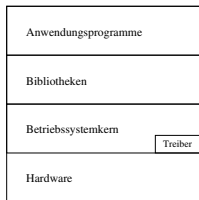


**Inhalt:**

- struktureller Aufbau
- Betriebssystemkern
- Benutzerverwaltung
- Prozessverwaltung
- Dateisystem
- Transferprotokolle



Ein Betriebssystem ist ein Programm, das dem Benutzer den Zugriff auf die Hardwarekomponenten eines Rechners ermöglicht.

Es stellt somit das Bindeglied zwischen Hardware und Benutzer dar.

## Rolle des Computers, des Betriebssystems und der Programmierung

- Computer sind die meist überschätzten Werkzeuge: hochkomplexe und damit (auch) fehleranfällige Systeme  
⇒ Kritischer und bewußter Umgang nötig!
- Der Computer ist ein nützlicher Idiot... aber ein schneller!
- Erfolg des Einsatzes steht und fällt daher mit korrekter Programmierung  
⇒ Mist rein, Mist raus!
- Systematisch-Logisches Denken und strukturiertes Programmieren wichtiger als Beherrschung der Syntax einer Programmiersprache!
- Algorithmus als abstrakter Lösungsweg  
Beispielaufgabe: „Kürze einen Bruch!“

## Aufgaben des Betriebssystemkerns

*Kernel* — Steuerung der gerätenahen Grundfunktionen des Rechners

- Prozessverwaltung
- Speichermanagement
- Ressourcenvergabe
- Benutzerverwaltung
- Dateiverwaltung
- Ein-/Ausgabesteuerung

## Multi-User-Betriebssystem

Mehrere Benutzer können den Rechner gleichzeitig benutzen, können sich aber nicht gegenseitig stören. ⇒ Benutzerverwaltung

## Multi-Tasking-Betriebssystem

Verschiedene Prozesse können auf dem Rechner zeitlich quasi-parallel abgearbeitet werden. ⇒ Prozessverwaltung

Jeder Benutzer eines Linux-Systems erhält vom Administrator eine eindeutige Benutzerkennung (engl. *account*).

- Benutzername *username*
- Passwort *password*
- Heimatverzeichnis *home directory*

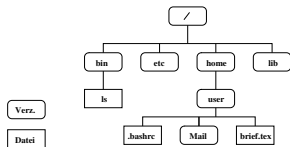
Es gibt einen speziellen Benutzer **root**, der keinen Beschränkungen im Linux-System unterliegt.

## Linux — Prozessverwaltung

## Linux — Dateisystem

Linux ist ein **Multi-Tasking-Betriebssystem**, d.h. mehrere Programme können parallel ausgeführt werden und sich durch Speicherschutz nicht gegenseitig behindern. Wobei zwischen echter Parallelität auf mehreren CPU-Kernen und quasi-parallelem Rechnen auf einem CPU-Kern (Time-Sharing) zu unterscheiden ist.

beispielhafter Aufbau einer hierarchischen Baumstruktur unter Linux



- Ausgangspunkt: / (Wurzel, root)
- Hardwarestruktur nicht erkennbar
- Dateiararten
  - ▶ normale Datei (*plain file*)
  - ▶ Verzeichnis (*directory*)
  - ▶ Link
  - ▶ Gerätedatei, Pipe, ... (EDV2)
- Zugriffsrechte
  - ▶ Eigentümer der Datei
  - ▶ Gruppenzugehörigkeit
  - ▶ andere
- Pfade
  - ▶ aktuelles Arbeitsverzeichnis
  - ▶ relativer Pfad
  - ▶ absoluter Pfad

### X ist eine leistungsfähige und netzwerkfähige grafische Oberfläche

- unabhängig vom Betriebssystem
- eigenständiges Programm
- sprachen- und rechnerunabhängig
- frei verfügbar

Die notwendige Software für die Vernetzung ist integraler Bestandteil von Linux. Das benutzte Kommunikationsprotokoll ist **TCP/IP**.

Das Netzwerk soll für den Benutzer völlig transparent sein, d.h. das Netzwerk arbeitet „unbemerkt im Hintergrund“.

Im Internet stehen verschiedene Dienste zur Verfügung (EDV2):

- HTTP — Hypertext Transfer Protocol (Webseiten)
- SMTP — Simple Mail Transfer Protocol (E-Mail-Versand)
- POP3 — Post Office Protocol 3 (E-Mail-Empfang)
- SSH — Secure Shell (Einloggen auf entferntem Rechner)
- ...

Die besten und ausführlichsten Informationen zu dem Thema bietet das Internet selbst.

z.B.: <http://www.cfd.tu-berlin.de/>