



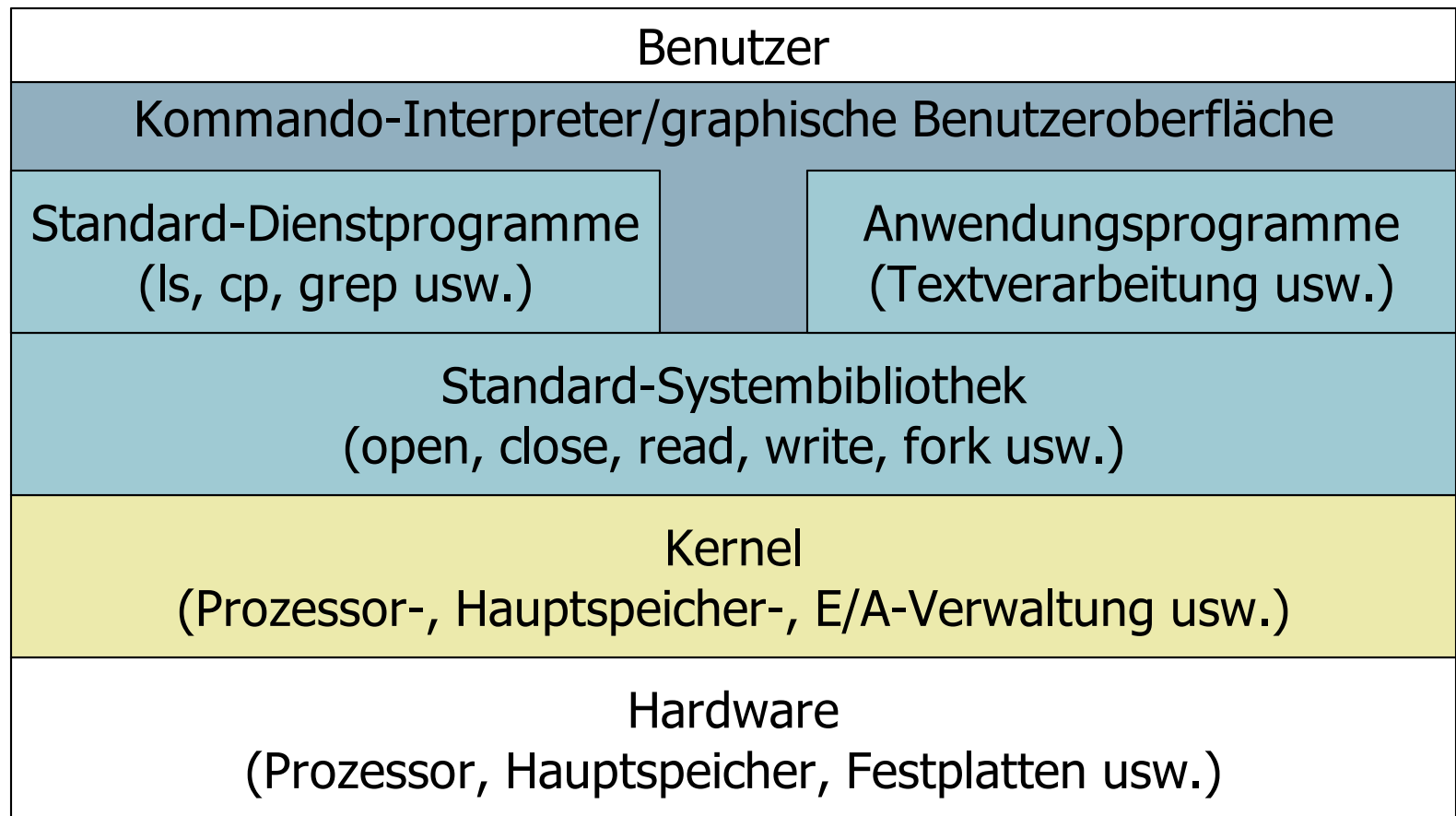
# Betriebssysteme

---

## Einführung in UNIX/Linux



# Grundlegende Architektur





# Bestandteile GNU/Linux

---

- Kernel (Linux im engeren Sinne)
  - Prozessor- und Speicherverwaltung
  - Dateisysteme, Gerätetreiber, ...
- Systemprogramme (vorwiegend GNU)
  - Kommandos: ls, cp, grep, awk, ...
  - Dienstprogramme: vi, emacs, gcc, gdb, ...
  - Daemonen: telnetd, sshd, lpd, crond, ...
- Shells und GUIs
  - Shells: sh, bash, csh, ...
  - GUIs: X-Server, KDE, GNOME, ...
- Anwendungsprogramme
  - Latex, Gimp, OpenOffice, ...



# Eigenschaften von Linux (I)

---

- Multi
  - -tasking (mehrere Prozesse zur gleichen Zeit)
  - -user (mehrere Benutzer zur gleichen Zeit)
  - -plattform (verfügbar für viele verschiedene Architekturen)
  - -processing (symmetrisches Multiprocessing bis zu 16 CPUs)
- Moderne Betriebssystemkonzepte, z. B.:
  - Virtuelle Speicherverwaltung mit Paging, Speicherschutz
  - Leistungsfähige Dateisysteme
  - Statisch und dynamisch gelinkte Bibliotheken
  - Vielfältige Netzwerkunterstützung (Protokolle/Dienste)

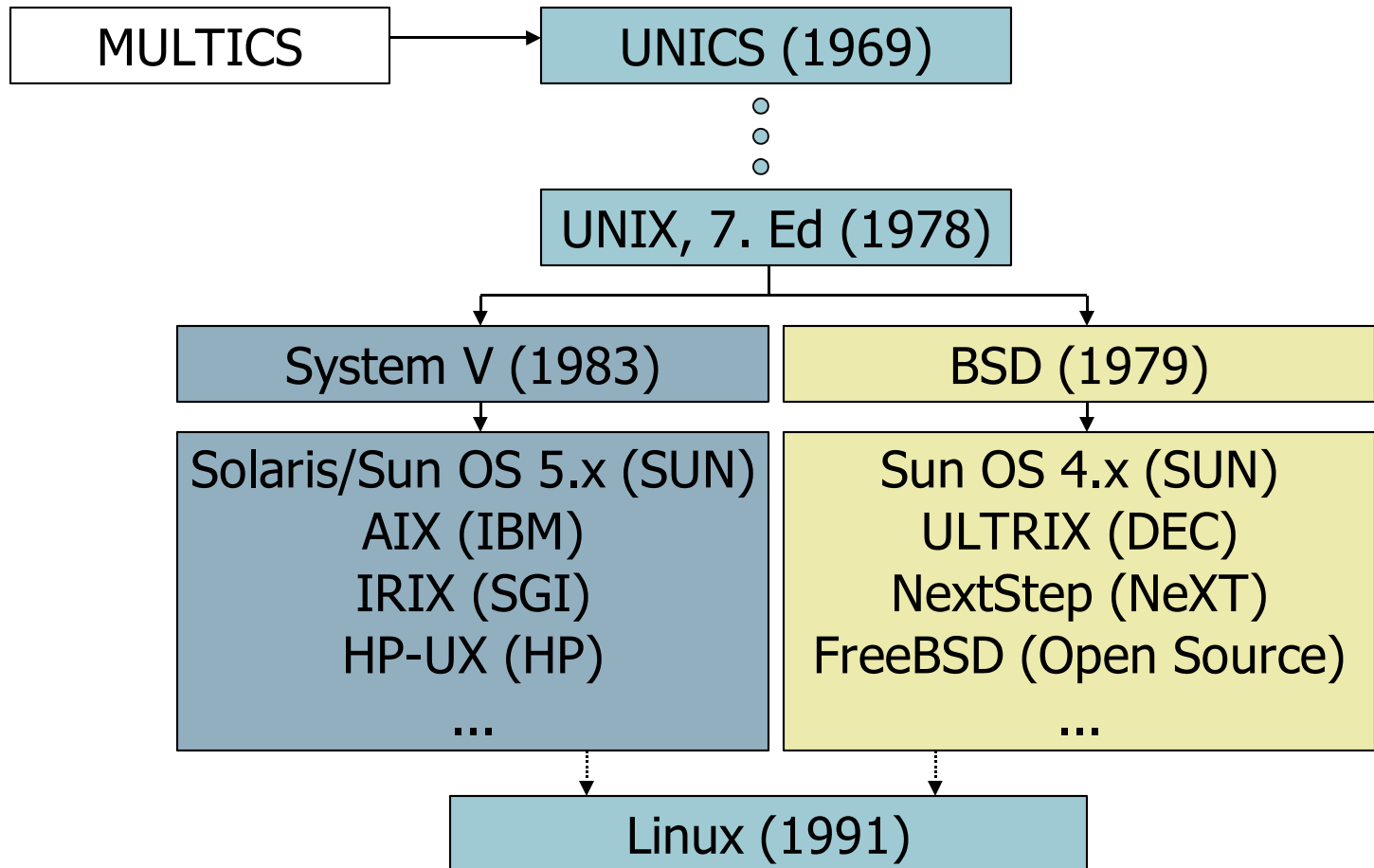


# Eigenschaften von Linux (II)

---

- Open-Source-Projekt mit riesiger Entwicklergemeinschaft:
  - Geringe Anschaffungskosten, keine Lizenzgebühren
  - Quellcode verfügbar (Kernel, Dienstprogramme, fast alle Treiber) → leicht erweiterbar, überprüfbar
  - Breite Hardware-Unterstützung
  - Großes Softwareangebot, zunehmend auch von kommerziellen Herstellern (IBM, Oracle, Sun, Software AG, ...)
- Stabil und schnell
- Größtenteils UNIX Quellcode-kompatibel (POSIX, System V, BSD)
- Vielfältige Einsatzmöglichkeiten (Server, Desktop, ...)

# Kurze UNIX-Historie





# Kurze Linux-Historie

---

- 1991: 21-jähriger, finnischer Student Linus Torvalds beginnt mit der Entwicklung einer freien UNIX-Version
- 1992: Version 0.12 erscheint und wird per ftp im Internet verfügbar gemacht, Newsgroup alt.os.linux wird ins Leben gerufen
- 1993: ca. 100 Programmierer arbeiten am Linux-Code, Linux wird an die GNU-Umgebung angepasst
- 1994: Version 1.0 erscheint, Kernel wird netzwerkfähig, grafische Oberfläche (Xfree86), Linux erscheint unter der GPL, 100.000 Nutzer
- 1995: Linux wird auf weitere Plattformen portiert
- 1996: Kernel-Version 2.0 erscheint, Mehrprozessorbetrieb möglich
- 1997: Immer mehr kommerzielle Softwarepakete für Linux
- 1998 – heute: stürmische Entwicklung, neue Software (KDE, Gnome, OpenOffice...), neue Architekturen (AS/400, ...), neue ...



# Arbeiten mit der Shell

---

- Welche Shell?: `echo $SHELL`
- Shell beenden: `exit`, `control-D`, (`logout`)
- Das Verzeichnis wechseln: `cd`
  - `/` Wurzelverzeichnis (root directory)
  - `/` auch Trennzeichen zwischen einzelnen Verzeichnissen
  - `~` Home-Verzeichnis des Benutzers
  - `..` übergeordnetes Verzeichnis
  - `.` augenblickliches Verzeichnis
- Aktuelles Verzeichnis anzeigen: `pwd` (print working directory)





# Grundlegende Dateibefehle I

---

- Dateien und Verzeichnisse auflisten: **ls**
    - Option **-a**: auch versteckte Dateien anzeigen
    - Option **-l**: ausführliche Informationen anzeigen
- Beispiel:
- ```
-rwxrwx-r-- 4 root root 96 Nov 27 18:28 test.txt
```
- (Zugriffsrechte, Linkzahl, Besitzer, Gruppe, Größe, zuletzt geändert, Name)
- Option **-d**: Verzeichnisisinformation anzeigen
  - Option **-R**: Verzeichnisse rekursiv anzeigen
  - Option **-i**: inode-Nummern (Dateien-IDs) anzeigen



# Grundlegende Dateibefehle II

---

- Verzeichnisse anlegen: `mkdir` (make directory)
  - Option `-p`: fehlende übergeordnete Verzeichnisse mit anlegen
- Änderungsdatum aktualisieren: `touch`
- Eingabe in der Shell ausgeben: `echo`
  - Umleitung in Datei mittels `>` Dateiname
- Inhalt einer Datei ausgeben: `cat`
- Datei kopieren: `cp` (copy)
- Datei verschieben/umbenennen: `mv` (move)



# Grundlegende Dateibefehle III

---

- Links erzeugen: **ln**
  - ohne Option **-s**: Hardlink (Link auf inode)
  - mit Option **-s**: symbolischer Link (Link auf Namen)
  - keine Hardlinks auf Verzeichnisse und andere Dateisysteme möglich
  - Symlinks auf Verzeichnisse und andere Dateisysteme möglich, Konsistenz kann allerdings verlorengehen
- Dateien/Verzeichnisse löschen: **rm** (remove)
  - Option **-i**: Löschen bestätigen
  - Option **-r**: rekursives Löschen von Unterverzeichnissen
- Leere Verzeichnisse löschen: **rmdir** (remove directory)



# Wildcards (Joker-Zeichen)

---

- \* : kein oder mehr Zeichen
- ? : genau ein Zeichen
- [ ] : genau eines der angegebenen Zeichen  
Beispiel:  
`text[12]` passt auf `text1` oder `text2`  
`text[1-3]` passt auf `text1`, `text2`, `text3`
- [ ! ] : genau ein Zeichen, aber keines der angegebenen
- Sonderzeichen entwerten (escaping)
  - durch Anführungszeichen
    - vollständig (einfache Anführungszeichen)
    - teilweise (doppelte Anführungszeichen)
  - durch einen Backslash

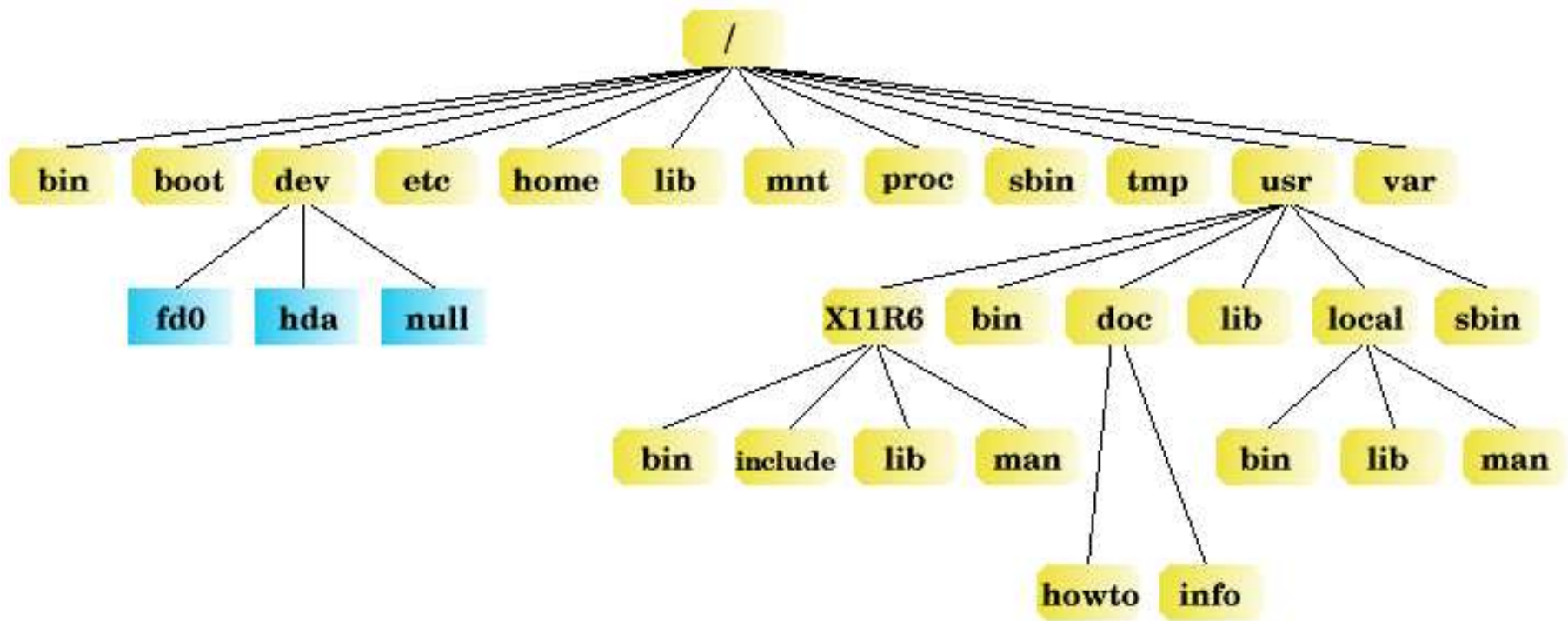


# Dateien durchsuchen und sortieren

---

- Textmuster in Dateien finden  
**grep** (general regular expression print)  
**grep** Optionen Muster Dateien
  - `grep -n hello *.txt`
- Dateien sortieren  
**sort** Datei
  - `sort text1.txt`
- Mehrfach vorkommende Zeilen löschen  
**uniq** Datei
  - `uniq text1.txt`

# UNIX-Verzeichnisstruktur



Graphik aus: „Das Linuxbuch“ von Thomas Ermer



# Dateien finden

---

- Dateien in Verzeichnissen finden  
**find** Verzeichnis **-name** Datei **-print**
  - **find ~ -name "\*.txt" -print**
- Datei im Gesamtsystem suchen  
**locate** Dateiname
  - **locate "\*.txt"**
- Pfad ausführbarer Dateien finden  
**which** AusführbareDatei
  - **which head**



# Zugriffsrechte verwalten

---

- Zugriffsrechte (change mode)  
**chmod** Optionen Datei
  - **chmod u+rw test.txt**  
(u:user, g:group, o:other,  
=:zuweisen, +:hinzufügen, -:entfernen)
  - **chmod 764 test.txt**  
(Drei Octal-Zahlen für die Zugriffsbits)
- Gruppenzugehörigkeit (change group):  
**chgrp** Gruppe Datei
  - **chgrp users test.txt**





# Dateien archivieren und komprimieren

---

- Arbeiten mit dem tape archiver (**tar**)
  - Archiv erstellen  
**tar -cvf** Archivname Dateinamen
  - Archiv entpacken  
**tar -xvf** Archivname
- Dateien (de-)komprimieren mit **gzip**
  - Datei komprimieren  
**gzip** Dateiname
  - Datei dekomprimieren  
**gzip -d** Dateiname



# Ausgabe-Umleitungen

---

- Ausgabe in Datei umleiten: `Befehl > Datei`  
Beispiel: `echo 'Ein Text' > test.txt`
- Ausgabe an Datei anhängen: `Befehl >> Datei`  
Beispiel: `echo 'Noch ein Text' >> test.txt`
- Ausgabe an weiteren Befehl schicken: `Befehl1 | Befehl2`  
Beispiel: `cat beispiel.txt | sort | wc`
- Ausgabe ausgeben und umleiten: `Befehl | tee Datei`  
Beispiel: `ls | tee verzeichnis.txt`
- Weitere Möglichkeiten: Eingaben aus Dateien (`<`), „Hier-Dateien“ (`<<`), Umleitung der Fehlermeldungen,...



# Prozess-Steuerung

---

- Prozess beenden (Strg-c), Prozess anhalten (Strg-z)
- Angehaltenen Prozess im Vordergrund laufen lassen: `fg`
- Angehaltenen Prozess im Hintergrund laufen lassen: `bg`
- Im Hintergrund starten: Befehl ein `'&'` anhängen
- Jobs anzeigen: `jobs`
- Signale senden: `kill -s Signal ProzessID`
  - Signale: `SIGSTOP`, `SIGTERM`, `SIGINT`, `SIGKILL`,...
- Prozesse anzeigen: `ps`
- Sich ständig aktualisierende Prozessdaten: `top`