

Betriebssysteme

Aufgaben
Bootprozeß
Basisprogramme

Kernel

- Ist ein Programm, also Software
- Läuft auf der Hardware, dem Computer
- Verwaltet die Ressourcen: Speicher, CPU-Zeit
- Virtualisiert Hardware: Ein-/Ausgabe, Festplatten, Wechselmedien, Netzwerk
- Erlaubt Kommunikation zwischen Prozessen
- Monolithisch, Makro-, Micro-, Nanokernel
 - Auslagerung von Aufgaben aus dem Kernel

Booten des Kernels

- BIOS: Minibetriebssystem des Herstellers
- Bootlader: sehr klein, lädt den Kernel
- Aufgaben des Kernels beim Booten:
 - Initialisiert Hardware
 - Mountet die Root-Partition (nur lesend)
 - Lädt benötigte Module und startet Kernel-Prozesse
 - Startet den init-Prozeß (z.B. /sbin/init)
 - Läuft als „Idle-Prozeß“ und im Interrupt

Init-Prozeß

- *SystemV*: Konfigurierbar mit **Runleveln** (/etc/inittab, /sbin/rc.X/start-script)
- *BSD*: Fester Bootvorgang (/etc/rc)
- Jedes Unix hat eigene Philosophie und Software!
- Init startet und restartet alle benötigten Prozesse
- Alle weiteren Prozesse sind Kinder von init
 - Init ist Prozeß Nummer 1
 - Init räumt auf: Entfernt Zombies

Shells

- *Kommandozeile*: Hauptschnittstelle zum Unix
- Stellt Laufzeitumgebung für Anwendungen bereit
- Navigiert im Dateisystem, Arbeitet mit Dateien
- Dateisuche (Globbing)
- Anhalten/Restarten/Parallelbetrieb (Job Control)
- Verkettung von Anwendungen (Pipes)
- Programmierbar, scriptfähig
- Eingabekomfort (Editing, History, Completion)

Shell-Varianten

- sh – Bourne Shell: minimal, für Skripte
- csh – C-Shell: C-Syntax, Job Control
- ksh – Korn Shell: POSIX, Job Control, Editing
- bash – Bourne Again Shell: Gnu-Rewrite, beliebt
- tcsh – Tenex-C-Shell: C-Shell plus Editing
- zsh – Leistungsfähig, bedienfreundlich
- ash – Almquist-Shell: Klein, schnell, embedded
- dash – Debian-Almquist-Shell: Für Skripte

Coreutils – „One tool, one job“

- Werkzeugkasten: flexibel, schnell
- Spezialisten, keine Allrounder
- Leistungsfähigkeit durch Pipes:
 - A|B: Ausgabe von Programm A ist Eingabe von B
 - Keine temporären Dateien, kein Speicherbedarf
 - Programme als Filter: A|B|C
- Eingabe in Argumente: xargs
- Als Argument: /dev/stdin, /dev/stdout, /dev/stderr

Textutils

- Dateiausgabe: cat, tac, nl, od
- Teilausgabe: head, tail, split
- Unterwegs abzweigen: tee
- Sortiertes: sort, shuf, uniq, comm, diff
- Zeilenweise Bearbeitung: tr, rev, grep, sed, awk
- Zusammenfassen: wc, md5sum, sha1sum
- (Ent)Packen: (un)compress, gzip, zcat, bzip2

Fileutils

- Verzeichnisse: cd, pwd, ls, find, dirname, basename, mkdir, rmdir
- Dateien: cp, mv, ln, stat
- Sonderdateien: mknod, mkfifo, dd
- Attribute: touch, chown, chgrp, chmod, attr
- Festplatten: df, du, sync
- Archivierung: ar, tar, cpio

Restliche Coreutils

- Prozesse: ps, kill
- Nutzer: id, whoami, groups, users, logname, who
- System: date, uname, hostname
- Tests: true, false, test, [, expr
- Ausgaben: echo, printf, yes
- Verschiedenes: sleep, chroot, nice, nohup, su
- *Wichtigstes Kommando: **man***

Dateisystem

- / – Root: Basis für Init
- /boot – Für Bootlader
- /dev – Geräte
- /bin – Programme
- /sbin – Admintools
- /etc – Konfigurationen
- /lib – Bibliotheken
- /root – Für Admin
- /usr – Anwendungen
- /var – Arbeitsdateien
- /home – Nutzerdateien
- /tmp – Kurzlebiges
- /mnt – Externes
- /opt – Fremdsoftware

Editoren

- Bearbeitung von Texten und Programmen
- Reine ASCII-Texte bearbeiten
- Hauptlinien: (ed), vi / emacs / xedit
- Aufgaben:
 - Navigieren, Suchen, Ersetzen, Einrücken
 - Hilfe anzeigen, Zugriff auf Dateisystem und Shell
 - Formatieren, Compilieren, Syntaxhervorhebung
 - Vervollständigen, Makros, Programmierung