

# Rechnerorganisation

MARTIN STEFFEN

Wintersemester 2003/04

# Abschnitt I

## Funktionsweise auf Aufbau von Rechnern

**Inhalt:** Geschichtliches · von Neumann Architektur · CPU · Speicher und Speichergrößen

# Geschichte des Computers

---

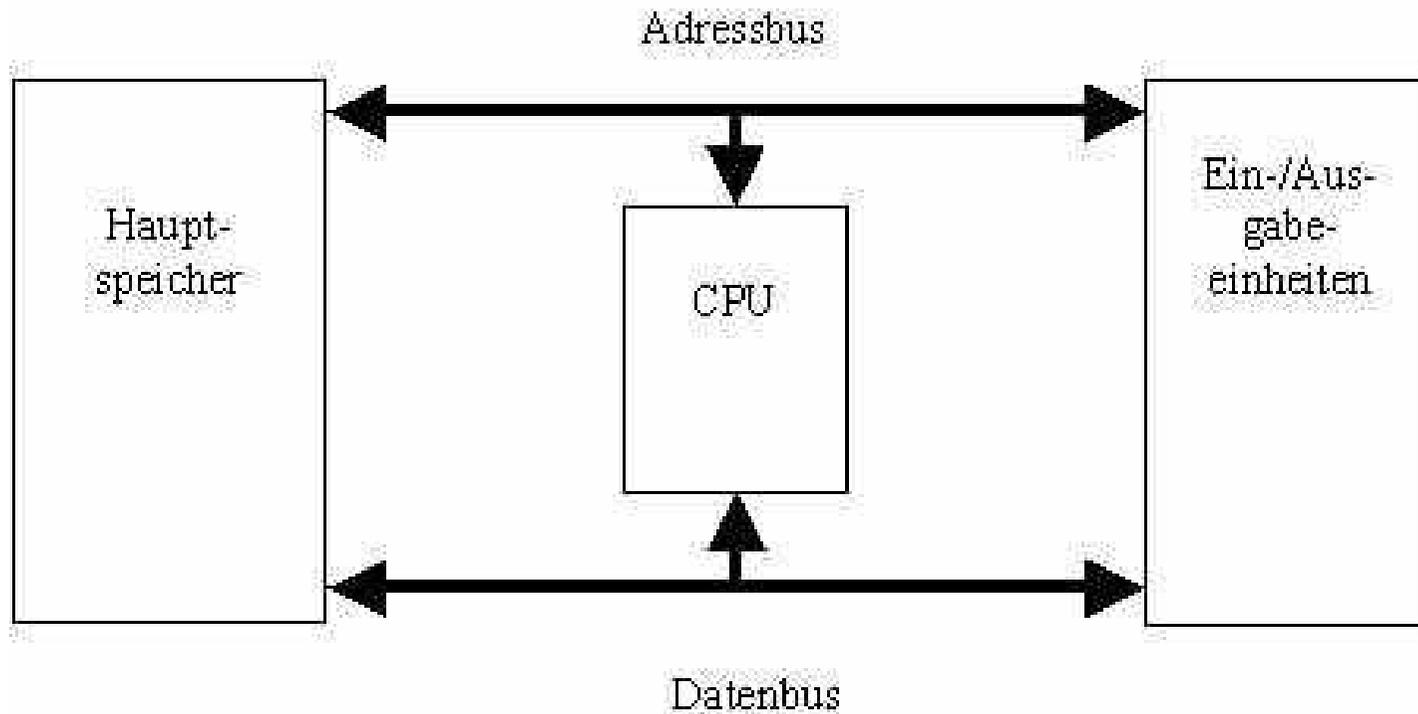
- lange Vorgeschichte von Rechenhilfen, aber keine “Computer” im modernen Sinne
- **Charles Babbage** (1791-1871): *Differential Engine, Analytical Engine* (nicht realisiert). **Ada Lovelace**: Gedankenspielereien zu “Software”
- **Konrad Zuse**: 1936 **Z1**: mechanisches, *programmgesteuertes* Rechnengerät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

# Prinzipien der Von Neumann-Architektur

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

Ein Computer besteht aus <b>Steuerwerk</b> , <b>Rechenwerk</b> , <b>Speicher</b> , und den <b>Ein-/Ausgabegeräten</b>	Steuerwerk = <b>CPU</b> . Rechenwerk = <b>ALU</b> , <b>Coproprozessoren</b> , verschiedene <b>Speicherebenen</b> .
Die Struktur des Computers muß <b>unabhängig</b> vom zu bearbeiteten Problem sein	
Im Speicher befinden sich <b>Daten</b> und das <b>Programm</b>	Heute meist mehrere Programme gleichzeitig
Der Speicher ist in <b>Zellen</b> gleicher Größe unterteilt und fortlaufend nummeriert	gilt im wesentl. unverändert (virtuelle Adressierung)
Das Programm besteht aus einer Folge von Befehlen, die nacheinander aufgeführt sind. 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”<sup>1</sup>
  - **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

---

<sup>1</sup>Es 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**:<sup>2</sup> 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, ...

---

<sup>2</sup>manchmal mehrstufig

# Bits und Bytes

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

1 bit	=	“ <i>binary digit</i> ” = 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
		⋮

---

<sup>3</sup>Das **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	100MB
Festplatte	10 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 Programme<sup>4</sup>
- grob: folgende **Schichten** im einem Rechner

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

<sup>4</sup>Die Grenzen sind nicht ganz so eindeutig.

# Abschnitt II

## Dateisystem

**Inhalt:** Dateien und Dateiformate · Organisation · spezielle Verzeichnisse  
· Navigation durch die Struktur · Veränderung der Verzeichnisstruktur

# 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**<sup>5</sup>
- z.B., `datei.txt` kennzeichnet eine **Textdatei**, weitere Beispiele folgen am Nachmittag
- Extensionen können **kombiniert** werden `datei.ps.gz`

---

<sup>5</sup>In 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:<sup>6</sup> enthält Menge von Dateien und (Unter)-Verzeichnissen
- Verzeichnis  $\approx$  Spezialform einer Datei<sup>7</sup>
- Pfad:
  - “Weg” durch den Baum  $\Rightarrow$  Lokalisierung von Dateien/Verzeichnissen
  - Trenner der Verzeichnisse: durch Schrägstrich (“slash”):<sup>8</sup> /
- Wurzel des Baumes (“root”-Verzeichnis), gekennzeichnet durch:

/

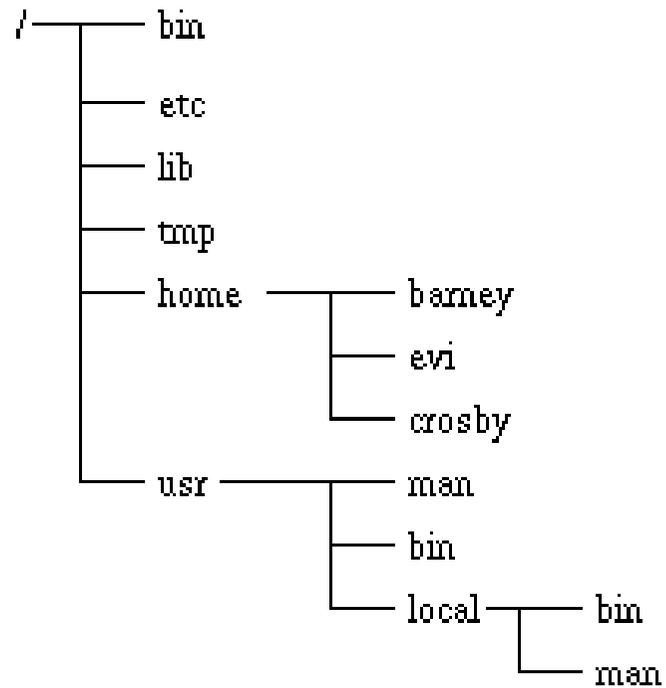
---

<sup>6</sup>Englisch: *directory*, beim Macintosh auch als Ordner bekannt.

<sup>7</sup>der Vollständigkeit halber: Unix 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.

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

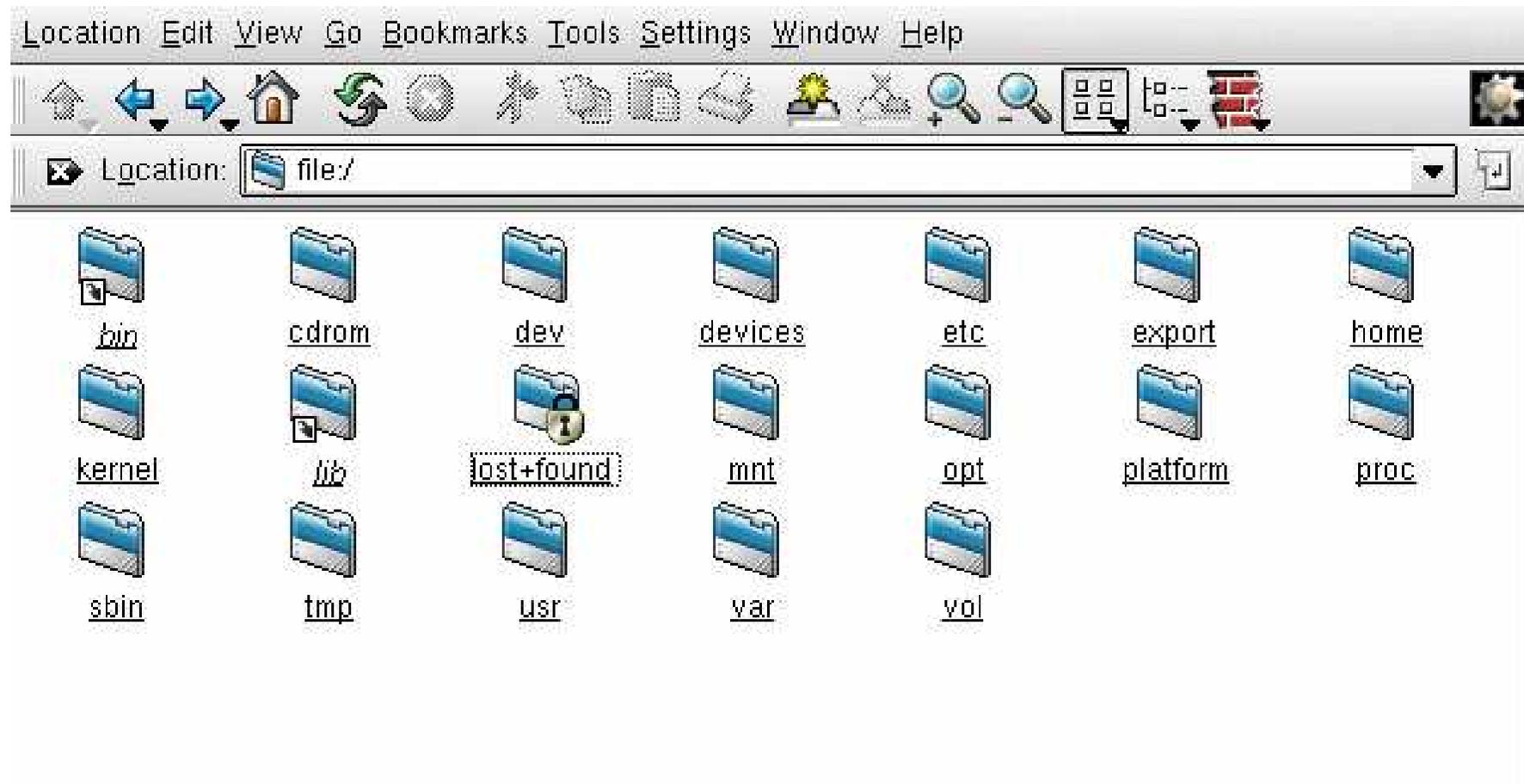
# Typische Struktur eines Verzeichnisbaumes



# Je nach "Browser": verschiedene Sichten (1)

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

## Je nach "Browser": verschiedene Sichten (2)



# Homeverzeichnis

---

- jede Benutzer: **Portion** des Plattenplatzes<sup>9</sup> = sein **Teilbaum** des Gesamtbau-  
mes ⇒
- **Einstiegspunkt** = **Heimatverzeichnis** (“*home*” oder “*home-directory*”)
- Beispiel: User `kurs001` ⇒  
`/home/kurs001/`

bezeichnet das entsprechende Home,

`/home/kurs001/texte/` und  
`/home/kurs001/texte/seminararbeit.ps`

ein **Unterverzeichnis** (vermutlich mit verschiedenen Texten) bzw. eine **Datei**,  
vermutlich die Postscriptdatei einer Seminararbeit

---

<sup>9</sup>Der einem von der Administration zugeteilte Anteil des Plattenplatzes heißt **Quota**, mit dem Befehl `quota` kann man sich über die Größe informieren.

# 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`<sup>10</sup> zeigt den absoluten Pfad des Arbeitsverzeichnisses
- **Inhalt** eines Verzeichnisses?: `ls <verzeichnis>`<sup>11</sup>
- Abkürzung: nur `ls`: listet das aktuelle **Arbeitsverzeichnis**
- **Wechsel** des Verzeichnisse: `cd`<sup>12</sup>

`cd <verzeichnis>`

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

---

<sup>10</sup>*print working directory*

<sup>11</sup>`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.

<sup>12</sup>*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

Befehl	Beispiel	Erklärung
<code>cp</code>	<pre>cp &lt;datei1&gt; &lt;datei2&gt; cp &lt;datei1&gt; &lt;verzeichnis&gt;</pre>	Kopieren
<code>mv</code>	<pre>mv &lt;datei1&gt; &lt;datei2&gt; mv &lt;datei1&gt; &lt;verzeichnis&gt;</pre>	<i>“move”</i>
<code>rm</code>	<pre>rm &lt;datei&gt;</pre>	<i>“remove”</i> : Datei löschen
<code>rmdir</code>	<pre>rmdir &lt;verzeichnis&gt;</pre>	<i>“remove directory”</i> : Verzeichnis löschen
<code>mkdir</code>	<pre>mv &lt;datei1&gt; &lt;verzeichnis&gt; mkdir &lt;verzeichnis&gt;</pre>	<i>“make directory”</i> : neues Verzeichnis anlegen



# Abschnitt III

## Betriebssystem

**Inhalt:** Aufgaben · Geschichte · Begriffsklärung

# 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. 1970<sup>13</sup>
- Nach 1980:
  - Aufstieg des **PC**: **DOS**, **Windows95** und seine Nachfolger<sup>14</sup>
  - Verbreitung von **lokalen Netzen** (verteilte Betriebssysteme), **Internet**
- Zukunft: **eingebettete Systeme?** **Netcomputer?** ...

---

<sup>13</sup>Man 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.

<sup>14</sup>Natürlich gibt's auch noch MAC-OS, und OS-2 &c

# Multitasking/Multiuser/Timesharing

---

- Heutzutage: **Betriebssystem** arbeitet viele Dinge (scheinbar) **gleichzeitig** ab<sup>15</sup>
- **Illusion** der Gleichzeitigkeit durch schnelles/geschicktes **Hinundherjonglieren** der zu bearbeitenden Aufgaben (Multitasking)
- insbesondere langsam: die menschlichen Reaktionen (Tastatur, Maus) und überhaupt E/A: ⇒ **interaktives** Verhalten für mehrere Benutzer möglich
- Mehrbenutzerbetrieb: erhöhte Anforderungen an **Sicherheit** (z.B. Trennung der Speicherbereiche etc.)

---

<sup>15</sup>Nur 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**.

# Abschnitt IV

## Sonstiges

**Inhalt:** Hilfe im System · Disketten · Drucken

# 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 xemacs<sup>16</sup>
  - Info-Information i.d.R. ausführlicher als *manpages*,
  - Manche (größere) Pakete verwenden Info-pages als primäre Nachschlagequelle
- **Internet**
  - bei **lokale** Problemen: FAQs des Instituts

---

<sup>16</sup>Emacs/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** zwischen<sup>17</sup> DOS  $\leftrightarrow$  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

<sup>17</sup>Es 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 gesprenkelt 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>`<sup>18</sup>
- Druckbar: **Text** (=“Ascii”), **Postscript**
- viele Tools: **print-Button**
- **Vorsicht** bei **Folien:** falsche Kombination von Folien/Drucker kann den Drucker **beschädigen**

---

<sup>18</sup>`lp` ist eine Alternative

# Abschnitt V

## Dateisystem (II)

**Inhalt:** Rechte und Privilegien · Spezielle Dateiformate

# 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 ⇒
  - **Lesen**: Inhalt mittel `ls` anzeigen lassen
  - **Schreiben** = Verzeichnisstruktur **ändern** d.h., neue Dateien/Unterverzeichnisse **anlegen/löschen**
  - **Ausführen** = in das Verzeichnis **wechseln** (mittels `cd`)<sup>19</sup>

---

<sup>19</sup>Beachte: 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.)

- **Anschauen** mittels `ls -lg` (g für „Gruppe“, l für „lang“, mit `ls -lag` bekommt man auch noch sog. **unsichtbare** Dateien angezeigt, a für „alle“)

---

```
[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
```

---

- **Ändern** mittels `chgrp` und `chown`<sup>20</sup>
- jeder Benutzer ist für das Setzen der Rechte selbst verantwortlich
- für Fortgeschrittene: Automatisierung mittels `umask`-Kommando

---

<sup>20</sup>für *change group* und *change owner*

# Dateiformate

Extension	Kürzel	Erklärung	Viewer/Bearbeitungsprogramm
*.txt	ASCII	einfachstes Text-format	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	L <sup>A</sup> T <sub>E</sub> X (oder T <sub>E</sub> X)		
*.dvi	“device independant”	Ergebnis eines L <sup>A</sup> T <sub>E</sub> X-Textes	xdvi
*.jpg, *.gif		zwei bekannte Bildformate, es gibt noch viele mehr	xv
*.fig		Zeicheneditorformat	xfig
*.gz *.Z		komprimierte Datei (mit Hilfe von gzip oder compress)	dekomprimieren: gunzip
*.tar	“tape archive”	Archiv	tar

Tabelle 1: Häufige Dateiformate und ihre Extensionen

# 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

# Abschnitt VI

## Interaktion

**Inhalt:** Kommandointerpreter · Befehle · Unix-Pfad · Sonstiges zur Bash

# Kommandointerpreter: die Shell

- wichtige System-**Schnittstelle**: **Kommandointerpreter**, **Shell**: interaktive Befehlseingabe
- In **X**-Menues: meist als **Xterm** benannt<sup>21</sup>
- es gibt eine Reihe von Un\*x-Shells:<sup>22</sup> 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)

---

<sup>21</sup>X-Terminal = ein Fensterrahmen um einen Kommandointerpreter herum

<sup>22</sup>Die 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 vorbelegt
  - von **Benutzer** anpassbar/erweiterbar
- Umgebungsvariable:
  - anzeigbar mittels `env`
  - verändern<sup>23</sup> mittels `export <VARIABLE>=<WERT>`
- wichtige Umgebungsvariable: **PATH**

---

<sup>23</sup>Umgebungsvariablen 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 Verzeichnisse<sup>24</sup> 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>`)

---

<sup>24</sup>der 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`<sup>25</sup>
- Nützliche, einfache Sache: sog. **alias**
- **Beispiel**: `alias zeigemiralles="ls -lag"`
- Falls man das **dauerhaft** will: einfügen der Zeile ins `~benutzer/.bashrc`

---

<sup>25</sup>Viele 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.