7. Direktories und Files unter Unix

pwd, cd, ls, mkdir, file, ...

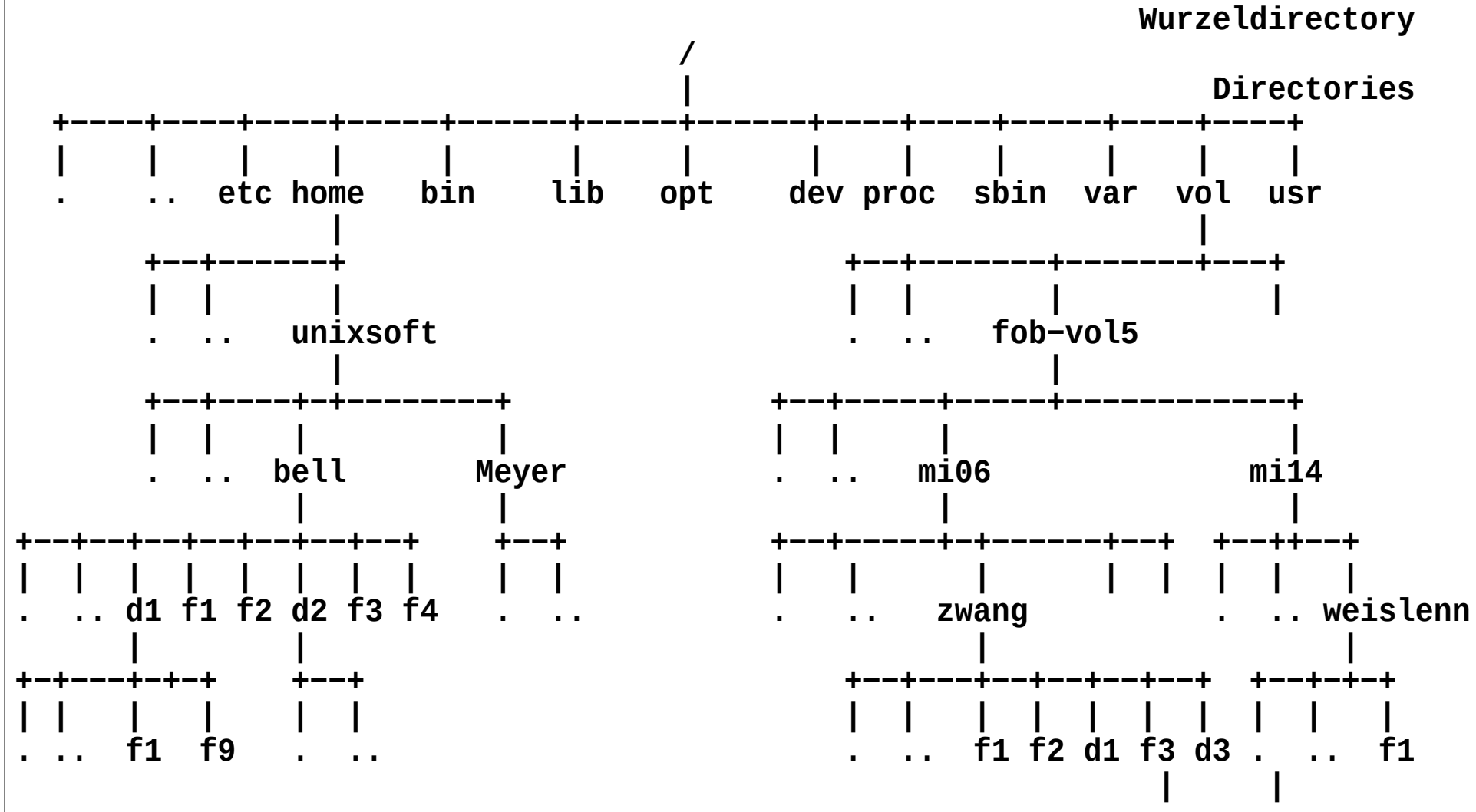
Was ist ein File?

File ist eine Folge von Bytes.
Es gibt "besondere Files" − für das Betriebssystem (Directories, Geräte, ...) und gewöhnliche Files − für den Nutzer.

- Gewöhnliche Files haben für das Betriebssystem keine Struktur.
  kein EOF, keine Zeilenstruktur, keine Blockstruktur.
- Besondere Files haben für das Betriebssystem eine Struktur.
- Files haben eine Länge.
- Auf Files kann sequentiell oder direkt (Adresse des Bytes) zugegriffen werden.
- Die Lage der Bytes eines Files auf den externen Medien ist für den Nutzer geheim.
- Files werden durch einen Namen referenziert. Der Name muß in einem Directory verzeichnet sein!!!!!
  Es gibt keine namenlose Files (Systemfehler!!!!).
- Files haben veränderbare Eigenschaften:
  Eigentumsrechte (Angabe des Besitzer):
  Eigentuemer
  Gruppe
  Zugriffsrechte:
  lesen, schreiben, ausführen
  unterteilt nach
  Eigentümer des Files
  Gruppe des Files
  den Rest der Welt (other)
  also 9 Zugriffsklassen

Die Files werden hierarchisch angordnet.
Die Wurzel allen Übels ist das Wurzeldirectory − "/".
In einem Directory können Directories und Files
enthalten sein − bunt gemischt.

Dateibaum des UNIX:



Spezielle Directory−Namen:

- **.** − **aktuelles Directory**
- **..** − **übergeorgnetes Directory**

Jeder Prozess (Kommando) "steht" in einem Directory − genannt
"Working Directory"
Aktuelles Directory
current directory
Nach dem Login (1.Prozess) steht man in seinem Homedirectory.

Namensbildung:

absolute Filenamen:
Beginnen mit einem "/" (Wurzelverzeichnis)
enthalten die Namen aller Directories bis zum konkreten File
z.B. /vol/fob−vol5/mi14/weislenn/d1/f1
/vol/fob−vol5/mi14/../mi06/d1/f1

relativer Filename:
Geht vom "Working Directory" aus.
z.B. "Working Directory": /vol/fob−vol4/mi14
d1/f1
./d1/f1
f1
./f1
../mi06

Die Folge der Namen der Directories, die zu einem File führt wird als "Path" (Pfad) bezeichnet.

Elementare Kommandos für Direktories

Bestimmen des "Working Directory"

Kommando:

pwd − print name of current/working directory

Synctax:

```
pwd
```

Beschreibung:

Ausgabe des vollen Filenamens (beginnend mit "/") des aktuellen Directories.

Beispiele:

```
pwd
```

Ändern des "Working Directory"

Kommando:

cd,chdir − change working directory

Syntax:

```
cd    <Directory−Path>
chdir <Directory−Path>
```

Beschreibung:

Ändern des Working Directories. Das Working Directory wird für den aktuellen Prozess (Shell) auf das angegebenen Directory gesetzt.

Voraussetzung:

- Das angegebenen Directory exisiert
- Der Prozess hat die notwendigen Zugriffsrechte (Ausführen)

Der Directory−Path kann absolute oder relativ angegeben werden.

Beispiele:

```
cd /tmp/D1
cd ..
cd /tmp/D2
```

Ich will wissen, welche Files in meinem Directory enthalten sind.

Directory besichtigen. Eigenschaften von Files feststellen.

Kommando:

ls − list directory contents

Syntax:

ls [OPTION]... [FILE]...

Beschreibung:

Ausgabe einer Liste mit Informationen über die Files des aktuellen Directories (wenn nicht anders angegeben). Die Liste ist alphabetische sortiert (wenn nicht anders gefordert). ist ein Filename spezifiziert worden wird die Ausgabe auf dieses File beschränkt. Ist der Filename ein Directory−Name, wird der Inhalt der Directory ausgegeben.

Optionen:

−−help

Die wichtigsten Optionen

!	−a, −−all	Einträge, die mit . beginnen, nicht verstecken
	−A, −−almost−all	implizierte . und .. nicht anzeigen
	−−author	den Urheber jeder Datei ausgeben
	−B, −−ignore−backups	Einträge, die mit ~ enden, nicht ausgeben
!	−c	mit −lt: Sortieren nach und Anzeige von ctime (Zeit der letzten Veränderung der Datei−Status−informationen); mit −l: ctime anzeigen und nach Namen sortieren
	−C	Einträge mehrspaltig ausgeben
!	−d, −−directory	Verzeichnis−Einträge statt der Inhalte anzeigen, symbolische Verknüpfungen nicht verfolgen
	−f	nicht sortieren, −aU aktivieren, −lst deaktiviern
	−g	wie −l, aber Eigner nicht auflisten
	−G, −−no−group	Ausgabe von Gruppen−Informationen unterdrücken
	−h, −−human−readable	Ausgabe von Größen in menschenlesbarem Format (z.B. 1K 234M 2G)
	−−si	wie −h, aber mit 1000 statt 1024 als Teiler
!	−i, −−inode	Ausgabe der INode−Nummer.
!	−l	Lange Listenformat verwenden.
	−L, −−dereference	Bei symbolischen Verknüpfungen die Eigenschaften der jeweiligen Zieldatei anzeigen.
	−n, −−numeric−uid−gid	Wie −l, aber numerische UIDs und GIDs anzeigen.
	−N, −−literal	Rohe Eintragsnamen anzeigen (z. B. Kontroll−zeichen nicht besonders behandeln).
	−o	Wie −l, aber ohne Gruppen−Informationen.
	−p, −−file−type	Anhängen eines Zeichens zur Typisierung jedes

```
                                Eintrags (eines aus »/=@|«).
!   -r, --reverse               Umgekehrte Reihenfolge beim Sortieren.
    -R, --recursive             Unterverzeichnissen rekursive ausgeben.
!   -s, --size                  Größe jeder Datei in Blöcken ausgeben.
    -S                          Nach Dateigröße sortieren.
!   -t                          Nach Änderungszeit sortieren.
    -T, --tabsize=SPALTEN       Tabstops auf alle SPALTEN Zeichen setzen
                                statt 8.
    -u                          mit -lt: Sortieren nach und Anzeige von
                                Zugriffszeit. Mit -l: Anzeige von Zugriffszeit
                                und sortieren nach Namen. Sonst:
                                Sortieren nach Zugriffszeit.
    -U                          Nicht sortieren; Einträge in Reihenfolge des
                                Verzeichnisses auflisten.
    -x                          Einträge in Zeilen statt in Spalten auflisten.
    -X                          Alphabetisch nach der Erweiterung sortieren.
    -1                          Eine Datei pro Zeile auflisten.
        --help                  diese Hilfe anzeigen und beenden.
        --version               Versionsinformation anzeigen und beenden.


    Wichtige Kombinationen von Optionen:

          -a        -   zeigt alle Filenamen an (einschließlich
                        Punkt-Files)
          -lisa     -   zeigt alles an
          -lisac    -   zeigt alles an, letzte Modifikationszeit
                        wird als Zeitangabe benutzt
          -lisact   -   wie -lisac aber nach Modifikationszeit
                        sortiert
```

Beispiele:

```
ls                  # Files
ls −a               # Files + Punktfiles
ls −l               # Files + Informationen
ls −la              # Files + Punktfiles + Informationen
ls −lac             #   ... + Modifikationszeit
ls −lact            #   ... + Sortierung nach Zeit
ls −lisa            # Alles mit Erzeugungszeit
ls −lisat           #   ... + Sortierung nach Zeit
ls −lisact          # Alles mit Modifikationszeit
ls −g               # wie −l ohne Nutzername
ls −gG              # wie −l ohne Nutzername und Gruppenname
ls −gGct            # ... + Modifikationszeit und Sortierung
ls −lisau           # Alles mit Accesstime
```

Erzeugen von Directories

Kommando:

mkdir − make directories

Syntax:

```
mkdir [OPTION] DIRECTORY...
```

Beschreibung:

Erzeugen des angegebenen Directories. Der Directory−Name kann als relativer oder absoluter Filename angegeben werden Das Directory darf noch nicht existieren. Der Nutzer muß Schreibrechte in dem Directory besitzen, in dem das Directory angelegt werden soll

Optionen:

−m, −−mode=MODUS	Zugriffsrechte setzen (wie chmod), nicht rwxrwxrwx − umask.
−p, −−parents	kein Fehler, wenn vorhanden; übergeordnete Verzeichnissen erzeugen, wenn notwendig
−v, −−verbose	für jedes angelegte Verzeichnis eine Meldung ausgeben
−−help	diese Hilfe anzeigen und beenden.
−−version	Versionsinformation anzeigen und beenden.

Beispiele:

```
cd /tmp
mdkir x1
mkdir x2
mkdir x3/x4
mkdir −p x3/x4
mkdir −pv x5/x4
mkdir −m 111 x6
       kleine Vorschau:
          Zugriffsrechte:
                 1 −  ausführen
                 2 −  schreiben
                 4 −  lesen
mkdir −−help
mkdir −−version
```

Welchen Inhalt haben meine Files?

Kommando:

file − Bestimmen des Filetypes

Syntax:

```
file  [  −bcikLnNsvz  ] [ −f namefile ] [ −F separator ]
                      [ −m magicfiles ] file ...
file −C [ −m magicfile ]      !!!! nur für Profis
```

Beschreibung:

Das Programm "file" testet jedes File und versucht das File zu klassifizieren. Folgende Tests werden durchgeführt:

- Filesystem−Test − Directory ermitteln
- Magic−Number−Test − Filetype bestimmen
  ASCII, ps, pdf, ..., ausführbares Programm (eventuell Architektur)
  Type−File: /usr/share/misc/magic
- Sprachtest − Programmiersprache (C, perl,...), natürliche Sprache (deutsch, englisch) bestimmen.

```
Optionen:

        -b, --brief      - nur Ergebnisausgabe, ohne Filename
        -c, --checking-printout
                         - zusammen -m für Debugging
                           !!!nur für Profis
        -C, --compile    - Erzeugen eines neuen Magic-Files
                           !!!nur für Profis
        -f, --files-from <namefile>
                         - Filename werden vom File <namefile>
                          gelesen. Ein Name pro Zeile.
        -F, --separator <separator-Zeichen>
                         - Trennzeichen zwischen Filename und
                          Ergebnis für die Ausgabe.
                          Standard ":"
        -i, --mime       - Mime-Type-Ausgabe
        -L, --dereference
                         - Symbolischen Links folgen
        -m, --magic-file <m1{,m2,...}>
                         - Liste von alternativen Magic-Number-Files
        -n, --no-buffer  - keine Pufferung der Ausgabe
        -N, --no-pad     - Keine Formatierung
        -s, --special-files
                         - Lesen von Special-Files
        -v, --version    - Version
        -z, --uncompress - uncompress versuchen
        --help           - Help ausgeben
```

Beispiele:

```
cd /tmp
file x3
file −n x3          # ohne Pufferung
file −N x3          # ohne Formatierung
file −N −F = x3     # "=" als Trennzeichen
file −s x3          # special Files lesen
file  y3            # Links nicht folgen
file −L y3          # Links folgen
file −b *           # ohne Filenamen
          !!!Leistung der Shell:
                    "*" bedeutet: alle Filenamen des akutellen
                                  Directories
```

Kommando:

touch − Erzeugen eines leeren Files, modifizieren der Zugriffszeit

Syntax:

```
touch [-a] [-c] [-d date] [-m] [-r referenzfile]
      [-t [[CC]YY]MMDDhhmm[.ss]][--help][--version]
       <filename> {<filename>}
```

Beschreibung:

Erzeugen eines leeren Files, falls dieses nicht vorhanden ist.
Setzen der Zugriffszeit.

Optionen:

−a	nur die Zugriffszeit des Files ändern
−c	File nicht erzeugen
−d date	benutze "date" als Zeitangabe
−m	nur Modifikationszeit des Files ändern
−r referenzfile	benutze Zeitangaben des Referenzfiles als Zeit
−t	benutze explizite Zeitangabe
−−help	Hilfe
−−version	Versionsangabe

Beispiele:

```
touch xxx
touch yyy zzz tttt
touch −a *
touch −a −c xxx yyyy cccc
```