

# Rechnerpraxis

SS2019 | INF 1

Hochschule  
für Technik  
Stuttgart

02 – Betriebssysteme & Einführung in Unix / Linux

Benjamin Lebherz

[benjamin.lebherz@hft-stuttgart.de](mailto:benjamin.lebherz@hft-stuttgart.de)

Folien zum Teil von Jan Seedorf & Rainer Keller

---

## Finale Abstimmung

**Termin verlegen auf  
Donnerstags, 17:30 - 19:00?**

# Student Life

---

- Benefits
  - z.B. Github Student Developer Pack
- Events
  - z.B. Science Slam, Powerpoint Karaoke
- Meetups & Hackathons

# Betriebssysteme

Hochschule  
für Technik  
**Stuttgart**

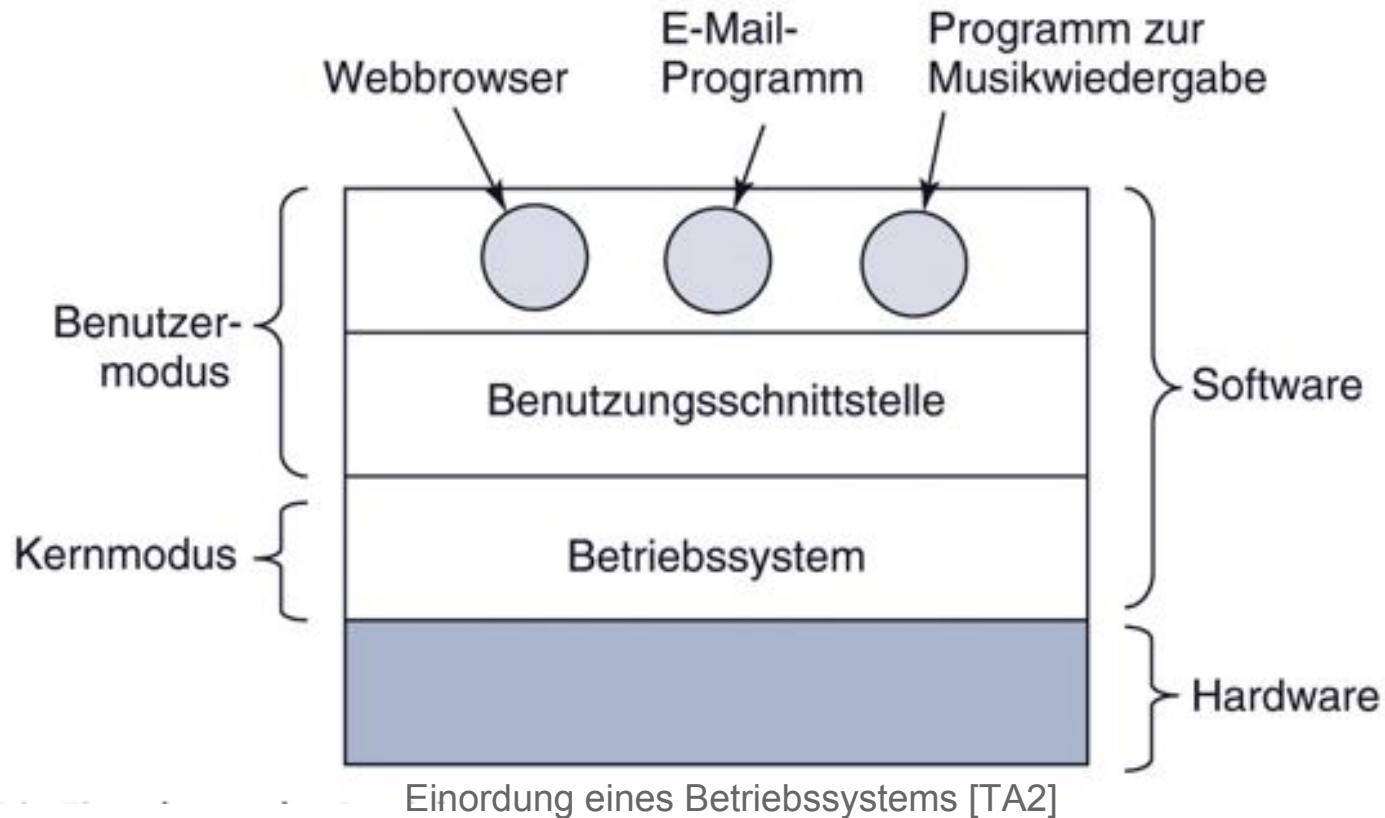
---

# Betriebssystem (Operating System, OS)

---

- „An OS is a program that controls the execution of application programs and acts as an interface between applications and the computer hardware“ [1]
- Ein Betriebssystem ist ein Programm, das als Schnittstelle zwischen Anwender(programmen) und Computer-Hardware fungiert.
  - Zweck: Umgebung bereitstellen, in der Anwender Programme ausführen können.

# Betriebssystem - Einordnung



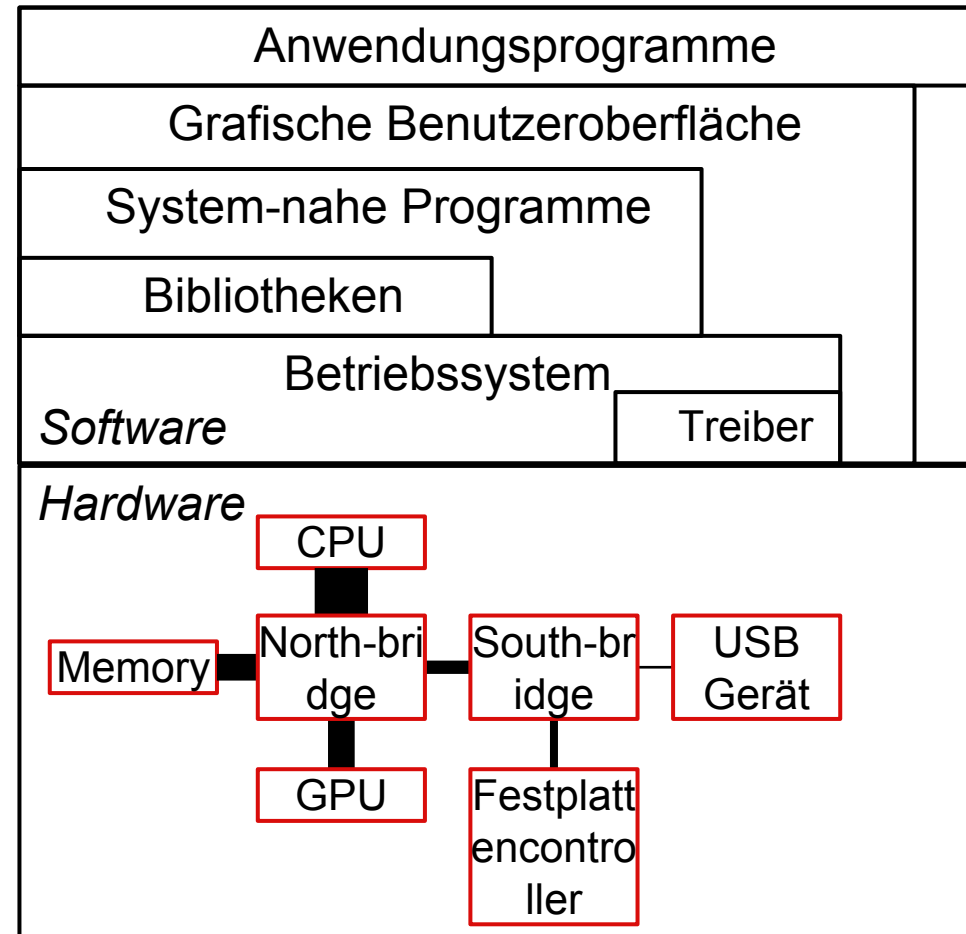
# Aufgaben eines Betriebssystems

---

- Verwaltung der Betriebsmittel (CPU, Speicher, I/O)
  - Koordination, welche Prozesse in welcher Reihenfolge die CPU benutzen dürfen (CPU Scheduling)
  - Eine CPU kann zu einem gegebenen Zeitpunkt nur an einem Prozess arbeiten
- Verwaltung der Daten (Filesystem)
  - Baum-Struktur
  - Verwaltet die Zusammensetzung der Files aus kleineren Blöcken, deren Zuteilung und Freigabe sowie Reorganisation
  - Wechsel der Daten zwischen Hauptspeicher und Festplatte verwalten.  
(**Swap**)

# Betriebssystem - Eingruppierung 1/3

- Was macht ein Betriebssystem
  - abstrahiert von der HW, virtualisiert die Ressourcen
  - Es teilt CPU, Speicher, etc. den Programmen zu; und nimmt sie wieder weg!
  - verwaltet die Dateien
  - Stellt einheitliche Programmierschnittstelle zur Verfügung (API/syscalls)





Zum Betriebssystem gehört **nicht**:

- BIOS
- IDE & Compiler - gcc, clang, ...
- Text-Editor, Browser...
- **Treiber?**

## Betriebssystem Kategorien

- **Mainframes**

High-End-Systeme IBM OS/390 (heute IBM zSeries9) oder Siemens Fujitsu BS 2000/OSD

- **Server Betriebssysteme**

Linux, Unix, Windows

- **PC Betriebssysteme**

Windows, Unix-Derivate, Linux, ...

- **Echtzeit Betriebssysteme**

VxWorks, QNX, Embedded Linux, RT-Linux

- **Embedded Systems**

Embedded Linux, VxWorks, QNX, MS Windows Mobile, MS Windows 8

- **Smartcard-BS**

Chipkarte mit spezieller JVM

OS zwingend nötig?

# Unix/Linux Historie



# Unix/Linux Historie

---



**Linus Torvalds**

Hochschule  
für Technik  
Stuttgart



# Linux Überblick

---

- Welche Linux-Distributionen gibt es? Und warum so viele?
  - Überblick, Rangliste, usw.: <https://distrowatch.com>
- Live-Distributionen? Welche Eigenschaften haben diese?
- Worin unterscheiden sich Distributionen?

# Filesystem

(Dateisystem)

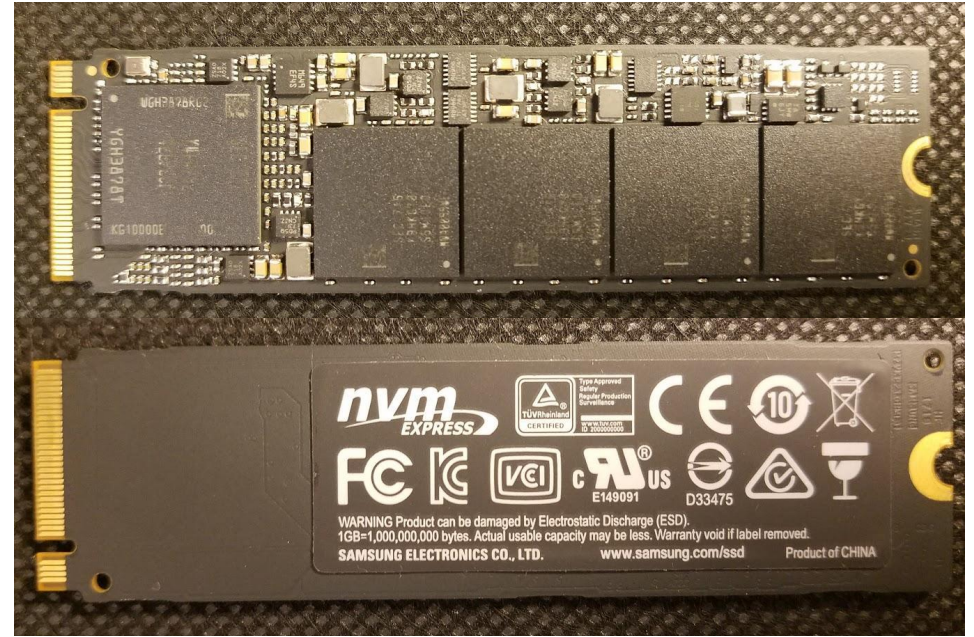
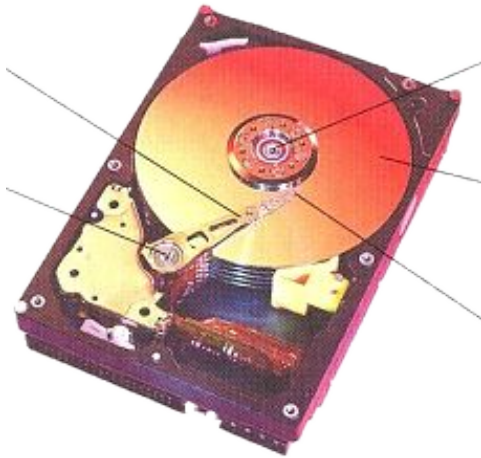
Hochschule  
für Technik  
**Stuttgart**

---



# Storage (persistenter Speicher)

- SSDs  
Speicherchips / Zellen
- Festplatten  
Magnetscheiben

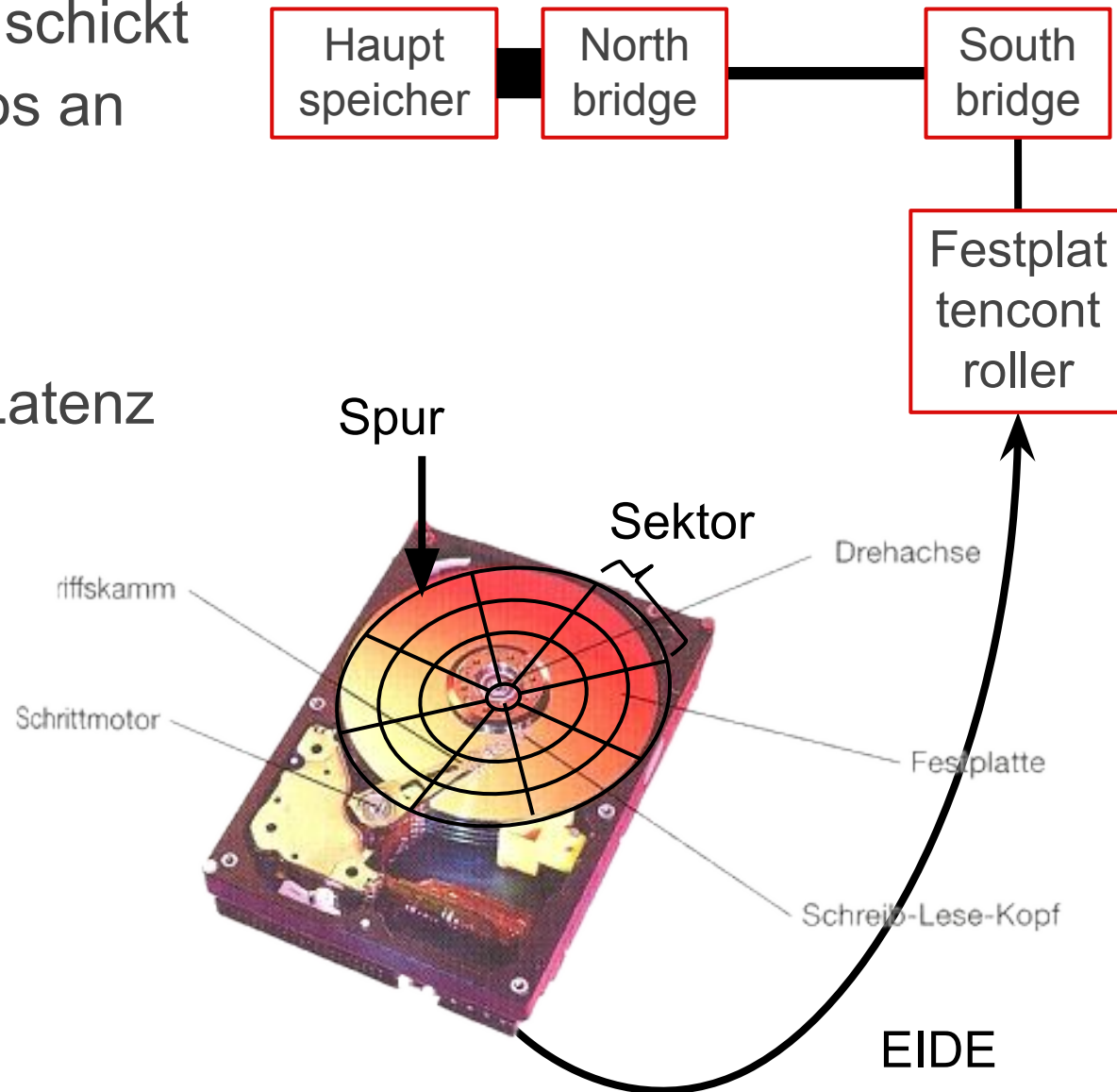


## Anschluss einer Festplatte

- PCIe/NVMe
- SATA
- SCSI/SAS (Serversysteme)
- EIDE (ältere PCs)

# Storage (persistenter Speicher)

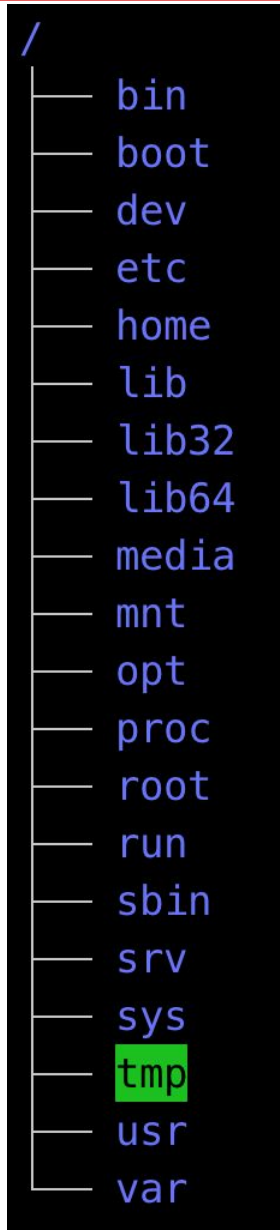
- Zum Auslesen von Daten schickt der Prozessor Kommandos an den Festplattencontroller
- Zugriffszeit = Suchzeit + Latenz



## Dateisysteme

- FAT32: Dateisystem von MS-Dos / Windows... Dann NTFS
- Ext2,3,4: Das Extended FS ist *das* Dateisystem unter Linux
- ReiserFS: vom russischen Programmierer Hans Reiser
- Iso9660: Filesystem von CD-Images
- NFS: Network File System von Sun
- Lustre: Paralleles Filesystem
- ZFS: Zetta File System von Sun (in Verwendung auf LIDA)
- ExFat: Extended File Allocation Table (SD-Karten)

# Filesystem

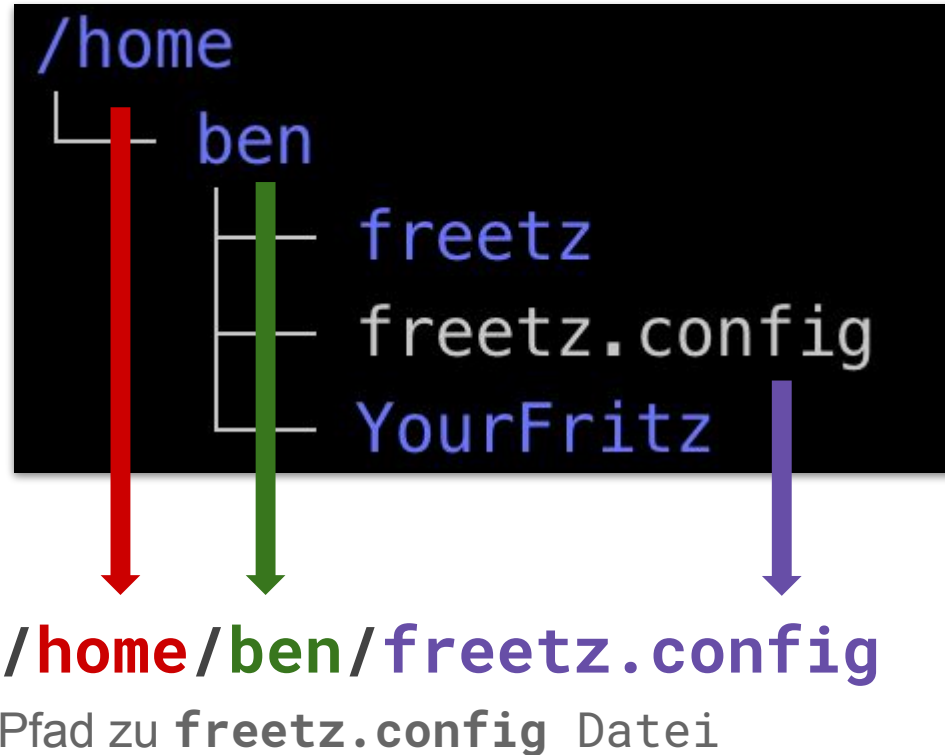


- Hierarchisches Dateisystem (**fs**)
- Baum-Struktur
- / (Slash) als Trenner

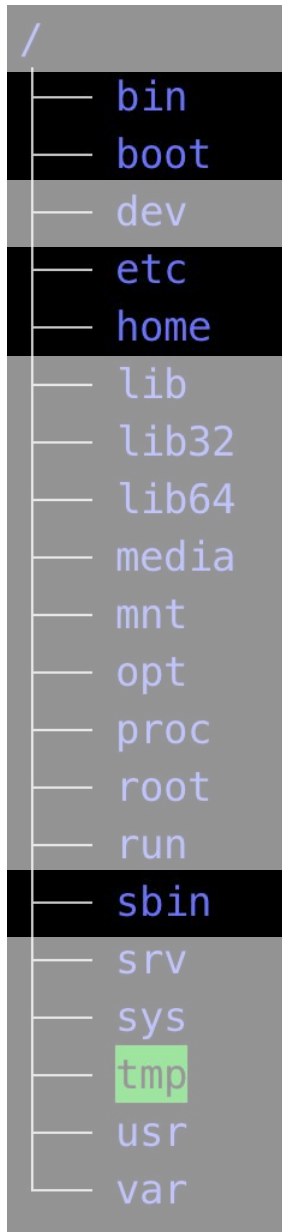


Pfad zu **freetz.config** Datei?

# Betriebssystem Ressource Dateisystem 1/3

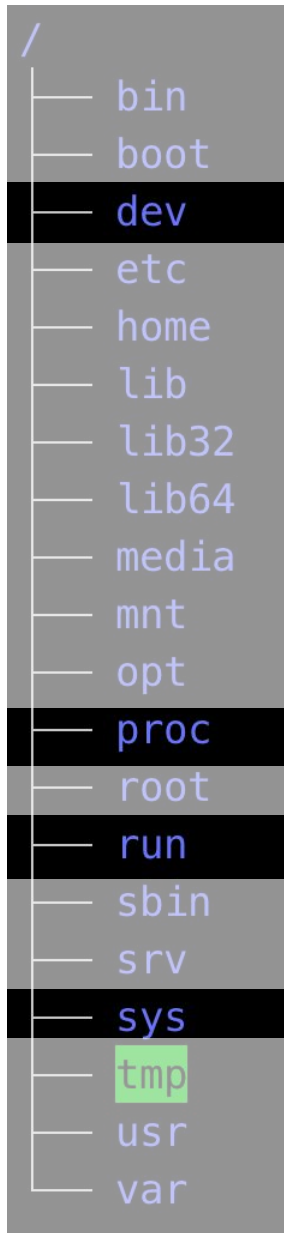


# Filesystem /bin & /boot



- /bin & /sbin
  - Binaries des Systems, zb die Coreutils
- /boot
  - Bootloader
  - Boot Konfiguration
  - EFI Partition/Dateien
- /etc
  - Konfigurationsdateien für Dienste usw.
- /home & /root
  - Home Verzeichnisse

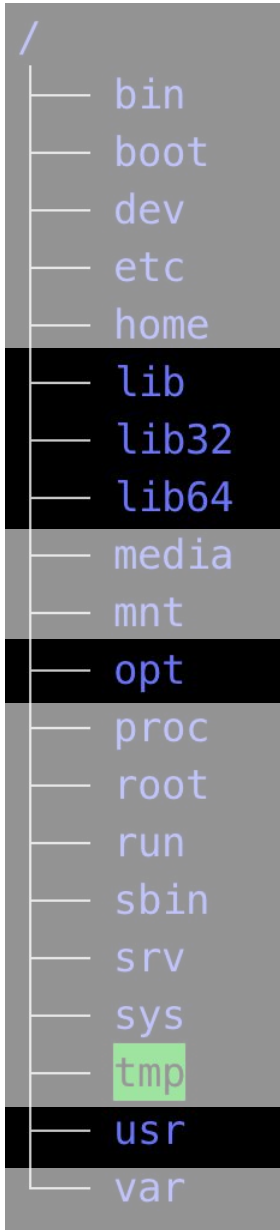
# Filesystem /dev, /proc, /run & /sys



- /dev
  - Geräte-"Dateien"
  - Festplatten, Bluetooth-Adapter, USB-Sticks, ...
- /proc & /sys
  - Infos/Einstellungen des Systems
  - Auslesen wie normale Textdateien
  - Vorsicht! Auch ändern der Werte möglich
- /run
  - Infos über laufende Prozesse/Dienste
  - z.B. PID-Files



# Filesystem **/lib**, **/usr**, **/opt**



- **/lib\***
  - Bibliotheken
- **/usr**
  - Installierte Apps (**/usr/bin**) und Bibliotheken (**/usr/lib\***)
  - **/usr/local**
    - Hier auch wieder bin/ und lib\*/ für lokale Pakete/Binaries
- **/opt**
  - z.B. Manuell Installierte Pakete

# Übung

---



# Shells

---

- Mit einer Shell (Kommandozeile) werden Programme gestartet
- Die Shell ist die niedrigste, systemnahe Software, mit der ein User interagiert
- Shells haben Eigenschaften von Programmiersprachen

## Shells

- Bourne Shell (sh)
- A Shell (ash)
- **Debian Almquist shell (dash)**
- Korn Shell (ksh)
- Z-Shell (zsh)
- **Bourne Again Shell (bash)**

## Mit der Bash können wir

- Systemnahe Programme starten
  - Grafische Programme starten
  - Programme killen
  - Komplexe Shell-Skripte ausführen
- 
- In der Kommandozeile steht die Eingabeaufforderung (**Prompt**)

```
(11:37:03) ben@Bea /usr/bin $ █
```

# Bash

- Diese Zeile setzt sich zusammen aus:

```
(11:37:03) ben@Bea /usr/bin $
```

Username  
**ben**

Server / PC  
**Bea**

PWD / akt.Verzeichnis  
**/usr/bin**

Allgemein: **<benutzer>@<rechner> <pfad> \$**

# Bash

- Diese Zeile setzt sich zusammen aus:



A screenshot of a terminal prompt: `(11:49:54) ben@Bea ~ $`. Three arrows point from parts of the prompt to their explanations below: a red arrow from `ben` to 'Username ben', a green arrow from `Bea` to 'Server / PC Bea', and a purple arrow from `~` to 'PWD / akt.Verzeichnis ~'.

Username  
**ben**

Server / PC  
**Bea**

PWD / akt.Verzeichnis  
**~**

~ (Tilde)  
= **\$HOME**

Home Verzeichnis des aktuellen Users  
als **Umgebungsvariable**

# Bash

---

Momentan in **/home**, wechsel nach **/home/ben/vorlesung/hft**

- Absolut                   \$ **cd** **/home/ben/vorlesung/hft**
- Relativ                   \$ **cd** **ben/vorlesung/hft**
- Tilde                    \$ **cd** **~/vorlesung/hft**
- Umgebungsvar.   \$ **cd** **\$HOME/vorlesung/hft**

# Bash

---

Momentan in **/usr**, wechsel nach **/home/ben**

- Absolut     \$ **cd** **/home/ben**
- Relativ     \$ **cd** **../home/ben**
- Relativ     \$ **cd** **./../home/ben**
  
- **.**        steht für das aktuelle Verzeichnis
- **..**        steht für das übergeordnete Verzeichnis



- \$HOME
  - Home Verzeichnis des aktuellen Users
- \$PWD & \$OLDPWD
  - (vorheriges) aktuelles Verzeichnis
- \$PS1
  - Prompt Konfiguration
- \$USER
  - Username
- \$SHELL
  - Verwendete Shell

# Coreutils

---

[ arch b2sum base32 base64 basename basenc **cat**  
chcon chgrp chmod chown chroot cksum comm coreutils  
**cp** csplit cut date dd df dir dircolors dirname du  
**echo** env expand expr factor false fmt fold groups  
head hostid hostname id install join kill link ln  
logname **ls** md5sum **mkdir** mkfifo mknod mktemp mv nice  
nl nohup nproc numfmt od paste pathchk pinky pr  
printenv printf ptx **pwd** readlink realpath **rm** rmdir  
runcon seq sha1sum sha224sum sha256sum sha384sum  
sha512sum shred shuf sleep sort split stat stdbuf  
stty sum sync tac tail tee test timeout **touch** tr  
true truncate tsort tty uname unexpand uniq unlink  
uptime users vdir wc who whoami yes | grep **man**

# man

Manual, Handbuch zum jeweiligen <Stichwort>

```
$ man [<Kapitel>] <Stichwort>
```

<Stichwort> ein Shell-Befehl oder System Call

<Kapitel> Nummer des Kapitels im Handbuch

## Beispiele

```
$ man ls
```

Info über **ls** Befehl

```
$ man chmod
```

Info über **chmod** Befehl & Syscall

```
$ man 1 chmod
```

Info nur über **chmod** Befehl

# ls

List, Verzeichnis Inhalt anzeigen

```
$ ls [<Optionen>] [<Verzeichnis>]
```

<Optionen>      Optionen, siehe **man** Page oder **--help**

<Verzeichnis>    Anzuzeigendes Verzeichnis

## Beispiele

```
$ ls
```

Inhalt des aktuellen Verzeichnisses

```
$ ls /home
```

Inhalt von /home

```
$ ls -lah
```

Inhalt des aktuellen Verzeichnisses  
als Liste, mit versteckten Dateien,  
“human readable” Größen

# Kopieren, Verschieben, Umbenennen

---

- Kopieren mit cp (copy)

```
$ cp $HOME/fileX directoryY/fileZ
```

```
$ cp -r dirX /backup/2019-04-02_backup
```

- Verschieben

```
$ mv fileX directoryY/fileZ
```

- Aktuelles Verzeichnis anzeigen

```
$ pwd
```

```
$ echo $PWD
```

- Leere Datei erzeugen

```
$ touch ../file42
```

# Übungen

---

- Personalisiere die **Bash Prompt** (`export PS1=" ... "`)
- Erkunden sie das FS mit **cd**. Haben sie überall Zugriff?
- Lege einen Ordner **/home/user/exercises-02** in deinem Home an und bearbeite die Aufgaben dort. Verwende beim anlegen des Ordner die **\$USER** Variable (Tipp: `env` Befehl)
- Liste alle Verzeichnisse (keine Dateien!) die **\*backup\*** enthalten im übergeordneten Verzeichnis, auch versteckte.
- Kopieren/Verschieben sie mit **relativen und absoluten Pfaden**
- Erstelle mit **echo** oder **cat** die Datei `tabelle.txt` mit folgendem Inhalt:  
first, last, age, sex    Zeige nun die Datei in der Shell an  
Jane, Tarzan, 23, F  
John, Doe, 25, M  
Bob, Stevens, 65, M