



SV3: Linux Server

Dateisysteme

Agenda

Dateisysteme

- Hierarchie des Dateisystems
- Dateitypen
- Systemverzeichnisse

Befehl des Tages

ls

- Dateien anzeigen
- Symbol für aktuelles Verzeichnis
- Symbol für übergeordnetes Verzeichnis
- Dateitypen anzeigen
- inode-Nummer anzeigen

Stellen Sie sich vor, Sie sind in einer fremden Stadt.

Was tun Sie, um sich nicht zu verlaufen? Sie kaufen sich einen Stadtplan. Mit einem Stadtplan findet man sich aber erst zurecht, wenn man weiß, wo man sich befindet.

- Genauso ist es in unserem Linux/Unix-System.
- Das System hat uns nach dem Anmelden irgendwo in dem Dateibaum unseres Systems platziert – und zwar in unser sog. »Home-Directory«.

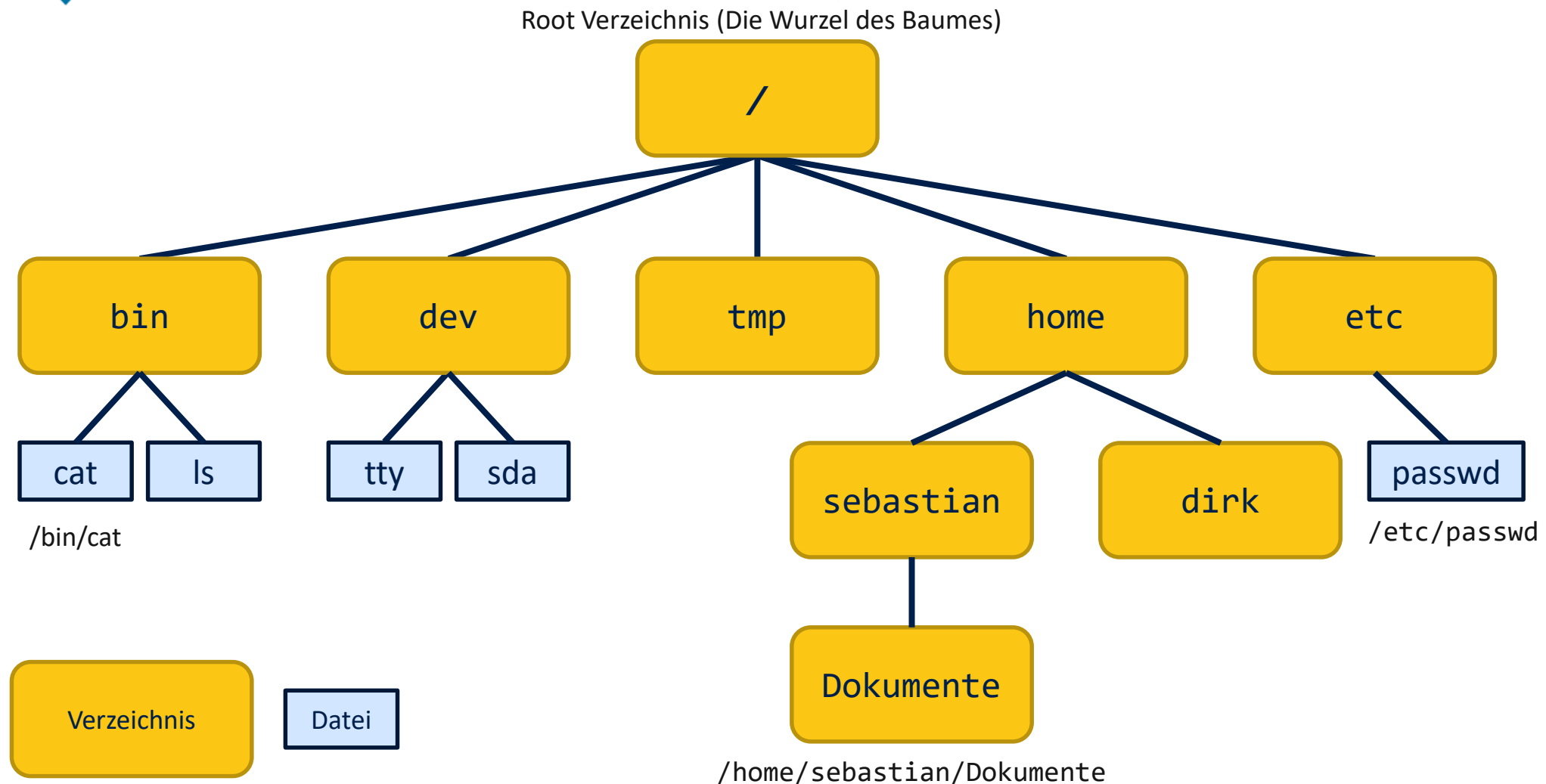
Wo Sie sich gerade in dem großen Dateibaum des Systems befinden, verrät das Kommando pwd.



pwd

pwd – Das Kommando, um sein aktuelles Verzeichnis anzuzeigen

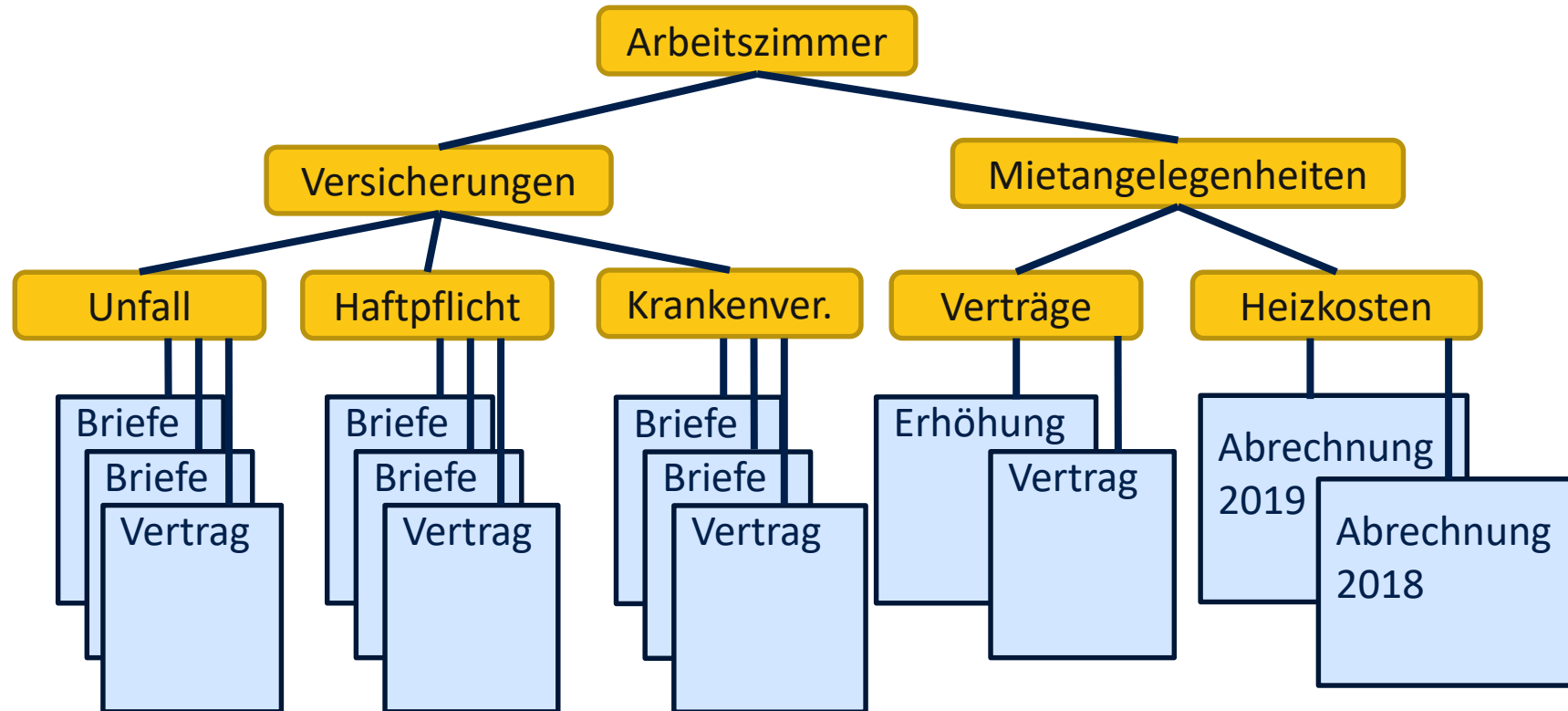
Dateisysteme



Beispiel:

Wenn Sie nun einen bestimmten Haftpflicht-Versicherungsvertrag nachsehen wollen, wissen Sie genau, wo dieser Ordner steht und welches Register Sie aufschlagen müssen.

- Ihr Weg dahin: ***Arbeitszimmer → Versicherungsordner → Register Haftpflicht***
- Überträgt man dies auf ein Linux/Unix-Dateisystem, so könnte dies wie in der Abbildung aussehen:



Unser Weg im Linux/Unix-Dateisystem über die einzelnen Directories zu der gewünschten Datei wird als Pfad bezeichnet.

- Dieser beginnt mit der **root**. Die **root** hat als Directory den Namen „/“

Der Schrägstrich am Anfang eines Pfadnamens bedeutet also immer, dass der Pfad im root-Directory beginnt. Zwischen den weiteren Directories wird der Schrägstrich als Trennungszeichen verwendet.

Dateisysteme

Mit dem Kurzformat `ls -FR` sieht die Ausgabe so aus:

```
sb@ub:~/test$ ls -FR
.:
befehle 'D*' daten/ Datum fehler inhalt inhalt2 neu projektA ueb1

./daten:
1/ 2/ 3/

./daten/1:

./daten/2:

./daten/3:
```

Absolute Pfadangabe

Ein absoluter Pfad zeigt unverwechselbar auf genau eine Stelle in einem Dateisystem, unabhängig von dem Arbeitsverzeichnis.

```
sb@ub:~$ pwd
/home/sb
sb@ub:~$ nano /home/sb/test/text.txt
sb@ub:~$
```

Bei der Angabe des absoluten Pfades wird die Datei immer von der Wurzel "/" des Dateisystembaums aus angegeben.

Relative Pfadangabe

Ein relativer Pfad ist relativ zum aktuellen Arbeitsverzeichnis des Benutzers oder der Anwendung. Um innerhalb dieses Arbeitsverzeichnisses zu agieren, müssen die vollständigen Pfade nicht mehr angegeben werden.

```
sb@ub:~/test$ pwd
/home/sb/test
sb@ub:~/test$ nano text.txt
sb@ub:~/test$
```

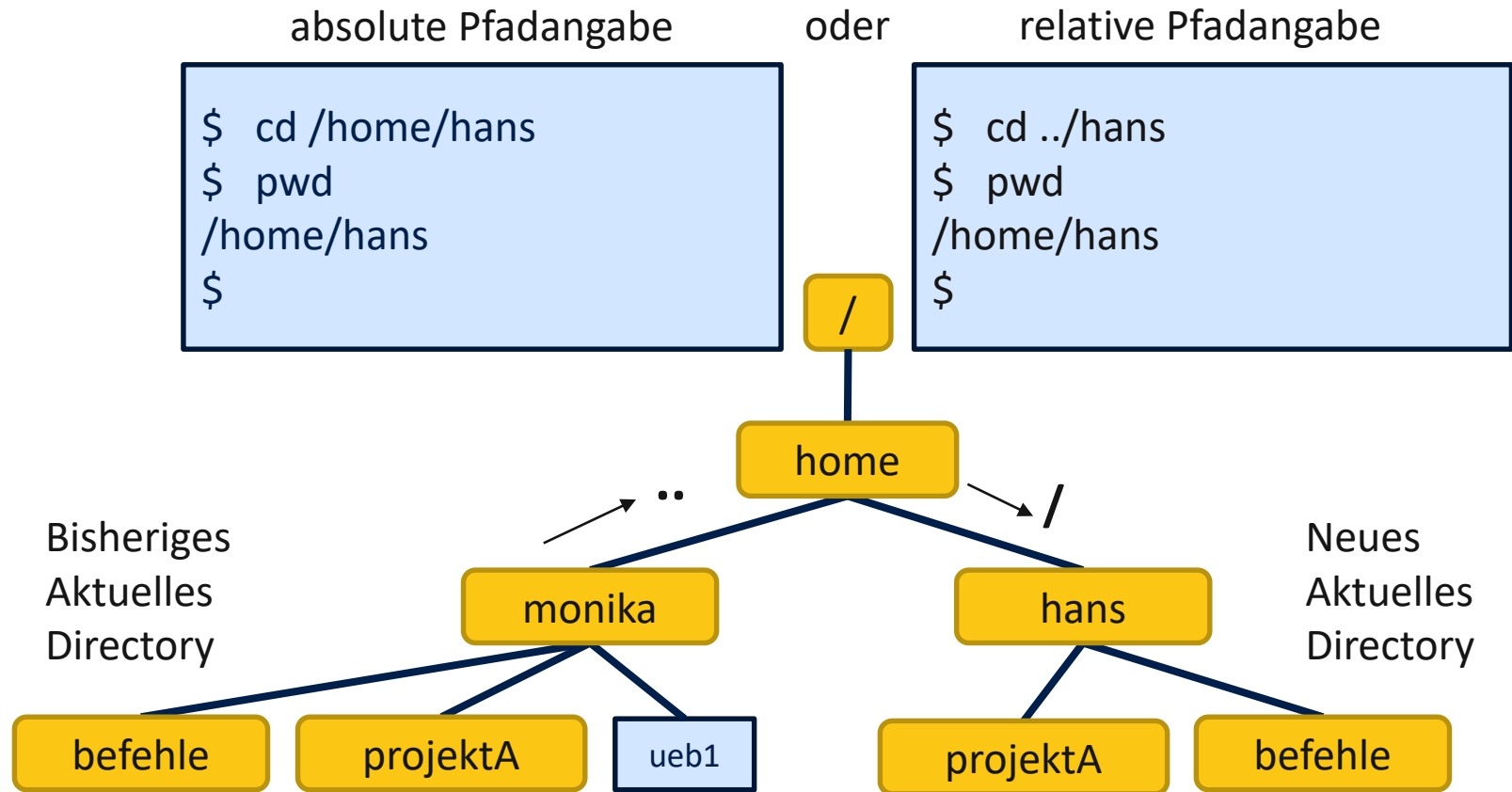
Bei der Angabe des relativen Pfades wird die Datei immer relativ zum aktuellen Arbeitsverzeichnis des Prozesses angegeben.

Das ls-Kommando zeigt uns dabei nicht den Inhalt der Dateien, sondern listet nur die Namen der Dateien auf.

- Dateien können mit ihrem vollständigen Pfadnamen angesprochen werden, dem sog. Absoluten Pfadnamen (beginnend ab dem root-Directory)
- Oder vom jeweils aktuellen Directory aus mit dem relativen Namen.

```
sb@ub:~/test$ ls
befehle 'D*' daten Datum fehler inhalt inhalt2 neu projektA ueb1
sb@ub:~/test$
```

Dateisysteme



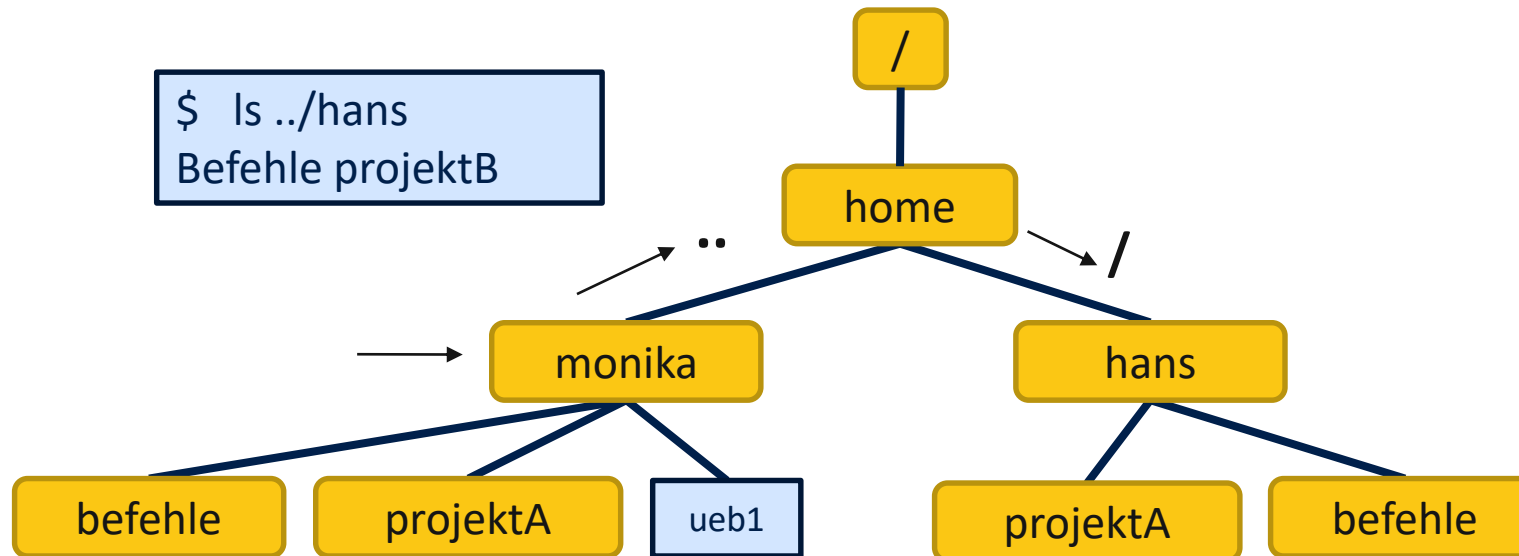
Hinweis:

Es gibt eine Reihe von Kürzeln für das einfache Adressieren von Pfaden. Folgende Tabelle zeigt die wichtigsten Kürzel:

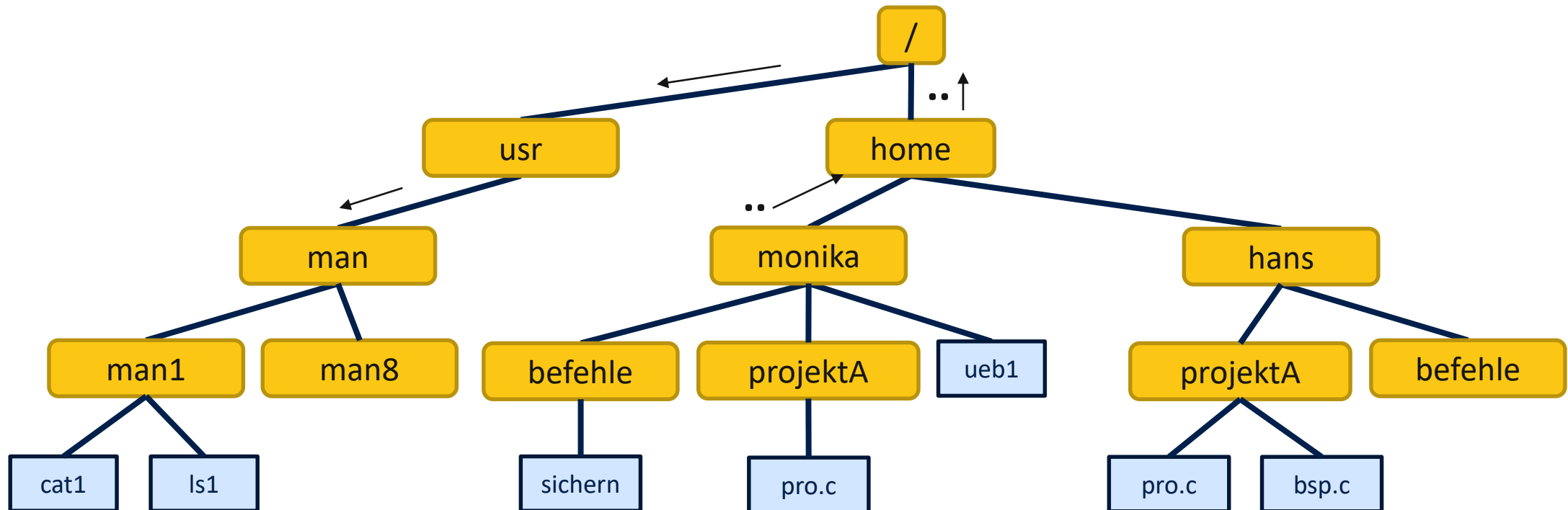
Kürzel	Beschreibung
.	Bezieht sich auf das aktuelle Arbeitsverzeichnis.
..	Bezieht sich auf das nächst höhere Verzeichnis, ausgehend vom aktuellen Arbeitsverzeichnis.
~	Bezieht sich auf das Homeverzeichnis des angemeldeten Benutzers.
-	Bezieht sich auf das Verzeichnis, welches als letztes besucht wurde.

Dateisysteme

- Bei langen Pfadnamen kann man sich leicht vertippen und muss viel schreiben. Deshalb stehen unter der Shell folgende Sonderzeichen zur Verfügung, um vom aktuellen Directory die umliegenden Dateien und Directories relativ ansprechen zu können:
 - Ein Punkt . bedeutet das aktuelle Directory, in unserem Fall **/home/monika**
 - Zwei Punkte .. bedeuten das darüberliegende Directory, hier **/home**.



- Beispiel: Relative (`../usr/man`) und absolute Pfadangabe (`/usr/man`)



Reguläre Dateien (files)

sind binäre, ausführbare oder einfache Textdateien. Diese verwenden normale Datenblöcke im darunterliegende Dateisystem und belegen somit Speicherplatz auf dem entsprechenden Medium.

```
sb@ub:~/test$ ls -l /bin/
insgesamt 12480
-rwxr-xr-x 1 root root 1113504 Apr  4  2018 bash
-rwxr-xr-x 1 root root  748968 Aug 29  2018 brlty
-rwxr-xr-x 3 root root   34888 Jul  4  2019 bunzip2
-rwxr-xr-x 1 root root 2062296 Mär  6  2019 busybox
-rwxr-xr-x 3 root root   34888 Jul  4  2019 bzip2
lrwxrwxrwx 1 root root      6 Jul  4  2019 bzip2 -> bzip2diff
-rwxr-xr-x 1 root root   2140 Jul  4  2019 bzip2diff
```

Verzeichnisse (directories)

sind spezielle Dateien, welche weiter Dateien enthalten können. Diese belegen ebenfalls Speicherplatz.

```
sb@ub:~$ ls -ld /home/*
drwxr-xr-x  4 hans      hans      4096 Apr 16 13:44 /home/hans
drwxr-xr-x 20 sb        sb        4096 Apr 17 15:25 /home/sb
drwxr-xr-x  2 testuser  testuser  4096 Apr 17 08:36 /home/testuser
drwxr-xr-x  4 tux      tux      4096 Apr 16 13:45 /home/tux
sb@ub:~$
```

Device Dateien (device files)

sind spezielle Dateien, welche den Anwendungen ermöglichen mit einem Gerät zu kommunizieren, repektive mit dessen Gerätetreiber. Dabei wird zwischen block und character device files unterschieden, welche über die major number mit dem Treiber in Verbindung gebracht werden. Das Gerät wird über die minor number identifiziert.

```
sb@ub:~$ ls -l /dev/sd*  
brw-rw---- 1 root disk 8, 0 Apr 17 08:25 /dev/sda  
brw-rw---- 1 root disk 8, 1 Apr 17 08:25 /dev/sda1  
brw-rw---- 1 root disk 8, 2 Apr 17 08:25 /dev/sda2  
sb@ub:~$
```

Sockets

sind spezielle Dateien, welche für Kommunikationsanwendungen verwendet werden. TCP/IP Sockets werden für Internetverbindungen verwendet. Anwendungen, welche nur lokal kommunizieren könnten auch UNIX domain sockets verwenden.

```
sb@ub:~$ netstat -lx
Aktive Sockets in der UNIX-Domäne (Nur Server)
Proto RefCnt Flags      Type       State      I-Node    Pfad
unix  2      [ ACC ]     STREAM    HÖRT       14534     /run/systemd/journal/stdout
unix  2      [ ACC ]     STREAM    HÖRT       31891     @/tmp/.ICE-unix/1675
unix  2      [ ACC ]     SEQPAKET  HÖRT       12219     /run/udev/control
unix  2      [ ACC ]     STREAM    HÖRT       30431     /run/user/1000/systemd/private
unix  2      [ ACC ]     STREAM    HÖRT       24011     /run/user/121/systemd/private
unix  2      [ ACC ]     STREAM    HÖRT       30435     /run/user/1000/gnupg/S.gpg-agent.ssh
unix  2      [ ACC ]     STREAM    HÖRT       24015     /run/user/121/gnupg/S.gpg-agent
unix  2      [ ACC ]     STREAM    HÖRT       30436     /run/user/1000/gnupg/S.dirmngr
```

Named pipes

sind spezielle Dateien, welche einen unidirektionalen Datentransfer zwischen Anwendungen erlauben.

```
sb@ub:~$ mkfifo fifo; ls -l fifo
prw-rw-r-- 1 sb sb 0 Apr 17 16:03 fifo
sb@ub:~$
sb@ub:~$ echo "Hello World" > fifo&
[1] 14133
sb@ub:~$
sb@ub:~$ cat fifo
Hello World
[1]+  Fertig                  echo "Hello World" > fifo
sb@ub:~$
```

Links

sind spezielle Dateien, welche auf andere Dateien verweisen.

Dabei wird zwischen symbolischen links und sogenannten „hard“ links unterschieden.

```
sb@ub:~/test$ ls -ilt
insgesamt 20
3933081 -rw-rw-r-- 1 sb sb    0 Apr 17 16:06 hart
3933080 lrwxrwxrwx 1 sb sb    4 Apr 17 16:05 symbol -> echt
3933079 -rw-rw-r-- 1 sb sb    0 Apr 17 16:05 echt
3932489 drwxrwxr-x 5 sb sb 4096 Apr 17 15:51 daten
3932335 -rw-r--r-- 1 sb sb    49 Apr 16 15:16 fehler
3932341 -rw-r--r-- 1 sb sb   478 Apr 16 15:06 inhalt
3932338 -rw-r--r-- 1 sb sb    26 Apr 16 13:31 ueb1
```

Softlinks:

Bei einem Softlink handelt es sich um eine Referenz auf ein Verzeichnis bzw. auf eine Datei. Wenn wir diese Datei verschieben oder löschen, ist der Softlink nicht mehr zu gebrauchen und produziert einen Fehler.

```
sb@ub:~/test$ # Syntax: ln -s Zieldatei Linkdatei
sb@ub:~/test$ ln -s /home/sb/test/text.txt /home/sb/test/unterverzeichnis/text.txt
sb@ub:~/test$
sb@ub:~/test$ ls -l unterverzeichnis/
insgesamt 0
lrwxrwxrwx 1 sb sb 22 Apr 17 16:10 text.txt -> /home/sb/test/text.txt
sb@ub:~/test$
```

Das Ganze ist in etwa vergleichbar mit einer Verknüpfung unter Windows.

Hardlinks:

Beim Hardlink sei als erstes erwähnt, dass man diesen nur bei Dateien Verwenden kann. Dies liegt daran, dass ein Hardlink die Datei direkt referenziert. Wenn wir diese Datei verschieben oder sogar löschen, bleibt diese trotzdem erreichbar.

```
sb@ub:~/test$ # Syntax: ln Zieldatei Linkdatei
sb@ub:~/test$ ln /home/sb/test/text.txt /home/sb/test/unterverzeichnis/text2.txt
sb@ub:~/test$
sb@ub:~/test$ ls -l unterverzeichnis/
insgesamt 0
-rw-rw-r-- 2 sb sb  0 Apr 17 16:08 text2.txt
lrwxrwxrwx 1 sb sb 22 Apr 17 16:10 text.txt -> /home/sb/test/text.txt
sb@ub:~/test$
```

Ist die Zieldatei gelöscht, dann bleibt der Inhalt der Datei trotzdem erhalten und ist noch lesbar.

Standardisierte Dateisystem Hierarchie (FHS)

Damit die Systemadministratorin sowie die Benutzerinnen einfacher die angebotenen Ressourcen finden, wird ein plausibles Dateisystem Layout benötigt.

- Die meisten GNU/Linux Distributionen sowie kommerzielle UNIX Anbieter implementieren heute mehr oder weniger den Filesystem Hierarchy Standard (FHS). Dieser basiert auf UNIX System V Release 4.
- <http://www.pathname.com/fhs/>

Shareable

Daten, die von mehreren Hosts gleichzeitig genutzt werden können. Dazu gehören Benutzerdaten, Emails, Textdateien, und Anwendungen.

- /home
- /var/mail
- /usr /opt

Unshareable

Daten, die nur für einen bestimmten Host von Bedeutung sind. Dazu gehören die Konfigurationen, Kernel Images und Prozesszustände.

- /etc
- /boot
- /var/run
- /var/lock

Das Root - Dateisystem „/“

Die Spitze des Dateisystems, besitzt kein übergeordnetes Verzeichnis. Das Root - Dateisystem muss alle nötigen Daten enthalten, damit ein System gebootet, wiederhergestellt oder repariert werden kann.

- **/usr**, **/opt** und **/var** sind so ausgelegt, dass diese auf anderen Dateisystemen liegen dürften als das Root - Dateisystem.
- Das Root - Dateisystem soll so klein als möglich sein. (kleines Medium, Netz-PC, Disk-Fehler)
- Die Verzeichnisse **bin/**, **boot/**, **dev/**, **etc/**, **lib/**, **mnt/**, **opt/**, **sbin/**, **tmp/**, **usr/**, **var/** sind erforderlich, **home/**, **lib/** und **root/** sind optional.

Das Boot - Verzeichnis „/boot“

Statische Dateien des Boot Loaders, inkl. Kernel. Muss bei Systemstart vorhanden sein;
Enthält zum Booten benötigte Dateien. Beispiele:

■ Kernel:

- "vmlinuz-versionsnummer-generic" (für Physikalische Systeme)
- "vmlinuz-versionsnummer-virtual" (für virtuelle Maschinen)

■ initiale Ramdisk

- "initrd.img-versionsnummer-generic"/"-server"/"-virtual,,

■ Das Programm für den Memorytest

- memtest86.bin

■ Unterverzeichnis(se):

- /boot/grub: Enthält Dateien der Bootloader GRUB 2 und GRUB
- /boot/efi: Enthält EFI-Programme - nur bei eingesetztem und aktiv genutztem EFI

Das Device File - Verzeichnis „/dev“

Das /dev Verzeichnis beinhaltet die Gerätedateien, welche die Verbindung zur Hardware darstellen.

- Die Gerätetreiber sind nicht in diesem Verzeichnis, normalerweise sollten sich diese in **/lib/modules/kernel-version** befinden.
- Von: devices (Geräte); muss bei Systemstart vorhanden sein; enthält alle Gerätedateien, über die die Hardware im Betrieb angesprochen wird
- Gerätedateien für Hot-Plug-Systeme (z.B. USB, IEEE1394) werden von udev erstellt

Das Konfigurations - Verzeichnis „/etc“

Von: et cetera ("alles übrige"), muss bei Systemstart vorhanden sein; enthält Konfigurations- und Informationsdateien des Basissystems.

Beispiele:

fstab, hosts, lsb-release, blkid.tab; hier liegende Konfigurationsdateien können durch gleichnamige Konfigurationsdateien im Homeverzeichnis überlagert werden. Beispiel: bash.bashrc

■ Unterverzeichnisse u.a.:

- **/etc/grub:** Enthält Skripte für Grub
- **/etc/opt:** Verzeichnisse und Konfigurationsdateien für Programme in /opt
- **/etc/X11:** Verzeichnisse und Konfigurationsdateien des XServer; Beispiel: xorg.conf
- **/etc/network:** Verzeichnisse und Konfigurationsdateien des Netzwerkes: Beispiel interfaces

Das Home-Verzeichnis /home

Das Homeverzeichnis ist das Verzeichnis des jeweiligen Benutzers innerhalb der Home-Partition. Heißt der Benutzer also z. B. otto, so heißt das Verzeichnis logischerweise **/home/otto**.

- Üblicherweise wählt man für **/home** eine eigene Partition. So gehen Einstellungen und persönliche Dateien bei einer Neuinstallation nicht verloren bzw. müssen vorher nicht aufwändig gesichert werden.
- Von der Größe sollte man **/home** so groß wie möglich erstellen. Hier werden ja alle privaten Daten abgelegt.

Das Bibliotheks - Verzeichnis „/lib“

Von: `libraries` (Bibliotheken); muss bei Systemstart vorhanden sein; enthält unverzichtbare Bibliotheken fürs Booten und die dynamisch gelinkten Programme des Basissystems; mindestens notwendige Dateien bzw. symbolische Verknüpfungen: `libc.so.*` (C-Bibliotheken), `ld*` (Bibliotheksdaten: Versionen, Zugriffspfade,...)

■ Unterverzeichnisse u.a.:

- `/lib/modules`: Kernelmodule
- `/lib/udev`: Bibliotheken und Programme für udev
- `/lib/linux-restricted-modules`: Speicherort für eingeschränkte Treiber (z.B. Grafikkarte)

Das Verzeichnis: **lost+found**

(verloren und gefunden), Dateien und Dateifragmente, die beim Versuch, ein defektes Dateisystem zu reparieren, übrig geblieben sind.

- Enthält nach einer automatischen Dateisystemreparatur unter Umständen noch Dateien, die gerettet werden können (ohne Garantie auf Erfolg!)
- Das Verzeichnis ist auf Partitionen mit **reiserfs** oder **xfs** nicht vorhanden

Das Verzeichnis `/media`

Für (Speicher-)Medien. Enthält Unterverzeichnisse, welche als mount- oder Einhängpunkte für transportable Medien wie z.B. externe Festplatten, USB-Sticks, CD-ROMs, DVDs und andere Datenträger dienen.

- Unterverzeichnisse sind u.a.:
 - `/media/usb`: Einhängpunkt für USB-Geräte
 - `/media/cdrom0`: Einhängpunkt für CD-ROMs
- Ab Ubuntu 12.10 wurde die Verwendung des Benutzernamens eingeführt.
 - `/media/BENUTZERNAME/usb`: Einhängpunkt für USB-Geräte
 - `/media/BENUTZERNAME/cdrom0`: Einhängpunkt für CD-ROMs

Das temporäre Mount Point „/mnt“

Von: mount (eingehängt); normalerweise leer; kann für temporär eingehängte Partitionen verwendet werden.

- Für Datenträger, die hier eingehängt werden, wird im Gegensatz zu `/media` kein Link auf dem Desktop angelegt (gilt nur für GNOME).

Das Verzeichnis /opt

Das Verzeichnis `/opt/` (für optional) ist ein Ordner, in dem sämtliche Zusatzprogramme installiert werden, die nicht unter einer freien Lizenz stehen oder bewusst vom Anwender an der Paketverwaltung vorbei geschleust werden sollen.

- Der Hauptvorteil gegenüber einer Installation im eigenen Homeverzeichnis ist die Möglichkeit, systemweit (durch alle auf dem Rechner vorhandenen Benutzer) auf die dort installierten Programme zugreifen zu können.

Das Verzeichnis /proc

Von: (laufende Programme); muss bei Systemstart vorhanden sein; enthält Processschnittstellen zum aktuell geladenen Kernel und seinen Prozeduren; Dateien lassen sich mittels cat auslesen

Beispiele: `version` (Kernelversion), `swaps` (Swapspeicherinformationen), `cpuinfo`, usw...

■ Unterverzeichnisse u.a.:

- `/proc/Nummern`: Jedes Programm mit Prozessnummer als Verzeichnisnamen; darin z.B. die Datei `status`, die Auskunft über den Prozess liefert (Programmname, Nummer, Speichernutzung,...)

Das Verzeichnis `/sbin`

Von: system binaries (Systemprogramme); muss bei Systemstart vorhanden sein; enthält alle Programme für essentielle Aufgaben der Systemverwaltung.

Programme können nur vom Systemadministrator (root) oder mit Superuserrechten ausgeführt werden.

Beispiele:

- **shutdown** (Herunterfahren)
- **make2fs**
- **lsmod** (Kernelmodule)

Das Verzeichnis /srv

Von: services (Dienste); Verzeichnisstruktur noch nicht genau spezifiziert; soll Daten der Dienste enthalten; unter Ubuntu in der Regel leer.

Das Verzeichnis /sys

/sys ist im eigentlichen Sinne kein Verzeichnis, sondern ein spezielles virtuelles Dateisystem vom Typ "sysfs", welches ebenso wie /proc lediglich Schnittstellen zu Kernelprozeduren beinhaltet.

Das Verzeichnis /tmp

Von: temporary (temporär); enthält temporäre Dateien von Programmen;

- Verzeichnis soll laut FHS beim Booten geleert werden.

Das Verzeichnis /usr

Von: Unix System Resources (früher User); enthält die meisten Systemtools, Bibliotheken und installierten Programme; der Name ist historisch bedingt - früher, als es /home noch nicht gab, befanden sich hier auch die Benutzerverzeichnisse

- „Als dann die Rollen der Benutzer und der Programmierer sich trennten, wurde irgendwann /home eingeführt, und irgendein Schlaumeier kam auf die Idee, /usr zukünftig mit "Unix System Resources" zu erklären“

Das Verzeichnis /var

Von variable; enthält nur Verzeichnisse; Dateien in den Verzeichnissen werden von den Programmen je nach Bedarf geändert (im Gegensatz zu /etc); Beispiele: Log-Dateien, Spielstände, Druckerwarteschlange

Unterverzeichnisse u.a.:

- `/var/log` : Alle Log-Dateien der Systemprogramme
- `/var/games` : Spielstände von Linux Spielen
- `/var/tmp` : Daten, die bei einem Reboot zwischengespeichert werden müssen
- `/var/www/html/` : Im FHS nicht vorhanden, bei Ubuntu das "document root" des Webserver Apache



**VIELEN DANK
FÜR IHRE
AUFMERKSAMKEIT!**