen kontrollierten Zugriff auf Daten zu ermöglichen, kann man unter UNIX Programme mit speziellen Ausführungsrechten versehen

* "suid": "set-user-id-on-execution", statt "x" (SIEHE 3.1.10)
wird ein "s" für den Eigentümer des Programms gesetzt; daraufhin wird das Programm stets im Namen seines Eigentümers und mit seinen Rechten ausgeführt, EGAL welcher Benutzer es startet (vorausgesetzt, er darf es überhaupt starten)

Bsp.: "ls -l /usr/bin/passwd /etc/passwd"

obwohl "/etc/passwd" für die Benutzer nur lesbar ist,
kann jeder Benutzer seine Daten in dieser Datei mit dem
Kommando "passwd" ("suid-root") ändern

* "sgid": "set-user-id-on-execution", statt "x" (SIEHE 3.1.10)
wird ein "s" für die Gruppe des Programms gesetzt; daraufhin
wird das Programm stets im Namen seiner Gruppe und mit
deren Rechten ausgeführt, EGAL welcher Benutzer es startet
(vorausgesetzt, dieser besitzt die notwendigen Rechte)

VORSICHT: diese Arten von Programmen sind potentielle
Sicherheitslücken !

Da sich Skripte ("Stapel-" oder "Batch-Dateien") unter UNIX
ausstricksen lassen (der Inhalt ist kurz nach dem Start vom
Benutzer austauschbar), ignoriert der Betriebssystemkern das
s-bit für Skripte.

Eine Ausnahme ist die Programmiersprache "perl", deren Skripte mit
den speziellen Rechten ausgeführt werden können (relativ sicher).

5. Bildschirm und Tastatur

Glossar:

- * die "Konsole" ist die Kombination aus direkt an die Workstation angeschlossenen Tastatur und Monitor
- * mit "Terminal" wird eine Kombination aus Tastatur und Bildschirm bezeichnet, die nicht direkt (sondern z.B. über ein serielles Kabel) an die Workstation angeschlossen sind

5.1 Konsolen und Terminals (als Ausgabegeräte)

- * Problem: jedes Programm braucht eine Ein- und Ausgabemöglichkeit
 - > multiuser / multitasking erfordert mehr Tastaturen und Monitore, als physikalisch vorhanden sind
 - > Pseudo-Konsolen `"/dev/tty0" - "/dev/tty8"` (und Pseudo-Terminals `"/dev/ttyp0" - "/dev/ttyqf"`)
 - * mit den Tastenkombinationen `"Alt+Ctrl F1" - "Alt+Ctrl F8"` kann man eine der 8 virtuellen Konsolen auf den Monitor blenden
[Bemerkung: auch die 8 virtuellen Konsolen reichen natürlich nicht aus; deshalb gibt es noch die "Pseudo-Terminals". Diese können aber nicht direkt mit der Konsole in Verbindung treten.
SIEHE AUCH "Internet" und "X-Windows-System"]
 - * Anzahl der Konsolen wird in der Datei `"/etc/inittab"` festgelegt
 - * durch den Eintrag `"vga=ask"` in `"/etc/lilo.conf"` und nachfolgendem Kommando `"lilo"` kann man beim nächsten booten die "Auflösung" der Textkonsolen wählen
 - * nützliche Eigenschaften
 - Zurückblättern mit `ctrl-pgup/pgdn` möglich
- (aus Sicherheitsgründen nicht mehr nach Wechsel der Konsole)

- Cut & Paste möglich via "selection" oder "gpm"
- VESA Powersave-Modus wird unterstützt (siehe Datei "/usr/src/linux/drivers/char/vesa_blank.c")
- * der Befehl "setfont" erlaubt die Wahl einer anderen Schriftart
- * das Programm "setterm" setzt Terminalattribute (Hintergrundfarbe, Bildschirmschoner, usw.)
- * die Umgebungsvariable "TERM" (SIEHE 6.4) enthält den Terminaltyp; gängige Werte sind:
 - "vt100": Standardwert ("ich bin ein grüner Textmonitor")
 - "console": die LINUX-Konsole
 - "xterm": Kommandozeile unter X-Windows
- * "tset" initialisiert die Konsole (oder das Terminal) anhand von "\$TERM" (termcap-basiert)
- * zur nachträglichen Manipulation dieser Eigenschaften kann man den Befehl "stty" verwenden:
 - Bsp.: "stty rows 10 cols 40" (10 Zeilen, 40 Spalten)
- * "termcap"-basierte Programme werten "TERM" anhand der (u.U. sehr grossen) Datei "/etc/termcap" aus
- * "terminfo"-basierte Programme werten "TERM" anhand der (alphabetisch geordneten) Terminal-Definitionen im Verzeichnis "/usr/lib/terminfo" aus

5.2 Die Tastatur (auf den Textkonsolen)

Eingabegeräte (Tastatur, Maus) müssen unter UNIX zwischen konkurrierenden Benutzern/Programmen aufgeteilt werden. Aus diesem Grund wird auch hier der direkte Zugriff auf die Hardware vermieden (welch ein Chaos gäbe das !), sondern die Programme erhalten die Eingaben der Benutzer vom Betriebssystem als Steuercode.

5.2.1 Zuordnung Taste -> Steuercode

- * der Befehl "loadkeys" lädt eine Tastaturbelegung aus dem Verzeichnis "/usr/lib/kbd/keytables" und weist anhand der darin enthaltenen Regeln den einzelnen Tasten bestimmte Steuercodes zu
- * evtl. werden diese Steuercodes schon von allen Programmen verstanden (sonst: SIEHE 5.2.2)
- * die Einstellungen durch "loadkeys" betreffen alle virtuellen Konsolen [insbesondere auch das "X-Windows-System"]
- * für deutsche PC-Tastaturen: "de-latin1-nodeadkeys" oder "de"

5.2.2 Zuordnung Steuercode -> Funktion

- * "readline"-basierte Programme werden durch "~/.inputrc" konfiguriert (z.B. die Kommandozeile "bash"); hier sind sehr komplexe Funktionen vorgesehen
- * grundlegende Funktionen ("welchen Steuercode hat BackSpace?") werden mit der Umgebungsvariable "TERM" (SIEHE 5.1) oder mit dem Befehl "stty" festgelegt:
Bsp.: "stty erase ^? susp ^z start ^q stop ^s werase ^w"
[beliebt ist auch "stty erase ^h"]
("erase"="Backspace", "susp"="Programm anhalten",
"stop"="Ausgabe anhalten", "start"="Ausgabe fortsetzen",
"werase"="wortweises Löschen")

5.2.3 Gebräuchliche Abkürzungen für Tastenkombinationen

* gleichzeitiges Drücken mehrerer Tasten ist eine Tastenkombination

* für Tastenkombination kommen in Frage:

"_^-_" (Umschaltung auf Großbuchstaben; engl. "shift")

"Strg" (sprich: "Steuerung"; engl.: "ctrl" wie "control")

"Alt"

"Alt Gr"

in Kombination untereinander und mit jeder anderen Taste

* die Tastenkombination werden abgekürzt durch das Voranstellen

oberer Tastenbezeichnungen; z.B. bedeutet "Strg-c": drücke Taste

"Strg" plus den Buchstaben "c"

* die "Meta"-Taste wird oft von der "Alt"-Taste simuliert; manchmal auch durch "ESC" (diese Taste muß man dann vorausschicken)

* meist sind obige Tasten doppelt vorhanden und u.U. wird zwischen rechter und linker Taste unterschieden (selten)

* ein vorangestelltes Dach "^" oder ein "C-" steht für "Strg"

* "A-" oder "M-" steht für "Alt" bzw. "Meta"

5.2.4 Verbreitete Funktionen einiger Tasten

ctrl-d ("EOF", Eingabe beenden), "q" ("quit", Programm verlassen)

h ("help", Hilfestellung anzeigen)

6. Die Kommandozeile

6.1 Allgemeines zur Kommandozeile

* die Kommandozeile wird als "shell" bezeichnet

* hier speziell: die "bash" ("bourne again shell")

(die "bash" ist POSIX-kompatibel)

* andere shells: csh, tcsh, (pd)ksh, zsh, bsh

* Konfigurationsdateien

"/etc/profile" (globale Konfiguration),

"~/bash_profile" (pers. --) oder

"~/profile" (pers. --)

6.2 Funktion der shell

* Job-Kontrolle

– Hintergrundprozesse:

Programme, die eine Möglichkeit zur (Ein- und) Ausgabe haben,

können mit "&" oder "^Z" und "bg" in den "Hintergrund gelegt

werden" (SIEHE 8.4),

Bsp.: "gunzip vortrag_ps.unix.gz &"

– Übersicht der von einer bash verwalteten Prozesse liefert

der Befehl "jobs"

– Bezugnahme auf jobs der bash mit "%"

Bsp.: "kill %xma" (beendet z.B. "xmaple" und "xmastree")

* Eingabe- und Ausgabeumleitung, Pipelines (Röhren)

– Ausgabeumleitung (von Programm in Datei) mit ">"

Bsp.: "cat /proc/cpuinfo > meineCPU"

– Eingabeumleitung (aus Datei in Programm) mit "<"

Bsp.: "mail truemper < viele_Gruesse"

– Pipelines mit "|"

Bsp.: "cat viele_Gruesse | mail truemper"

* Dateiverwaltung

– der Befehl "umask" setzt die Zugriffsrechte für NEU erstellte Dateien (invertiertes Format)

Bsp.: "umask 077" (nur Eigentümer kann zugreifen)

6.3 Kommandozeile contra Dateimanager: einige Tricks

* die letzten 200 (einstellbar) eingegebenen Befehle werden in der Datei "~/.bash_history" gespeichert und können mit den Cursortasten durchstöbert werden

* gezielte Suche in der "bash-history" mit "^r" möglich

* der Stern "*" ist ein Joker und vervollständigt unvollständige Dateinamen,

Bsp.: "ls -l /dev/lp*"

* die Tabulatortaste "TAB" ergänzt Befehls- und Dateinamen; ist dies nicht EINDEUTIG möglich, so piepst die bash

* bei zweimaligem Drücken von "TAB" werden alle möglichen Alternativen angezeigt

* das erste Wort auf der Kommandozeile wird stets als Programm interpretiert (somit auch als solches vervollständigt), das

zweite und alle folgenden Wörter auf der Kommandozeile werden als Dateinamen interpretiert (bezogen auf das aktuelle Verzeichnis)

* Cursor setzen Anfang: "^a" Ende: "^e" Delete: "^d"

6.4 Umgebungsvariablen

* Umgebungsvariablen beeinflussen sowohl die Kommandozeile selbst, als auch die in ihr gestarteten Programme:

– setzen von Variablen innerhalb der shell durch Zuweisung

Bsp.: "PS1='\h \u \w Kommandozeile:'"

– setzen mit Vererbung an Kindprogramme durch "export"

Bsp.: "export TERM=vt100"

* Löschen mit "unset"

* Bezugnahme auf Variablen mit "\$"

Bsp.: "echo \$PATH"

* wichtige Variablen:

– "TERM": Terminaltyp (z.B. "vt100", "console" oder "xterm")

– "DISPLAY": Internetadresse des Monitors (SIEHE 10.1.1)

– "PATH": Suchpfad für ausführbare Programme

– "HOME": Heimatverzeichnis des Benutzers

– "PRINTER": Name des Standarddruckers (SIEHE)

– "EDITOR" oder "VISUAL": der persönliche Lieblingseditor

6.5 Shell-Skripte

* Dokumentation des Sprachumfangs durch die man–page der bash

* statt eines bash–Skripts sollte man stets ein perl–Skript in

Erwägung ziehen (aber nie für den Boot–Vorgang !)

* debug–Modus wird durch "set –x" (Parameter "–x") eingeschaltet

* werden Skripte mit Parametern aufgerufen, so bezeichnet

– "\$@" alle Parameter

– "\$1" – "\$9" die ersten neun Parameter (weitere mit "shift")

– "\$0" den Programmnamen (ohne Pfadanteil; dies kann

u.U. der Name eines links auf das eigentliche Programm sein)

-----8<----- hier schneiden -----8<-----

```
#!/bin/sh
```

```
echo "Sie riefen das Programm mit folgenden Parametern auf: $@"
```

```
echo –n "Bitte geben Sie etwas ein: "; read $Eingabe
```

```
if [ "$Eingabe" = "w" ]; then
```

```
echo "Sie haben nichts eingegeben"
```

```
else
```

```
echo "Sie haben eingegeben: $Eingabe"
```

```
fi
```

-----8<----- hier schneiden -----8<-----

6.6 Aliase

* Umbenennung von Kommandos (z.B. bei DOS–Entzugserscheinungen)

Bsp.: "alias del='rm –i'

* Abkürzung von Kommandos

Bsp.: "alias wech='clear; logout'"

* Erzeugung komplexer Kommandos

Bsp.: "alias play='cat \$@ > /dev/audio'"

7. Drucken

Glossar:

- * "spoolen" = Zwischenspeichern (der Druckaufträge)
- * "postscript" ist DER Standard unter den Druckersprachen
- * ein "Druckerfilter" bereitet Druckaufträge speziell auf
(z.B. Umwandlung postscript -> Pixeldaten)

7.1 Ansteuerung der Hardware

- * der Betriebssystemkern muß den Druckertreiber enthalten
(sonst kommen Fehlermeldungen wie: "lp0: no such device")
 - * Gerätedateien zum Drucken sind i.d.R. "/dev/lp0" – "/dev/lp2"
(bei seriellen Druckern: "/dev/ttyS0" – "/dev/ttyS3")
- Bsp.: "cat vortrag.txt > /dev/lp1"
- * meist ist "/dev/lp1" der Drucker an der parallelen Schnittstelle
 - * Konfiguration der Gerätedateien mit "tunelp" (tune–lineprinter)
 - * das Drucken mittels der Gerätedateien ist unkomfortabel, s.u.
 - * i.d.R. erwarten Drucker bei einem Text das DOS–Textformat
(SIEHE 3.1.8); bei Texten ist also evtl. eine Konvertierung
des Textes mit "unix2dos" notwendig (oder man stellt den

entsprechenden Mini-Schalter am Drucker entsprechend ein)

* ein postscript-fähiger Drucker ist ein Quell steter Freude

7.2 Der "line-printer daemon"

* Druckaufträge können durch das Programm (genauer: den daemon)

"lpd" verwaltet werden:

– Speicherung der Druckaufträge in den Verzeichnissen

"/var/spool/lpX" (X=0,1,2) und "/var/spool/lpd/DRUCKERNAME"

bis zum Freiwerden des Druckers

– Konvertierung der Daten in die Druckersprache (Pixeldaten)

(klappt nur für LOKALE Drucker! Pixeldaten an fremde Drucker

zu schicken ist ziemlich uncool; z.B. ist dieser Text im

Pixelformat ganze 10 MB groß, als postscript nur 60kB !)

* der "lpd" BENÖTIGT ein Netzwerk (das "loopback" reicht bereits)

* Konfiguration in "/etc/printcap"

* Steuerung mit "lpc"

* Druckauftrag abschicken: "lpr" (lineprinter)

Bsp.: "lpr vortrag.txt"

* die Umgebungsvariable "\$PRINTER" legt den Drucker fest; ist diese

Variable nicht gesetzt, so ist der erste Drucker in der Datei

"/etc/printcap" der Standarddrucker

* die Auswahl des Druckers kann auch mit der Option "-P"

vorgenommen werden (überschreibt andere Einstellungen)

Bsp.: "lpr -Plaser-post vortrag.ps"

* zu grosse Dateien müssen mit der Option "-s" gedruckt werden

Bsp.: "lpr -s vortrag.txt"

Alternative: max. Größe der spool-Dateien erhöhen

* Druckerwarteschlange abfragen mit "lpq"

* Druckauftrag löschen mit "lprm"

7.3 Nützliche Zusatzprogramme

* "ImageMagick" ist ein komfortables Programm zum Betrachten,

Bearbeiten und Konvertieren von Grafiken

* "magicfilter" ist ein Druckerfilter für die Zusammenarbeit mit dem "lpd" (-> lokale Drucker).

* "apsfilter" ist ein weiterer Druckerfilter. Aktuelle

Versionsnummer ist > 4.9; verbreitet ist leider die alte und

leicht fehlerhafte Version 2.3 .

* zur Konvertierung der verschiedenen Grafikformate greifen

die meisten Druckerfilter auf die sog. "pbmplus-utilities" zurück

* "gs": ghostscript ist DAS Programm zur Erzeugung von

Pixeldaten aus postscript (oder zum Anschauen von

postscript-Dateien; s. "man ghostview")

* "mpage", "atp", "gslp", "a2ps", "x2ps", "nenscript": ascii-to-ps

Wandler; sparen Papier indem sie 2 (oder mehr) Seiten Text

verkleinert auf eine Seite drucken (im postscript-Format)

8. Prozesse

8.1 Prozesse

- * jedes Programm erhält beim Start eine eindeutige Nummer im Bereich von 2–32767, die sogenannte "process id", kurz "PID"
- * standardmässig können unter LINUX 512 Prozesse gleichzeitig verwaltet werden
- * im Verzeichnis "/proc" werden Informationen über die Prozesse zur Verfügung gestellt
- * die Programme "ps", "pstree", "top" und "fuser" werten diese Informationen aus
- * mögliche Zustände für Prozesse sind z.B. "running", "sleeping", "zombie", etc.
- * "zombies" sind bereits beendete Programme, deren Daten noch für andere Programme zur Verfügung stehen müssen
- * Programme, die unerlaubt auf Daten anderer Programme zugreifen wollen, werden SOFORT mit der Fehlermeldung "SEGMENTATION FAULT" abgebrochen (SIEHE 8.5)

8.2 Prozeßkontrolle

- * beeinflussen laufender Programme mit durch schicken von Signalen mittels dem Programm "kill"
- * Liste aller Signale mit "kill -l"
- * "kill -KILL" beendet ein Programm unmittelbar (gnadenlos)
- * "kill -TERM \$" ist der Standard und läßt die Chance zum Beenden wichtiger Vorgänge (Speichern, usw.)

Bsp.: "kill \$PPID" (= "logout")

* Signale senden unter Angabe des Programmnamens mit "killall"

Bsp.: "killall -STOP emacs"

"kill -STOP %emacs" (SIEHE 6.2)

* grafische Oberfläche zur Prozeßverwaltung: "tkps"

* textbasierte ----- "" ----- : "top"

8.3 Priorität von Prozessen

* Prozesse können mit unterschiedlicher Dringlichkeit versehen werden:

"-14" ist die höchste Dringlichkeit (wegen der Gefährlichkeit

sind Prioritäten <0 nur von RUTH zu vergeben; VORSICHT !)

"0" bezeichnet die gewöhnliche Dringlichkeit (Standardwert)

"14" läßt einem Programm nur noch die Rechenzeit zukommen, die ihm andere Programme übrig lassen

Dazwischen lassen sich alle Abstufungen wählen

* Starten eines Programms mit Priorität ungleich "0": "nice"

Bsp.: "nice -n 14 xdos"

* nachträgliches herabsetzen (ins Positive) der Priorität: "renice"

* das nachträgliche Erhöhen der Dringlichkeit ist nur der Superuserin RUTH erlaubt

8.4 Hintergrundprozesse und Daemonen

* der Befehl "nohup" leitet die Ausgabe eines Programms in eine

Datei ("nohup.out") um und läßt es mit Priorität 5 starten

Bsp.: "nohup make zImage" (SIEHE 12.6)

* ständig laufende Programme werden als "Daemonen" bezeichnet;

diese schalten sich von selbst in den Hintergrund und

verursachen bis auf evtl. Fehlermeldungen keine Ausgabe auf dem

Bildschirm

* Fehlermeldungen der Daemonen werden i.d.R. in den Dateien

"/var/adm/messages" oder "/var/adm/syslog" mitgeschrieben

* Programmendung "d" weist auf "daemon" hin

* Übersicht einiger häufig eingesetzter Dämonen:

"syslogd" protokolliert Meldungen versch. Programme

"klogd" protokolliert Meldungen des Betriebssystemkerns

"crond" startet Programme zu einstellbaren Zeiten

"inetd" Internet-Super-Daemon (SIEHE 9.4.1)

"xinetd" ist eine flexiblere Version des inetd

"lpd" Drucker-Daemon (SIEHE 7.1)

8.5 Nach einem Programmabsturz ...

* Programmabstürze verursachen i.d.R. einen letzten

Schnappschuß des Programms, der in Form einer sogenannten

core-Datei in das aktuelle Verzeichnis geschrieben wird

* mit dem Kommando "ulimit" kann man die Erzeugung von

core-Dateien unterbinden (bzw. begrenzen)

* core-Dateien können u.U. "wiederbelebt" werden (als

Programme weiterlaufen) oder zur Fehleranalyse mit einem

"debugger" untersucht werden

9. Netzwerk und Internet

9.1 Das Internet

- * Anfänge im ARPANET des amerik. Verteidigungsministeriums
- * Übertragungsprotokoll ist TCP/IP
- * Verantwortung: ??
- * Kosten: ??
- * Folgen: ??

9.2 Nutzung anderer Maschinen

9.2.1 Herkömmliche Programme

- * Terminalsitzungen: "telnet"
Bsp.: "telnet localhost"
- * Datentransfer: "ftp"
Bsp.: "ftp localhost" oder "xftp &"
"ncftp localhost"
- * weltweite Suche in den Katalogen der ftp-Server: "xarchie"
- * veralteter Informationsdienst: "xgopher"

9.2.2 Berkeley r-utilities

* die r-utilities erlauben die Arbeit mit anderen Maschinen ohne die Eingabe eines Namen und Geheimwortes

* "telnet" entspricht "rlogin"

Bsp.: "rlogin localhost"

* Programme auf anderem Rechner ausführen: "rsh"

Bsp.: "rsh localhost ls -l"

* "cp" (copy) entspricht "rcp"

Bsp.: "rcp vortrag.txt localhost:vortrag.text"

* X11-Anwendungen starten: "xon"

Bsp.: "xon localhost"

* Zugriffsregelung via "/etc/hosts.equiv" (globale Erlaubnis an andere Maschinen ohne Angabe bestimmter Benutzer) oder via "~/.rhosts"

* Format von ".rhosts" ist "MASCHINE BENUTZER"; mehrere Einträge sind möglich

Bsp.: "ElFi.MI.Uni-Koeln.DE truemper"

Diese Datei muß für jederman lesbar sein (insbesondere muß das HOME-Verzeichnis betretbar sein)

9.2.3 Moderne Informationsdienste

* Programme wie "NCSA Mosaic" oder "netscape" ermöglichen einen einfachen Zugriff auf alle Informationen im Internet

* die herkömmlichen, etwas umständlichen Programme wie "ftp", "gopher", usw. werden dadurch fast überflüssig

* bunte Grafiken belasten das Internet

9.3 Kommunikation mit anderen Benutzern

9.3.1 Häufig vorkommende Abkürzungen

IMHO: in my humble opinion (meiner bescheidenen Meinung nach)

in my honest opinion (meiner ehrlichen Meinung nach)

<sp?> correct spelling? (die korrekte Schreibweise ist unklar)

ASAP: as soon as possible (möglichst bald)

RSN: real soon now

BTW: by the way (übrigens, nebenbei bemerkt)

FYI: for your information (zur Kenntnisnahme)

ROTFL: rolling on the floor, laughing (die Person wälzt sich vor Lachen am Boden)

RTFM: "read the fucking manual" (ein Hinweis, sich mit dem Handbuch auseinanderzusetzen)

cu see you (wir sehen uns)

g grin (grinsen)

gr&d grinning, running, ducking (grinst, flüchtet, duckt sich)

LOL laughing out loud (lautes Lachen)

MORF male or female (Mann oder Frau?)

OIC oh, I see (ich verstehe)

THX thanks (Danke)

9.3.2 Smilies

:-) Standard-Smiley
;-) zwinkender Smiley
:-/ zweifelnder Smiley
:-(unglücklicher oder verärrerter Smiley
:-(trauriger Smiley
:-0 diesem Smiley steht der Mund vor Schreck offen
(-: Benutzer ist Linkshänder

9.3.3 Palaver in Echtzeit: "talk", "ntalk", "ytalk"

* "ytalk" ist die komfortabelste Version (mit Menü)
Bsp.: "ytalk truemper@EIFi.MI.Uni-Koeln.DE"
* leider sind nicht alle "talk"-Programme zueinander kompatibel
(selbst bei gleicher Bezeichnung)

9.3.4 Elektronische Post

* Elektronische Post heißt "Email" (electronic mail)
* Email-Adressen haben die Form Benutzername@Maschinenname
* "@" wird engl. "at" gesprochen -----^
* einfaches und leistungsfähiges email-Programm: "pine"
* der X.500-Dienst ist ein Adressbuch für Email-Adressen
Bsp.: "finger truemper@X500.Uni-Koeln.DE"
* der Inhalt der Datei "~/signature" wird von vielen Programmen
an die ausgehende Email angehängt
* weiterleiten von email wird von "~/forward" gesteuert; mehrere

Einträge sind möglich, z.B. "truemper@FeilSurf.Bayenbeach.net"

* durch den Eintrag "/usr/local/bin/vacation" wird die mail
beispielsweise an das "vacation"-Programm weitergeleitet
(antwortet automatisch)

9.3.5 Öffentliche Diskussionsforen

* eine "newsgroup" ist die Sammlung der Diskussionsbeiträge
zu einem Thema

* i.a. kann sich jeder an der Diskussion beteiligen, d.h.

Beiträge lesen und verfassen; es gibt aber auch sogenannte

"moderierte newsgroups"

* Programme zur Teilnahme an den "news": "tin", "nn", "emacs"

* für alle die, die es noch nicht gelesen haben: "Emily Postnews",
"Netiquette", "flame-Anleitung" aus der newsgroup "de.newusers"

* LINUX ist aus einer newsgroup heraus entstanden

* einige LINUX-spezifische newsgroups:

"comp.os.linux.announce" (moderiert)

"comp.os.linux.answers" (moderiert)

"de.comp.os.linux"

9.3.6 Spiele

Spiele für UNIX basieren i.d.R. auf einem Netzwerk; auf möglichst
realistische Grafiken wird keinen Wert gelegt. Die beliebtesten
Spiele (potentielle "Promotionskiller") sind:

- * "mud": textbasiertes Rollenspiel, meist im Stil von "Dungeons&Dragons", Welten werden von erfahrenen Spielern gestaltet; Informationen in den newsgroups "rec.games.mud.announce" und "de.alt.mud"
- * "xpilot": Raumschiff-Kriegsspiel in 2D-Welt, Team-Modus möglich; Information via "telnet xpilot.cs.uit.no 4400"
- * Schach: "xboard" (Oberfläche f. "gnuchess"); Kontaktaufnahme zum Schach-Server mit "xboard -ics True -icshost chess.lm.com"

9.4 Serverfunktionen (Netzwerkdienste aus Sicht des Systemverwalters)

9.4.1 Der Internet-Super-Daemon "inetd"

- * der Daemon "inetd" startet bei Anforderung von Netzwerkdiensten die speziell notwendigen Daemonen
- Bsp.: "telnet"-Anforderung startet "/usr/sbin/in.telnetd"
- * Konfiguration durch "/etc/inetd.conf" und "/etc/services"
- * die meisten Dienste sind vorkonfiguriert
- * häufig in Anspruch genommene Dienste sollten als eigenständige Prozesse laufen; z.B. "sendmail" und "httpd"
- * "xinetd" ist eine erweiterte Version des "inetd"

9.4.2 Dateien im Netz gemeinsam nutzen

- * das "Network FileSystem (NFS)" erlaubt das mounten fremder

Massenspeicher (z.B. Festplatten)

Bsp.: "mount -t nfs rs1:/afs/rrz/pub/packages/linux /mnt"

* sinnvoll z.B. für das HOME-Directory

* FSSTND erleichtert effektive Nutzung von NFS

* Anzeige von nutzbaren Verzeichnissen: "showmount"

Bsp.: "showmount -e rs1.rrz.uni-koeln.de"

* überlegenswerte Alternative zu NFS: "rdist"

9.4.3 Vermischtes

* Drucken: der lpd-Daemon ist standardmässig netzwerkfähig

* SMB: das Packet "samba" ermöglicht den Zugriff auf Dateien und

Drucker des LINUX-Systems von jedem anderen Betriebssystem

aus

* X11: netzwerkbasierte, grafische Oberfläche (SIEHE 10.1)

9.5 Die Logistik des Internet

9.5.1 Adressen und Netze

* jede Netzwerkkomponente hat (mind.) zwei Adressen:

– die IP-Adresse

Form: "aaa.bbb.ccc.ddd", nur Zahlen von 0-255

Bsp.: "134.95.201.169"

– den sybolische Maschinenname ("hostname")

Form: Namen mit Buchstaben und/oder Zahlen

Bsp.: "annexR1.rrz.Uni-Koeln.DE"

* sogenannte "nameserver" kennen die zu den Maschinennamen gehörenden IP-Adressen (und umgekehrt)

* DNS ist die Abkürzung für das "domain name system" und bezeichnet das System der nameserver

* die eigene Maschine hat u.a. immer die Adresse "127.0.0.1" und den Namen "localhost" (Gerät: "loopback-device")

* die "0" ist als Adresse für ein Netz reserviert ("Vorwahl")

Bsp.: "134.95.213.0" ist "mi-ether-0.MI.Uni-Koeln.DE"

"134.95.0.0" ist "Uni-Koeln.DE"

* die "255" ist für Nachrichten "an alle" reserviert

Bsp.: "134.95.213.255" ist "an alle im 213er Netz"

"134.95.255.255" ist "an alle in der Uni-Koeln"

* folgende Adreßbereiche sind zum Aufbau privater Netzwerke freigegeben (also nicht reservierbar):

10.bbb.ccc.ddd

172.16.ccc.ddd

192.168.ccc.ddd

* man spricht von Class-A, -B, -C Netzwerken

Bsp.: "134.95.213.0" ist ein Class-C Netz

* Sub-Netze

Bsp.: "134.95.213.0" ist Sub-Netz von "134.95.0.0"

* auf Hardware-Adressen wird hier nicht eingegangen

9.5.2 Zustellung von Datenpaketen

Das Verschicken von Datenpaketen funktioniert im Prinzip wie das Verschicken eines Packets per Post: Empfänger und Postleitzahl draufschreiben und zur Post bringen. Die Postleitzahl (IP-Adresse) des Empfängers (Maschinename) kann von sogenannten "nameservern" erfragt werden. Die Funktion des Zustellers übernehmen die sogenannten "gateways" oder "router".

Man muss dem LINUX-System im wesentlichen nur sagen, welcher nameserver und welcher router zuständig sind, der Rest funktioniert dann automatisch. Beispiel für "Ethernet" (s.u.):

```
Destination Gateway Genmask Use Iface
```

```
loopback * 255.0.0.0 1084523 lo
localnet * 255.255.255.0 5661788 eth0
default 134.95.213.254 * 4187638 eth0
```

Der erste Eintrag ist ein Standard-Eintrag und schickt alle Daten für "127.*.*.*" an das LINUX-interne Pseudo-Netz ("loopback"):

```
/sbin/route add -net 127.0.0.0
```

Der zweite Eintrag (eine Feineinstellung) wurde mit dem Befehl `/sbin/route add -net localnet netmask 255.255.255.0` erzeugt und sorgt dafür, daß Pakete an direkte Nachbarn im Netz nicht den "Umweg" über den Router ("Post") nehmen.

Wenn die Daten nicht lokal zugestellt werden können, schlägt der dritte Eintrag ("default") zu und überläßt die Zustellung dem router:

```
/sbin/route add default gw 134.95.213.254 metric 1
```

Vor der Festlegung von Wegen für den Datenverkehr müssen ("routing") die Schnittstellen zum Netz ("network-interfaces") eingeschaltet und mit einer IP-Adresse versehen werden:

```
/sbin/ifconfig lo 127.0.0.1
```

[...]

```
/sbin/ifconfig eth0 134.95.213.175 [...]
```

9.6 Netzwerkkonfiguration

9.6.1 Grundlegende Einstellungen

* notwendige Informationen:

– IP-ADDRESS: numerische Adresse der eigenen Maschine

– HOSTNAME: symbolischer Name der eigenen Maschine

– NETMASK: 255.255.255.0

– BROADCAST-ADDRESS: Daten, die alle angehen

(eigene IP-Adresse mit 255 als vierter Zahl)

– NETWORK: Adresse des lokalen NETZES

(eigene IP-Adresse mit 0 als vierter Zahl)

– GATEWAY: Adresse des lokalen Gateways/Routers

– NAMESERVER: numerische (!) Adresse des nächsten Nameservers

* jede gute Distribution wird die obigen Daten bei der Installation

erfragen, in die entsprechenden Dateien schreiben und dafür

sorgen, daß die nachfolgend beschriebenen Befehle diese Daten

auswerten

* der Befehl "ifconfig" weist der eigenen Maschine IP-Adressen zu
(genauer: den Netzwerk-Komponenten; u.U. mehrere !)

Bsp.: "ifconfig" (zeigt die momentanen Einstellungen an)

* der Befehl "route" legt die Wege fest, auf denen Datenpakete
verschickt werden sollen

Bsp.: "route" (zeigt die momentanen Einstellungen an)

* bei der Verwendung eines Modems wird man die Einstellungen
z.B. dem Programm "dip" mitteilen, das ebenfalls die oben
genannten Befehle aufruft

9.6.2 Feineinstellungen

* die wichtigsten IP-Adressen können in die Datei "/etc/hosts"
eintragen werden

Bsp.:

-----8<----- hier schneiden -----8<-----

127.0.0.1 localhost

134.95.201.227 a2288354.slip.Uni-Koeln.DE a2288354

134.95.221.169 annexR1.rrz.Uni-Koeln.DE annexR1

134.95.19.4 smail.rrz.Uni-Koeln.DE smail

-----8<----- hier schneiden -----8<-----

Vorteile:

- bei Ausfall des nameservers kann mit Einschränkungen
weitergearbeitet werden

– bei Anbindung via Modem müssen die Informationen nicht ständig über die langsame Telefonleitung angefordert werden (allerdings benutzen einige Programme stets das DNS, d.h. umgehen die Datei "/etc/hosts"; Abhilfe: "nameserver=localhost" in "/etc/resolv.conf" eintragen und "named" konfigurieren)

* weltweite Gültigkeit haben nur die Auskünfte der Nameserver (!)

* Abfrage der nameserver mit "nslookup" (keine man–page, dafür ein help–Befehl auf dem nslookup–prompt)

* in "/etc/resolv.conf" kann ein Suchpfad für symbolische Namen angegeben werden

Bsp.:

```
-----8<----- hier schneiden -----8<-----  
search MI.Uni–Koeln.DE Informatik.Uni–Koeln.DE Uni–Koeln.DE  
nameserver 134.95.100.208  
nameserver 134.95.100.209  
-----8<----- hier schneiden -----8<-----
```

9.6.3 Programme zum Test des Netzes

* Netz–Radar: "ping"

Bsp.: "ping nic.funet.fi"

* Mitschnitt des Datenverkehrs: "tcpdump"

Bsp.: "tcpdump"

* welcher Weg führt nach Rom: "traceroute", "tkined"

Bsp.: "traceroute nic.funet.fi"

9.6.4 Zugriffsregelungen

- * Konfiguration der über den Super-Daemon "(x)inet" zur Verfügung gestellten Dienste: "/etc/(x)inetd.conf"
- * ftp: "/etc/ftpaccess"
- * NFS: "/etc/exports"
- * rlogin: "/etc/hosts.equiv"
- * Drucker: "/etc/hosts.lpd"

9.7 Möglichkeiten zur Realisierung eines Netzes

- * loopback (Pseudo-Hardware)

Interface: "lo" (loopback-device)

Reichweite: 0 m Transferrate: 0.2 – 1 MB/s

Teilnehmer: 1 (verbindet den Rechner nur mit sich selbst)

Protokoll: "TCP/IP"

Kabel: keins, ein reiner Software-Trick

Kosten: entfallen

- * serielle Schnittstellen, bzw. Modem ("COM1–COM4")

Interface: "sl0" bis "sl3" (erweiterbar)

Reichweite: max. 120 m (ohne Modem)

Transferrate: 0.3 – 11kB/s (typisch: 1.5kB mit 14.400 Baud)

Teilnehmer: 2 oder mehr (setzt Server am anderen Ende voraus)

Protokoll: "(C)SLIP": (compressed) serial line TCP/IP

Kabel: Nullmodem, bzw. Telefon

Kosten: ca. DM 20 (5m Kabel); ca. DM 140 für ein Modem

* ISDN

Interface: "isnd0" – "isdn15" (z.B.)

Transferrate: 8kB/s und mehr

Teilnehmer: 16 und mehr

Protokoll: X75 (LAPB), HDLC, CISCO

Verbindung: Telefon (ISDN–Anschluß)

Kosten: ca. DM 300 (plus laufende Gebühren)

* Ethernet (Einsteckkarte)

Interface: "eth0", "eth1", ...

Reichweite: 300 m Transferrate: max 1 MB/s

Teilnehmer: 253 (theoretisch und ohne zusätzliche Hardware)

Protokoll: TCP/IP (u.a. "IPX", "X.25", "appletalk")

Kabel: Koax–Kabel, T–Stücke (NOTWENDIG !) und

Endwiderstände (NOTWENDIG !)

Kosten: ab DM 70 (Ethernet–Karte) + DM 40 (Kabel+Zubehör)

* Token–Ring

Interface: "tr0" bis "tr2"

Reichweite: 300 m Transferrate: 0.5 MB/s oder 2 MB/s

Teilnehmer: 8 (theoretisch und ohne zusätzliche Hardware)

Protokoll: TCP/IP

Kabel: spez. Token–Ring Kabel

Kosten: >> DM 400

* parallele, bzw. Druckerschnittstelle (nicht attraktiv)

9.8 Sicherheitsaspekte

* Daten werden i.d.R. im Klartext durch das Internet geschickt;

insbesondere also die passwords; aber dies ist nur das

einfachste Beispiel für die Angriffspunkte; also Vorsicht !

* Verschlüsselung von Daten: "pgp" (pretty good privacy)

* "CERT": Computer Emergency Response Team (nicht LINUX-spezifisch)

ftp://info.cert.org/pub/cert_advisories

newsgroup "comp.security.announce"

* mailing-Liste "Majordomo@linux.nrao.edu" bietet LINUX-spezifische

Diskussionen und Ergebnisse ("linux-security-digest"), sowie

Hinweise auf dringliche Sicherheitslücken ("linux-alert-digest")

* Kerberos / Project Athena:

durch nicht öffentlich zugängliche Hardware und spezielle

Schlüssel-Techniken wird ein hohes Maß an Sicherheit erreicht

(SIEHE Spektrum der Wissenschaft 1/95)

10. X-Windows-System

Glossar:

- "X" und "X11" sind Abkürzungen für das "X-Windows-System 11"
- ein "X-Terminal" ist ein Monitor (ohne irgendwelche Laufwerke), der (z.B.) über Ethernet mit einer Workstation verbunden ist und von dieser X11-Kommandos erhält

10.1 X11

10.1.1 Grundlegende Eigenschaften von X11

- * X11 ist ein Netzwerkprotokoll für grafische Informationen
- * das Programm "X" heißt "X-Server" und setzt das Netzwerkprotokoll in ein Bild auf dem Monitor um; dazu bedient sich das "X" einer freien virtuellen Konsole (ohne login-prompt)
- * X für LINUX heißt offiziell "XFree86"; aktuelle Version ist "3.1.1" und entspricht dem "Release 6" (X11R6)
- * der Monitor erhält eine Netzwerkadresse; dazu wird an den Maschinennamen ein ":0.0" angehängt
Bsp.: "localhost:0.0" oder "nic.funet.fi:0.0"
- * anhand der Umgebungsvariable "DISPLAY" wird Programmen mitgeteilt, welchen "Monitor" sie benutzen sollen
- * der Zugriff auf die eigene X-Sitzung kann mit "xhost" entweder allen (anderen) Benutzern einer (anderen) Maschine erlaubt werden (unsicher), oder
- * vor dem Zugriff auf die X-Sitzung muß sich ein Programm mit einem (für diese Sitzung) speziellen Schlüssel ("MIT-Magic-Cookie" aus der Datei "~/.Xauthority") ausweisen; dieser kann z.B. mit "xauth" weitergegeben werden (siehe "xon"

oder "xrsh")

* wird "X" mehrmals gestartet (unter LINUX möglich), so erhalten die daraus entstehenden Oberflächen anstelle der Endung ":0.0" die Endungen ":1.0", ":2.0" usw.

* Wechseln zwischen Konsolen (auch Textkonsolen) mit der Tastenkombination Alt-Strg-Funktionstaste (1-8)

* unsanftes Beenden (Notbremse) der Sitzung mit Alt-Strg-BackSpace

10.1.2 Konfiguration von XFree86, Version 3.1.1

* Grafikkarte:

um die Hardware möglichst optimal auszunutzen, existieren für die verschiedenen Familien von Grafikkarten (z.B. "S3", "Mach") spezielle Versionen des XFree86

* Monitor:

trotzdem ist noch eine Konfiguration notwendig, da der Monitor ebenfalls optimal ausgenutzt werden soll, bzw. nicht überlastet werden darf; der Distribution sollte ein einfaches Programm zur Erzeugung der Konfigurationsdatei beiliegen; z.B. "xf86config"

* Tastatur:

XFree86 übernimmt beim Start die Tastenbelegung der Textkonsolen ... mit Ausnahme der Taste "Alt Gr"; folgender Eintrag in der Konfigurationsdatei gibt diese frei

-----8<----- hier schneiden -----8<-----

LeftAlt Meta

RightAlt ModeShift

ScrollLock ModeLock

-----8<----- hier schneiden -----8<-----

10.1.4 Der Startvorgang

* "startx" startet XFree86 einmalig und ohne login-Aufforderung;

Konfiguration im Verzeichnis "/etc/X11/xinit", Datei

"xinitrc" (sowie "Xmodmap" und "Xresources")

* "xdm", der X-Display Manager sorgt für einen ständigen Betrieb
des XFree86 auf der Konsole; Konfiguration des xdm im Verzeichnis

"/etc/X11/xdm", Datei "Xsession" (u.a.)

Bsp.: "xdm"

* den "xdm" anderer Maschinen ("localhost" ersetzen) benutzen

"X -indirect localhost" ("chooser")

"X -query localhost"

10.2 Windowmanager

* das reine "X" stellt die grafischen Informationen lediglich dar,

verwaltet die Fenster aber NICHT im eigentlichen Sinne

* die Verwaltung der Fenster, deren Dekoration und Aussehen wird
vom Windowmanager (Fenstermanager) gesteuert

* sogenannte "virtual" Fenstermanager vergrößern die nutzbare
Oberfläche; Bildschirmgröße legt max. sichtbaren Ausschnitt fest

10.2.1 Übersicht Windowmangager

- * "fvwm" (Free Virtual Window Manager) ist unter LINUX das Standardmodell: klein, schnell und vielseitig (mwm-Emulation, twm-Emulation), Konfigurationsdateien "~/.fvwmrc" (privat) und "/etc/X11/fvwm/system.fvwmrc" (global)
- * "mwm" (Motif Window Manager) kommerzielles Produkt, deutlich größer und langsamer als "fvwm"
- * "ol(v)wm" (Open Look (Virtual) Window Manager) Auslaufmodell von SUN, Konfiguration unter "~/.openwin-init" oder "/usr/openwin/lib" (global)
- * "(c)twm" gehört zum Standard-Lieferumfang von X11; schlichte Standard-Konfiguration, schlichtes Aussehen (BAH !)

10.2.2 Herumspielen mit Windowmanagern

- * "X &" starten
- * mit "ALT-STRG-F1" ("F1"- "F10") auf die Textkonsole zurückwechseln
- * "export DISPLAY=localhost:0.0" zum Setzen der Monitoradresse
- * "xterm &" startet Kommandozeilen-Fenster (-> Konsole wechseln)
- * ohne Rahmen unbefriedigend: Windowmanager (s.o.) starten (ohne &)
- * Windowmanager mit "strg-c" beenden und anderen starten (falls vorhanden)

10.3 Konfigurationsmöglichkeiten

* die Konfiguration bezieht sich stets auf eine Monitoradresse,
ist also NICHT hardwareabhängig

* Tastatur: "xmodmap", bzw. "xkeycaps"; Konfigurationsdateien
".Xmodmap" (privat) und "/etc/X11/Xmodmap" (?)

Bsp.: "xmodmap .Xmodmap"

* allgemeine Eigenschaften: "xset"

* allgemeine Informationen über X anzeigen: "xdpyinfo"

* Farbnamen anzeigen: "showrgb"

* Zeichensätze auflisten: "xlsfonts", bzw. "xfontsel"

X11–Anwendungen lassen sich über die Kommandozeile und über die
sogannanten "X–Ressourcen" konfigurieren.

Die Standardkonfiguration der X11–Programme findet man in
"/usr/lib/X11/app–defaults", diese sollten nicht geändert werden,
sondern nur als Beispiel dienen.

Stattdessen kann man die Einstellungen mit Einträgen in den Dateien

".Xresources" oder ".Xdefaults" (privat)

"/usr/X11R6/lib/X11/xinit/Xresources" (global)

überschreiben.

Bsp.:

-----8<----- hier schneiden -----8<-----

```
*Background: lightgrey
xwpe.font: 10x20
Mosaic*HomeDocument: http://www.rrz.uni-koeln.de/
Emacs*cursorColor: red
XDvi.paper: a4
XDvi*geometry: 800x720
XTerm*ttyModes: intr ^c quit ^\ kill ^u erase ^?
XTerm.VT100.Translations: #override \n\
<Key>Home: string(0x01) \n\
<Key>End: string(0x05) \n\
<Key>Delete: string(0x04) \n\
<Key>BackSpace: string(0x7f) \n\
Ctrl<Key>Left: string(0x1b) string("b")\n\
Ctrl<Key>Right: string(0x1b) string("f")\n
-----8<----- hier schneiden -----8<-----
```

Die Konfiguration werden geladen mit dem Befehl "xrdb"

Bsp.: "xrdb -merge .Xresources"

11. Editoren zu Systemadministration

11.1 "ee"

* der "easy editor" ist sehr klein (80kb) und mit einem bescheidenen Menü versehen

- * zum Editieren von kleinen Dateien und für Anfänger zu empfehlen

11.2 "emacs"

- * DER Editor schlechthin, aber KEINE Textverarbeitung
- * automatische Erkennung des Textes (C, TeX, postscript)
- * Akronym: eight megabytes and constantly swapping
- * Konfigurationsdateien ".emacs" (privat) und
"/usr/lib/emacs/site-lisp/site-start.el" (global)

11.3 "vi" (benutzerunfreundlich, aber leider Standard)

- * Unterscheidung zwischen Eingabe- und Kommandomodus
- * Einschalten des Eingabemodus mit "i" (insert) oder "a" (append)
- * Einschalten des Kommandomodus mit "ESC"
- * Verlassen des "vi" mit Speichern: ESC zz (thanks, Bert)

12. Systemadministration

12.1 Daten über den Zustand des Systems

- * Auslastung des Rechners: "top" ("xmem", "xload", "xidle")
- * verfügbarer Festplattenplatz: "df" (disk free)
- * freier Speicherplatz (RAM): "free"

12.2 Logbücher

* unter LINUX wird zum Protokollieren von Meldungen

der Daemon "syslogd" eingesetzt

* Konfiguration des "syslogd" unter "/etc/syslogd.conf";

Standard ist:

– gewöhnliche Meldungen: Datei "/var/adm/messages"

– Systemmeldungen: Datei "/var/adm/syslog"

* Meldungen des Betriebssystemkerns werden durch den Daemon

"klogd" ("kernel-log-daemon") verwaltet; diese sind abrufbar

mit dem Befehl "dmesg"

12.3 Datensicherung

- "tar -cMf /dev/fd0 ..."
- „tar -cfv /dev/st0 ...“
- Programm "kbackup"

12.4 Überwindung der Speicherknappheit

Da das physikalisch vorhandene RAM zum gleichzeitigen Betrieb

vieler Programme selten ausreicht, bietet LINUX die Möglichkeit,

Bereiche der Festplatte wie RAM zu behandeln. Diese Bereiche

(Partitionen oder Dateien) nennt man SWAP (-Geräte).

* SWAP ist sehr viel langsamer als RAM (Faktor 100)

* Erzeugen von SWAP-Geräten mit "mkswap"

- * Einschalten von SWAP-Geräten mit "swapon"
- * SWAP-Geräte können in gewissen Grenzen im laufenden Betrieb ein- und ausgeschaltet werden

12.5 Abschalten des Computers

- * der Computer darf nicht einfach abgeschaltet werden
- * stattdessen muß für ein geregeltes Beenden aller Programme; das Sichern aller Daten auf Platte und für eine Demontage des Dateibaums gesorgt werden
- * dies geschieht mit den Programmen "halt" oder "reboot" bzw.
- * im Mehrbenutzer-Betrieb mit "shutdown"
- * der Affen-, bzw. Informatikergriff ALT-CTRL-DEL funktioniert und darf benutzt werden

12.6 Inbetriebnahme des LINUX-Systems

12.6.1 LILO (Linux LOader)

- * LILO lädt LINUX und leitet den Bootvorgang ein
- * ermöglicht Wahl zwischen versch. Betriebssystemen (Tasten "ALT" und "TAB")
- * Konfigurationsdatei "/etc/lilo.conf"
- * Konfiguration muss durch Eingabe von "lilo" aktiviert werden
- * Dokumentation unter "/usr/lib/lilo"
- * LILO kann Parameter an den LINUX-Betriebssystemkern übergeben:

- Eingabe der Parameter auf dem LILO–Prompt (nach dem Namen des zu ladenden kernel–Namens)
- mit 'append = "..."' in "/etc/lilo.conf"

12.6.2 Der Bootvorgang

- * der Betriebssystemkern startet nach dem Erkennen und dem Initialisieren der Hardware das Urprogramm "init"
- * "init" führt alle Kommandos zum Starten des LINUX–Systems durch
- * Haupt–Konfigurationsdatei ist "/etc/inittab"
- * Verzweigung auf shell–Skripte in "/etc/rc.d" oder "/etc/init.d" (vergleichbar mit "autoexec.bat")
- * für eigenen Ergänzungen sollte man "/etc/rc.d/rc.local" verwenden
- * Bootvorgang wird z.T. in "/var/adm/messages" protokolliert

12.7 Die Uhrzeit

WARNUNG: vor dem Setzen der Zeitzone alle anderen Arbeiten beenden, d.h. Dateien schliessen und alle Programme beenden! Ansonsten können Dateien mit "kaputten" Zeitmarken versehen werden und ziemlich üble, "nicht erklärbare" Fehler verursachen.

Unter LINUX gibt es "drei verschiedene Uhrzeiten":

- die der auf der PC-Hauptplatine installierten CMOS-Uhr,
- die LINUX-interne Uhrzeit (Systemzeit) und
- die von verschiedenen "Uhren" angezeigte Zeit

clock -u -w date -s

CMOS-Uhr <----- LINUX <-----

(Mainboard) Systemzeit Benutzer-Uhr

(GMT) -----> (GMT) -----> (MET)

clock -u -s date; xclock

Die Option "-u" des clock-Befehls bewirkt eine unveränderte Übernahme der CMOS-Zeit als Systemzeit (üblicherweise während des Bootvorgangs). Unter Verwendung der Regeln aus der Datei "/usr/lib/zoneinfo/localtime" berechnen Programme wie "date" und "xclock" aus der Systemzeit die Uhrzeit der gewählten Zeitzone.

* zu diesen Regeln gehört z.B. die automatische Umstellung von

Sommer- auf Winterzeit

* ist LINUX das einzige Betriebssystem auf dem Rechner, so setzt

man die Systemzeit auf "GMT" (Greenwich Mean Time) und legt

einen Verweis von "localtime" auf "MET" an

* ansonsten genau umgekehrt (d.h. Regeln werden nicht beachtet)

12.8 Konfiguration des Betriebssystemkerns

- * der Betriebssystemkern wird als "kernel" bezeichnet
- * neuste Kernelversionen (lehz) werden als "patches" vertrieben und können mit "cd /usr/src; patch -p0 < patch_1.3.54" in die bestehenden (!) Quelltexte integriert werden
- * Quelle: "nic.funet.fi", Verz. "/pub/Linux/PEOPLE/Linus/v1.3"
- * Kernelversionen unter "1.3" erzeugen i.d.R. schwere Minderwertigkeitsgefühle
- * neuen kernel aus Quelltexten selbst bauen:
"cd /usr/src/linux"
"make config" (Fragen beantworten)
"make dep; make clean; make zImage"
diese Befehle erzeugen im Verzeichnis
"/usr/src/linux/arch/i386/boot" eine Datei "zImage",
die den Betriebssystemkern darstellt; dieser muß nach "/"
(oder "/boot") kopiert und in "/etc/lilo.conf" eingetragen
werden; SIEHE 12.6
- * Erzeugen von NACHTRÄGLICH ladbaren Treibern ("module")
mit
make modules; make modules_install
(das ist coool)
- * Verwaltung der Treiber mit "insmod", "lsmod", "rmmod"

13. Beschaffung und Installation von Software

13.1 Beschaffung von Software (oder: "Gibt es XYZ für LINUX?")

- * die ftp-Server "sunsite.unc.edu" und "tsx-11.mit.edu" sind die Sammelstellen für Software für LINUX; beide werden von vielen anderen ftp-Servern weltweit gespiegelt (nächtliche Aktualisierung) und befinden sich auf den meisten CD-sets
- * die "LINUX Software Map" (LSM) gibt einen Überblick über die auf diese ftp-Server aufgeladenen Programme; z.B. via URL

<http://siva.cshl.org/lsm/lsm.html>

- * das ganze Internet ist voll mit Programmen, die nicht speziell für LINUX geschrieben sind, die aber einwandfrei unter LINUX laufen
- * um im Internet komfortabel nach Software zu suchen, kann man das Programm "xarchie" oder die verschiedenen Suchformulare im WWW benutzen; z.B. die URL (Adresse im WWW):

<http://query1.lycos.cs.cmu.edu/lycos-form.html>

13.2 Installation von Software allgemein

- * der überwiegende Teil der Software für LINUX liegt im Quelltext vor und muß i.d.R. noch übersetzt werden (in den Paketen der Distributionen ist dies natürlich schon geschehen)
- * die Programmpackete sind i.d.R. mit "tar" gepackt und mit "gzip" komprimiert, und enden mit ".tar.gz" oder kurz ".tgz";
entpacken ist mit "tar -xzf dosemu0.60.tgz" (Beispiel) möglich
[entpacken in den Verzeichnissen "/usr/src" oder "\$HOME/src"]

- * das Erzeugen eines lauffähigen Programms und dessen Installation wird in den meisten Fällen durch ein sogenanntes "Makefile" gesteuert; eine genaue Beschreibung der Installation ist in den Dateien "INSTALL" und/oder "README" des jeweiligen Packets zu finden
- [Installation unbedingt nach "/usr/local"]
- * für X11-basierte Programme wird das "Makefile" meist automatisch durch den Befehl "xmkmf" erzeugt
- * für GNU-Programme kann ein "Makefile" automatisch durch den Befehl "./configure" erstellt werden
- * das "Makefile" wird durch den Befehl "make" abgearbeitet
- * durch "make install" wird das Programm i.d.R. installiert

13.3 Platzbedarf einiger Verzeichnisse

- * die root-Partition "/" benötigt ohne die Verzeichnisse "/usr", "/var", "/tmp" und "/home" in allen Distributionen weniger als 20 MB und sollte auf einer eigenen Partition untergebracht sein
- * das "home"-Verzeichnis sollte ebenfalls auf einer eigenen Partition untergebracht werden (mind. 20 MB (pro Benutzer))
- * hat man genügend Festplattenplatz, so sollte man eine eigene Partition für das Verzeichnis "/usr/local" spendieren (mind. 50MB)

Wird einmal das System durch ein neues ersetzt, so braucht man diese Verzeichnisse nach der Neuinstallation lediglich in den Baum zu hängen und hat die persönlichen Daten und die mit viel

Mühe nachträglich installierte Software ("/usr/local") sofort wieder zur Verfügung.(Setzt natürlich voraus, daß man immer brav nach "/usr/local" installiert hat.)

* die Verzeichnisse "/var" und "/tmp" mit den ständig ändernden Informationen, sollten auf Systemen mit mehreren Benutzern (ab 5) ebenfalls auf eigenen Partitionen untergebracht werden (var: mind. 20 MB, tmp: mind. 20 MB)

Pfad- und Dateinamen

Sollten die erwähnten Namen nicht auf den eigenen Dateibaum zutreffen, so kann man sich mit den folgenden Befehlen behelfen:

```
"strace -f -o dump emacs"
```

ruft den "emacs" auf und protokolliert unter anderem die Zugriffe auf Konfigurationsdateien in der Datei "dump"

```
"find /usr -name "xinitrc" -print"
```

sucht nach der Datei "xinitrc" im "/usr"-Ast und zeigt alle Vorkommen mit dem kompletten Pfadnamen an

```
"grep -i loadkeys /etc/rc.d/*"
```

zeigt die Dateien im Verzeichnis "/etc/rc.d" an, in denen

der Befehl "loadkeys" ausgeführt wird

Auszug aus dem Vorwort zur "GNU GENERAL PUBLIC LICENSE"

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

[..]

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

Linus Thorvalds hat es mal als beste Entscheidung in Zusammenhang mit der Entwicklung von LINUX bezeichnet, daß er LINUX unter die GNU Lizenz gestellt hat.