

Rechnerorganisation

Martin Steffen

Lehrstuhl für Softwaretechnologie

Vorkurs Informatik

Wintersemester 2004/05



Funktionsweise und Aufbau von Rechnern

Dateisystem

Intermezzo

Betriebssystem

Dateisystem (II)

Interaktion

Geschichte des Computers

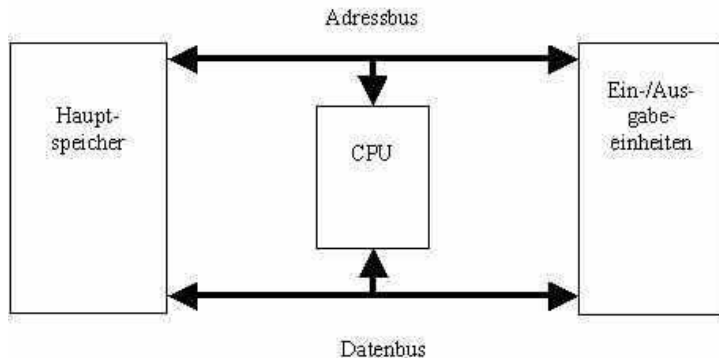
- lange Vorgeschichte von Rechenhilfen, aber keine “Computer” im modernen Sinne
- **Charles Babbage** (1791-1871): *Differential Engine*, *Analytical Engine* (nicht realisiert). **Ada Lovelace**: Gedankenspielerien zu “Software”
- **Konrad Zuse**: 1936 **Z1**: mechanisches, *programmgesteuertes* Rechnengerät
- **Howard H. Aiken**: Mark I (1944): elektromechanischer Rechner
- 1946: **ENIAC**: Erster „elektronischer“ Rechner (Röhren)
- **John von Neumann**: 1946: Prinzip des frei programmierbaren Rechners
- **Alan Turing**: Theoretische Grundlagen (**Turingmaschine**), Rechenmaschinen zum Entschlüsseln
- ...

Prinzipien der von Neumann-Architektur

Prinzipien des Rechners: bereits früh formuliert: **John von Neumann**:

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| Ein Computer besteht aus Steuerwerk , Rechenwerk , Speicher , und den Ein-/Ausgabegeräten | Steuerwerk = CPU . Rechenwerk = ALU , Coprozessoren , verschiedene Speicherebenen . |
| Die Struktur des Computers muß unabhängig vom bearbeiteten Problem sein | |
| Im Speicher befinden sich Daten + Programm | Heute meist mehrere Programme gleichzeitig |
| Der Speicher ist in Zellen gleicher Größe unterteilt und fortlaufend numeriert | im wesentl. unverändert (virtuelle Adressierung) |
| Das Programm besteht aus einer Folge von Befehlen, die nacheinander aufgeführt werden. Die Ausführung eines Sprunges kann von gespeicherten Werten abhängig gemacht werden | |
| Die Maschine benutzt Binärcodes. | |

Schematische Architektur



CPU

- CPU
 - **Central processing unit**: “Prozessor”= da wo die Arbeit gemacht wird
 - Integrierte Schaltungen = “Chip”¹
 - **Taktrate** = Anzahl der **elementaren Operationen** pro Sekunde (gemessen in **Hz**, heutzutage eher MHz und bereits GHz)
- Bekannte Hersteller: **Intel**, **AMD**
- Beachte: Taktrate bestimmt nicht alleine die Geschwindigkeit des Rechners
- Spezialprozessoren und **Ko-Prozessoren** für Spezialaufgaben

¹Es gibt natürlich auch Speicherchips.

Speicherhierarchie (abstrakt)

- Speicher = Hardware zum **Aufbewahren** von **Daten**
- Es gibt nicht einen monolithischen Speicher, sondern eine **Hierarchie**, unterschieden nach
 - Speicherkapazität
 - Geschwindigkeit
 - Preis
 - „Nähe“ zur CPU
 - **flüchtig** oder **dauerhaft**
 - Zugriffsarten
 - nur **lesbar** (ROM), les- und schreibbar
 - Wahlfreier Zugriff (**RAM**), sequentieller Zugriff (Magnetband)
 - **physikalische** Speicherprinzipien

Speicherhierarchie

- Speicherstufen, aufsteigend nach Größe:
 - **Register**: sehr klein/schnell, flüchtig
 - **Cache**:² flüchtig
 - **Hauptspeicher**: flüchtig, Speicher in dem der Prozessor die für das laufende Programm benötigten Daten hält (oder zumindest zu halten versucht . . .)
 - **Festplattenspeicher**: Permanent, enthält das **Dateisystem** (siehe Abschnitt III)
 - Hintergrundspeicher (zum Archivieren, d.h., permanent): Bänder, Zip, . . .
- Sonderspeicher:
 - Videospeicher (flüchtig),
 - **Floppy**: am Aussterben
 - CD-Rom, beschreibbare CD, Flash-Memory . . .

²manchmal mehrstufig

Bits und Bytes

- Datenmenge werden in **Bits**³ gemessen, beziehungsweise in **8er-Gruppen** von bits = **Byte**

1 bit = “*binary digit*” = 0 oder 1

Byte = 8 Bit

1 KByte = 1024 Bytes (ca. 1000 bytes)

1 Megabyte (MB) = 1 048 576 Bytes (ca. 1 Mio bytes)

1 Gigabyte (GB) = 1 073 741 824 Bytes

⋮

³Das **Binärsystem**, also das 2er System, ist schon recht alt, es wurde 1679 von Leibniz erfunden.

Speicherkapazitäten

| Medium | Kapazität (ca.) |
|-------------------------|-------------------------------|
| 1 Schreibmaschinenseite | 2 KB |
| Diskette ("Floppy") | 1 MB |
| Hauptspeicher | 500MB |
| Festplatte | 100 GB |
| Magnetband | 2 GB und mehr |
| Audio-CD/CD-Rom | ca. 1GByte |
| DVD | 20 Gbyte |
| menschl. Gehirn | 10^{15} Bytes (geschätzt..) |

Hard/Software im System

- **Hardware**: alles was man anfassen kann, **der Rest** = Software/die Programme⁴
- grob: folgende **Schichten** im einem Rechner

| | | |
|------------------------|---------------|---------------------|
| Anwenderprogramme | Fenstersystem | ... |
| Compiler | Editoren | Kommandointerpreter |
| Betriebssystem | | |
| Maschinensprache | | |
| Mikroprogrammirebene | | |
| Physikalische Hardware | | |

⁴Die Grenzen sind nicht ganz so eindeutig.

Funktionsweise und Aufbau von Rechnern

Dateisystem

Intermezzo

Betriebssystem

Dateisystem (II)

Interaktion

Dateien

- **Datei** (engl. *file*) = zusammenhängende Menge gespeicherter **Daten**
- mögliche **Inhalte**: Text, Programm, Programmcode, Musik, Bilder, ...
- Typ der Datei bestimmt **Bearbeitungsart** (z.B. Musikdateien wird man nicht anschauen, sondern abspielen/anhören)
- Typ: gekennzeichnet per **Konvention** durch bestimmte **Dateiextensionen**⁵
- z.B., `datei.txt` kennzeichnet eine **Textdatei**, weitere Beispiele folgen am Nachmittag
- Extensionen können **kombiniert** werden `datei.ps.gz`

⁵In Dateibrowsern in graphischen Oberflächen werden, gemäß den Extensionen, die verschiedenen Typen durch mehr oder minder selbsterklärende Bildchen/Icons symbolisiert.

Organisation der Dateien: der Verzeichnisbaum

- hierarchische Strukturierung als **Baum von Verzeichnissen**
- **Verzeichnis**:⁶ enthält Menge von **Dateien** und (Unter)-Verzeichnissen
- Verzeichnis \approx **Spezialform** einer Datei⁷
- **Pfad**:
 - “Weg” durch den Baum \Rightarrow **Lokalisierung** von Dateien/Verzeichnissen
 - Trenner der Verzeichnisse: durch **Schrägstrich** (“slash”):⁸ /
- **Wurzel** des Baumes (“root”-Verzeichnis), gekennzeichnet durch:

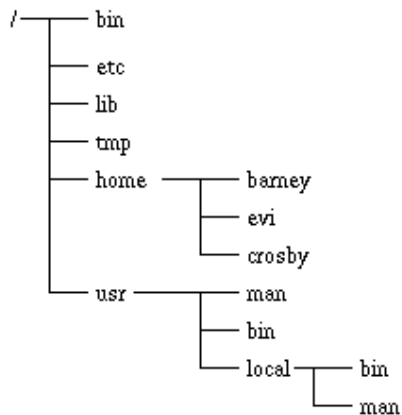
/

⁶Englisch: *directory*, beim Macintosh auch als Ordner bekannt.

⁷der Vollständigkeit halber: Un*x kennt vier prinzipielle (aus Sicht des Betriebssystems unterschiedliche) Klassen von Dateien: **normale** Dateien, **Verzeichnisse**, **Pipes** und **Devices**. Wenn wir von „Datei“ reden, werden wir im folgenden Datei im engeren Sinne meinen, keine Verzeichnisse etc.

⁸Bei DOS wird der umgekehrte Schrägstrich (“backslash”) zum selben Zweck verwendet.

Typische Struktur eine Verzeichnisbaumes



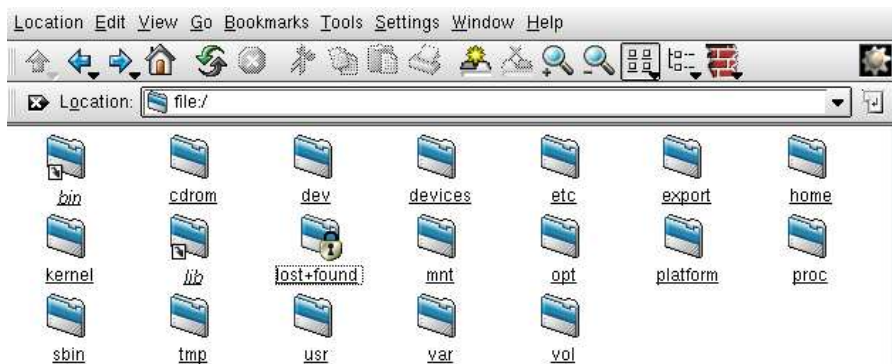
Je nach "Browser": verschiedene Sichten (1)

Location Edit View Go Bookmarks Tools Settings Window Help

Location: file:/

| Name | Size | File Type | Modified | Permissions | Owner | Group |
|------------|---------|-----------|------------------|-------------|---------|---------|
| bin | 12.5 KB | Directory | 2003-09-24 08:42 | rxwx-rx-x | root | bin |
| cdrom | 512 B | Directory | 2003-08-16 11:05 | rxwx-rx-x | root | nobody |
| dev | 3.5 KB | Directory | 2003-08-10 15:10 | rxwx-rx-x | root | sys |
| devices | 512 B | Directory | 2003-08-10 15:06 | rxwx-rx-x | root | sys |
| etc | 4.0 KB | Directory | 2003-08-10 15:22 | rxwx-rx-x | root | sys |
| export | 512 B | Directory | 2003-08-09 16:10 | rxwx-rx-x | root | sys |
| home | 13 B | Directory | 2003-10-08 14:21 | r-xr-xr-x | root | root |
| coqtool | 1,024 B | Directory | 2003-08-30 12:22 | rxwx-rx-x | coqtool | coqtool |
| esterel | 1,024 B | Directory | 2003-09-03 17:33 | rxwx-rx-x | esterel | esterel |
| info | 1,024 B | Directory | 2003-09-23 16:11 | rxwx-rx-x | info | www |
| java | 1.5 KB | Directory | 2003-09-10 13:47 | rxwx-rx-x | java | java |
| ms | 3.5 KB | Directory | 2003-10-08 14:27 | rxwx-rx-x | ms | mwpr |
| bin | 512 B | Directory | 2003-10-06 13:43 | rxwx-rx-x | ms | mwpr |
| components | 512 B | Directory | 2003-07-09 12:52 | rxwx-rx-x | ms | mwpr |
| cvsroot | 1,024 B | Directory | 2003-10-07 16:17 | rxwx-rx-x | ms | mwpr |
| dc | 512 B | Directory | 2002-08-07 08:53 | rxwx-rx-x | ms | mwpr |
| Desktop | 512 B | Directory | 2001-06-05 16:20 | rx----- | ms | mwpr |
| dotfiles | 512 B | Directory | 2003-06-27 07:48 | rxwx-rx-x | ms | mwpr |

Je nach "Browser": verschiedene Sichten (2)



Arbeitsverzeichnis

- **aktuelles** Verzeichnis (wo man “sich gerade befindet”):
Arbeitsverzeichnis (*working directory*)
- Default/nach dem **Einloggen**: = **Homeverzeichnis**
- Anzeigen des Arbeitsverzeichnisses mittels

```
pwd
```

Absolute und relative Pfadnamen

- 1. **absolut** = Pfad von der **Wurzel** aus
- 2. **relativ** = Pfad vom **Arbeitsverzeichnis** aus

- Spezialverzeichnisse:**

| Abkürzung | Bedeutung |
|-----------|-----------|
|-----------|-----------|

| | |
|---|----------------------------|
| . | „Hier“, Arbeitsverzeichnis |
|---|----------------------------|

| | |
|----|-----------------------------------------|
| .. | „eins hoch“, übergeordnetes Verzeichnis |
|----|-----------------------------------------|

| | |
|-----------|------------------------------------------------------|
| ~benutzer | Homeverzeichnis von benutzer (i.d.R. /home/benutzer) |
|-----------|------------------------------------------------------|

- 1. absolut

`/home/kurs001/texte/`

- 2. relativ (ohne / zu Beginn). Angenommen, man befindet sich in `/home/kurs001/Mail`, dann bezeichnet

`../texte/`

das absolute Verzeichnis `/home/kurs001/texte`

Einfache Orientierung & Navigation


- **Wo bin ich?**: `pwd`⁹ zeigt den absoluten Pfad des Arbeitsverzeichnisses
- **Inhalt** eines Verzeichnisses?: `ls <verzeichnis>`¹⁰
- Abkürzung: nur `ls`: listet das aktuelle **Arbeitsverzeichnis**
- **Wechsel** des Verzeichnisse: `cd`¹¹

`cd <verzeichnis>`

- Abkürzung: `cd = cd ~` = in's Homeverzeichnis wechseln

⁹*print working directory*

¹⁰`ls = list`. Es entspricht dem DOS-Kommando `dir`. Der Befehl hat eine Reihe von Optionen, mit der man weitere Informationen (Zugriffsrechte, Dateigröße, Besitzer ...) bekommen kann. Man kann den Befehl auch auf ein einzelne Dateien, nicht nur Verzeichnisse anwenden.

¹¹*change directory*. Genaue Verwendung bitte selbst nachschlagen. 

Veränderung der Dateistruktur

- Anlegen von neuen Dateien: meist mittels allgemeiner/spezieller Editoren
- Auf dem Übungszettel: ein paar einfache Möglichkeiten zur Dateierzeugung
- **Achtung!** Nach dem Löschen sind die Daten weg

Veränderung der Dateistruktur

| Befehl | Beispiel | Erklärung |
|--------------------|-----------------------------------------------------------------------------------|---------------------------------------------|
| <code>cp</code> | <pre>cp <datei1> <datei2> cp <datei1> <verzeichnis></pre> | Kopieren |
| <code>mv</code> | <pre>mv <datei1> <datei2> mv <datei1> <verzeichnis></pre> | <i>“move”</i> |
| <code>rm</code> | <pre>rm <datei></pre> | <i>“remove”</i> : Datei löschen |
| <code>rmdir</code> | <pre>rmdir <verzeichnis></pre> | <i>“remove directory”</i> : Verz. löschen |
| <code>mkdir</code> | <pre>mkdir <verzeichnis></pre> | <i>“make directory”</i> : neues Verzeichnis |

Funktionsweise und Aufbau von Rechnern

Dateisystem

Intermezzo

Betriebssystem

Dateisystem (II)

Interaktion

Das Wichtigste überhaupt: Hilfe im System

- primäre Hilfe zu einzelnen Befehlen: **Manual-Pages**:

`man` <command> oder `xman`

die *manpages* zeigen *kurz* die Verwendung des jeweiligen Befehl (Liste der **Optionen**) und oft ein paar **Standardbeispiele**


- **Info-System** im emacs oder xemacs¹²
 - Info-Information i.d.R. ausführlicher als *manpages*,
 - Manche (größere) Pakete verwenden Info-pages als primäre Nachschlagequelle
- **Internet**
 - bei **lokalen** Problemen: FAQs des Instituts

¹²Emacs/Xemacs steht erst übermorgen auf dem Kursprogramm. Die Info-Seiten bekommt man auch mit `xinfo`.

Diskettenbenutzung

- bequeme Sammlung von Programmen zur **Diskettenbenutzung**: **mtools**
- verwendet DOS-Dateiformat \Rightarrow brauchbar zum **Datentransport** zwischen¹³DOS \leftrightarrow Un*x

| | |
|---------------------------------|------------------------------------------------------|
| <code>mformat a:</code> | Floppy formatieren (d.h., insbesondere auch löschen) |
| <code>mcopy datei.txt a:</code> | auf Floppy kopieren |
| <code>mcopy a:* .</code> | alles von Floppy ins System kopieren |
| <code>mdel a:datei.txt</code> | Datei auf Floppy löschen |
| <code>mdir a:</code> | Verzeichnis (directory) anzeigen |
| <code>(eject)</code> | Auswurf der Diskette (oder CD-Rom) |

¹³Es gibt (neben unterschiedlichen Filesystemen) einen Unterschied zwischen Textdateien in beiden Systemen: die Darstellung der "**Neuen Zeile**". Unter Un*x erkennt man eine DOS-generierte Textdatei daran, daß lauter $\wedge M$ durch die Datei gesprengelt sind. `dos2unix` und `unix2dos` sind zwei Konvertierungskommandos, die man in dem Zusammenhang brauchen kann, falls einem der Editor die Angelegenheit nicht ohnehin ausbügeln hilft. Alternativ ist die **Option** `-t` von `mcopy` hilfreich. 

Drucken

- Arten von Drucker: Tintenstrahl/Laser
- **Leider:** im Kurs drucken nicht möglich, keine Übungen dazu . . .
- Voreingestellt pro Rechner (i.d.R): **Standarddrucker** (=der nächste erreichbare)
- Druckkommando: `lpr <datei>`¹⁴
- Druckbar: **Text** (=“Ascii”), **Postscript**
- viele Tools: **print-Button**
- **Vorsicht** bei **Folien**: falsche Kombination von Folien/Drucker kann den Drucker **beschädigen**

¹⁴lp ist eine Alternative

Funktionsweise und Aufbau von Rechnern

Dateisystem

Intermezzo

Betriebssystem

Dateisystem (II)

Interaktion

Aufgaben des Betriebssystems

- Ressourcenverwaltung
- Resource =
 - Zeit (Stichwort **Multitasking**, **Multiuserbetrieb**)
 - Speicherverbrauch (Festplatten, Hauptspeicher)
 - Zuteilung und Verwaltung von **I/O-Ressourcen** (Drucker, Tastatur, Bildschirm, Maus, (Daten-)Netzanschlüsse . . .)
- **Verbergen** der Hardware (Abstraktion) insbesondere **Dateisystem**
- **Schutz**
- Plattform/einheitliche Schnittstelle für **Anwenderprogramme**

Geschichte der Betriebssysteme

- bis ca. 1955: **keinerlei** Betriebssysteme
- 1955-65: **Batch**-Systeme (HW: Transistoren, Lochkarten/Magnetbändern als E/A)
- 65-80: **Multiprogramming** und **Time-sharing**-Betrieb (Interaktiv) auf Großrechnern (IBM360), Ursprung von Un*x: ca. 1970¹⁵
- Nach 1980:
 - Aufstieg des **PC**: **DOS**, **Windows95** und seine Nachfolger¹⁶
 - Verbreitung von **lokalen Netzen** (verteilte Betriebssysteme), **Internet**
- **Zukunft**: eingebettete Systeme? Netzcomputer? Grid-Computing ...

¹⁵Man schreibt oft Un*x anstelle Unix, denn Unix ist ein geschütztes Markenzeichen von Bell Corp. Mit Un*x meint man alle Unixe — oder Unices, wie die Lateiner sagten — z.B. Berkely, BSD, Solaris, System V, HPUX, und auch die verschiedenen Linux-Abkömmlinge.

¹⁶Natürlich gibt's auch noch MAC-OS, und OS2 &c

Multitasking/Multiuser/Timesharing

- Heutzutage: **Betriebssystem** arbeitet viele Dinge (scheinbar) **gleichzeitig** ab¹⁷
- **Illusion** der Gleichzeitigkeit durch schnelles/geschicktes **Hinundherjonglieren** der zu bearbeitenden Aufgaben (Multitasking)
- insbesondere langsam: die menschlichen Reaktionen (Tastatur, Maus) und überhaupt E/A: \Rightarrow **interaktives** Verhalten für mehrere Benutzer möglich
- Mehrbenutzerbetrieb: erhöhte Anforderungen an **Sicherheit** (z.B. Trennung der Speicherbereiche etc.)

¹⁷Nur scheinbar, wenn man nur *einen* Prozessor besitzt. Letzten Endes ist es (fast) nur eine von-Neumann Maschine, und von Neumann hat u.A. gesagt: **ein Befehl nach dem anderen**.

Funktionsweise und Aufbau von Rechnern

Dateisystem

Intermezzo

Betriebssystem

Dateisystem (II)

Interaktion

Un*x-Rechte

- drei Dinge, die man mit Dateien **machen** kann
 1. **Lesen** (**r** für *read*) Anschauen, kopieren
 2. **Schreiben** (**w** für *write*): Verändern, Überschreiben
 3. **Ausführen** (**x** für *execute*): Programme aufrufen/laufen lassen
- **Verzeichnis** = Sonderform einer Datei \Rightarrow
 - **Lesen**: Inhalt mittel `ls` anzeigen lassen
 - **Schreiben** = Verzeichnisstruktur **ändern** d.h., neue Dateien/Unterverzeichnisse **anlegen/löschen**
 - **Ausführen** = in das Verzeichnis **wechseln** (mittels `cd`)¹⁸

¹⁸Beachte: Lesen eines Verzeichnisses und in das Verzeichnis wechseln sind verschiedene Dinge.

Benutzerklassen und Un*x-Gruppen

- drei hierarchische **Klassen** von Benutzern
 1. **Benutzer** `u` *user*
 2. **Gruppe** `g` *group*
 3. **Alle** `o` *others*
- Jeder Benutzer gehört einer oder mehrerer **Gruppen** an, die seine **Privilegien** festlegen
- z.B. `user`, `student`, `mitarbeiter`, verschiedene administrative Gruppen
- Jede Datei/Verzeichnis hat **Zugriffsprivilegien** für Lesen/Schreiben/Ausführen getrennt für jede der drei Benutzer
- zwei wichtige Attribute einer Datei: **Besitzer** (*owner*) + **Gruppe**

Un*x-Gruppen (Forts.)

- **Ändern** mittels `chgrp` und `chown`¹⁹
- jeder Benutzer ist für das Setzen der Rechte selbst verantwortlich
- für Fortgeschrittene: Automatisierung mittels `umask`-Kommando
- **Anschauen** mittels `ls -lg` (`g` für „Gruppe“, `l` für „lang“, mit `ls -lag` bekommt man auch noch sog. **unsichtbare** Dateien angezeigt, `a` für „alle“)

¹⁹für *change group* und *change owner*

Un*x-Gruppen (Forts.)

```
[ms @ sokrates] > ls -lag
total 1023
drwxr-xr-x  47 ms      mwpr      2560 Oct  8 11:40 .
dr-xr-xr-x   9 root     root      9 Oct  8 13:44 ..
drwxrwxr-x  11 ms      mwpr      512 Jun  2 07:29 Lehrstuhl
drwx-----  5 ms      mwpr      4096 Oct  8 08:51 Mail
drwxr-xr-x   3 ms      mwpr      512 Aug 11 07:50 bin
drwxr-x----  8 ms      mwpr      512 Aug 17 08:59 papers
drwx----- 14 ms      mwpr      1024 Oct  4 09:38 priv
```

Un*x-Gruppen (Forts.)

oder Sicht mittels "Browser":



Dateiformate

| Extension | Kürzel | Erklärung | Viewer/Bearbeitungsprogramm |
|------------------|-------------------------|----------------------------------------------------------------------------------|-----------------------------|
| *.txt | ASCII | einfachstes Textformat | jeder Editor |
| *.html, *.htm | hypertext markup lang. | WWW-Dokumente | Browser/Editor |
| *.ps | Postscript | Dokumente, wird von Druckern verarbeitet | ghostview, gv |
| *.pdf | port. document format | Dokumente | acroread |
| *.tex | \LaTeX (oder \TeX) | | |
| *.dvi | “device independent” | Ergebnis eines \LaTeX -Textes | xdvi |
| *.jpg, *.gif | | zwei bekannte Bildformate, es gibt noch viele mehr | xv |
| *.fig | | Zeicheneditorformat | xfig |
| *.gz *.Z | | komprimierte Datei (mit Hilfe von <code>gzip</code> oder <code>compress</code>) | dekomprimieren: gunzip |
| *.tar | “tape archive” | Archiv | tar |

Wildcards

- Hilfreiche **Notation**:
- auf **Häufigsten** gebraucht: * = "Wildcard"= stellvertretend für beliebiges
- Zur **Bezeichnung** von Dateien
 - `ls a*`: Liste alle Dateien die mit a beginnen
 - `rm *.ps *.dvi` entferne alle postscript und alle dvi-Dateien

Funktionsweise und Aufbau von Rechnern

Dateisystem

Intermezzo

Betriebssystem

Dateisystem (II)

Interaktion

Kommandointerpreter: die Shell

- wichtige System-Schnittstelle: **Kommandointerpreter**, **Shell**: interaktive Befehlseingabe
- In X-Menues: meist als **Xterm** benannt²⁰
- es gibt eine Reihe von Unix-Shells:²¹ Bourne shell (der Urvater), C-Shell, tcshell, Korn shell. Bei uns *Bash* ("*bourne-again shell*")
- **Kommando**: Textueller Befehl an das System, allgemeines Format

```
<commando> <optionen> <file>
```

- **Optionen** sind oft **optional**, verfügbare Optionen hängen vom Befehl ab (die *manpages* geben jeweils Auskunft)

²⁰X-Terminal = ein Fensterrahmen um einen Kommandointerpreter herum

²¹Die Unterschiede sind für uns unerheblich. Der Benutzer merkt Unterschiede zwischen den Shells nur am Ausmaß der Bequemlichkeit. In diesem Sinne gibt es bequeme Shells, d.h. solche, bei denen man alte Befehle wiederholen kann und edieren etc., und unbequeme, bei denen man das nicht kann.

Umgebung

- Benutzerinteraktion unter bestimmten **Voreinstellungen** (Defaultdrucker? Lieblingseditor? Windowsystem? ...)

Umgebung (*environment*)

- Umgebung
 - von **Systemadministration** (hoffentlich) sinnvoll vorgelegt
 - von **Benutzer** anpassbar/erweiterbar
- Umgebungsvariable:
 - anzeigbar mittels `env`
 - verändern²² mittels `export <VARIABLE>=<WERT>`
- wichtige Umgebungsvariable: **PATH**

²²Umgebungsvariablen sind unabhängig von der Wahl der Shell, aber die Syntax `export .. = ..` ist bash-spezifisch.

Der Unix-PATH

- Annahme: ausführbare Datei ("Programm"):

```
/usr/local/games/bin/tetris
```

⇒ Eingabe von `/usr/local/games/bin/tetris` führt es aus

⇒ Auf Dauer unpraktisch ⇒ wichtige Umgebungsvariable `PATH`

- Aufruf eines Programmes in einem der dort erwähnten Verzeichnisse²³
Verzeichnisse: der Pfad kann weggelassen werden ⇒ Aufruf nur mit `tetris`
- welches Programm wird aufgerufen: das, welches mit passendem Namen als erstes im `PATH` gefunden wird (Ausprobieren `which <programm>`)

²³der Wert von `PATH` ist eine Liste von Verzeichnissen, sie sind durch ein `:` jeweils getrennt.

Sonstiges zur Shell-Anpassung (speziell Bash)

- bei gutverwalteten Systemen: vernünftige Voreinstellungen
- vom Benutzer **änderbar**
- dauerhafte Änderungen z.B. in `~benutzer/.bashrc`²⁴
- Nützliche, einfache Sache: sog. **alias**
- **Beispiel:** `alias zeigemiralles="ls -lag"`
- Falls man das **dauerhaft** will: einfügen der Zeile ins `~benutzer/.bashrc`

²⁴Viele Tools haben bestimmte Steuerdateien, oft **verborgene** Dateien, das sind die, die mit einem Punkt beginnen und die man mit einem einfachen `ls` nicht zu Gesicht bekommt. Oft (durchaus nicht immer) enden sie mit dem Kürzel `rc`, was für *resource* steht. Wen's interessiert kann sich das `.bashrc` anschauen.