

Betriebssysteme

Systemprogrammierung

Prof. Dr. Rainer Werthebach

Studiengang Informatik

Hochschule Aalen - Technik und Wirtschaft

Kapitel 1

Einführung

Stand 01.03.2024

Gliederung – Vorlesung

1. Einführung

- 1.1 Komponenten einer Rechenanlage
- 1.2 Was ist ein Betriebssystem
- 1.3 Das Schichtenmodell
- 1.4 Schnittstellen und virtuelle Maschine
- 1.5 Die Geschichte von Betriebssystemen

2. Prozesse

- 2.1 Prozesszustände
- 2.2 Scheduling
- 2.3 Synchronisation
- 2.4 Kommunikation

3. Speicherverwaltung

- 3.1 Speicherbelegungsstrategien
- 3.2 virtueller Speicher
- 3.3 Seitenverwaltung
- 3.4 Segmentierung
- 3.5 Cache

4. Dateiverwaltung

- 4.1 Dateisysteme
- 4.2 Dateiattribute
- 4.3 Dateifunktionen
- 4.4 Dateiorganisation

5. Ein- und Ausgabeverwaltung

- 5.1 Das Schichtenmodell
- 5.2 Gerätemodelle
- 5.3 Treiberprogrammierung

6. Netzwerkdienste

- 6.1 Der Netzwerkanschluss
- 6.2 Kommunikation im Netz
- 6.3 Dateisysteme im Netz
- 6.4 Sicherheitsmechanismen

Gliederung - Übung

- allgemein

1. Einführung, Linux als Beispielbetriebssystem
2. Benutzer, Gruppen, Rechte, Links, Ein-/Ausgabeumlenkung
3. Pipes, einige Kommandos, Editoren

- Shellprogrammierung

4. Shell Skripte, Variablen, Subshell, Benutzereingaben, here-Document, for-Anweisung
5. case-, test-, if-Anweisungen, Funktionen, Steuerkommandos
6. Das tree Skript – ein ausführliches Beispiel

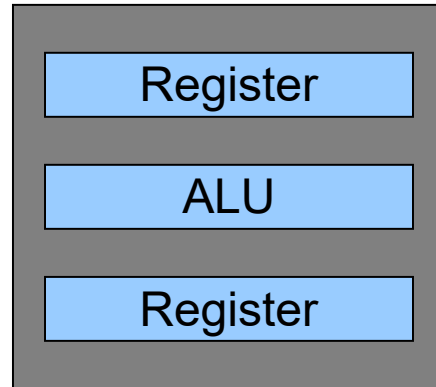
- Systemprogrammierung

7. Prozesse, Signale
8. Makefiles, fork, wait, exit, daemons, zombies
9. Pipes, Threads, Mutexe, Semaphore, Shared Memory

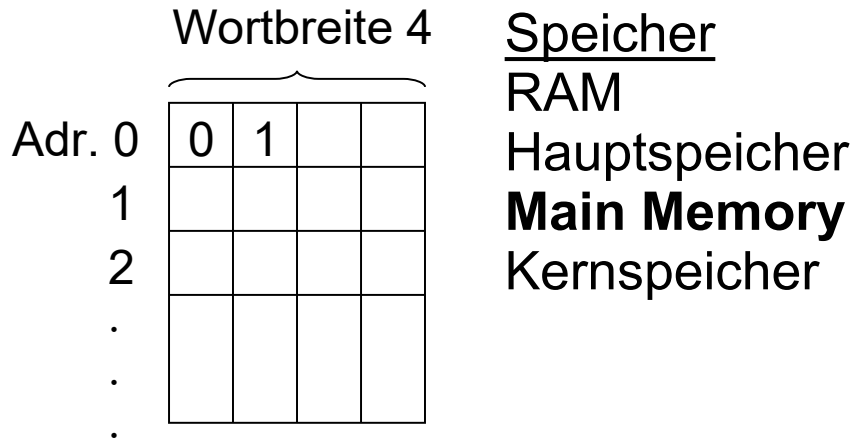
1.1 Komponenten einer Rechenanlage

CPU:
Central
Processing
Unit

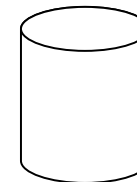
Prozessor



ALU: Arithmetisch Logische Einheit
+, -, *, /, and, or, not, <<, >>, ...)

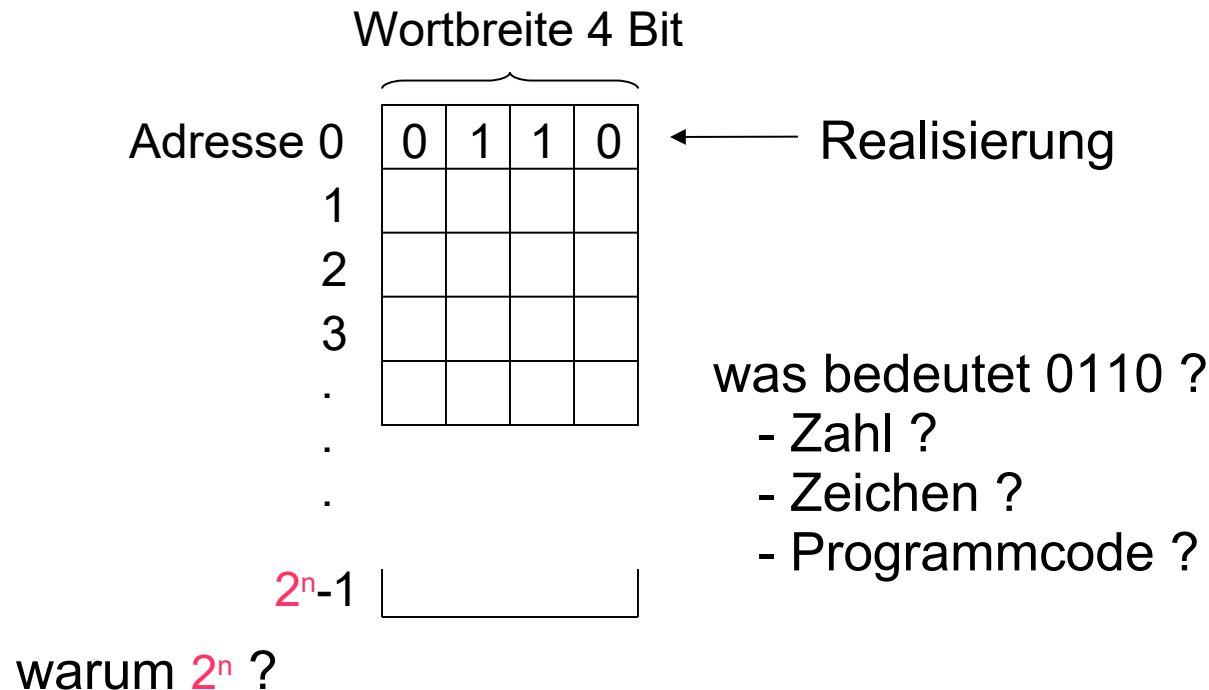


Externer Speicher



(Haupt-) Speicher

- organisiert in Zeilen und Spalten
- enthält Speicherwörter



Warum wird **0,1** genutzt: leicht zu implementieren

1 ... Spannung, Strom, Ladung,

N
S

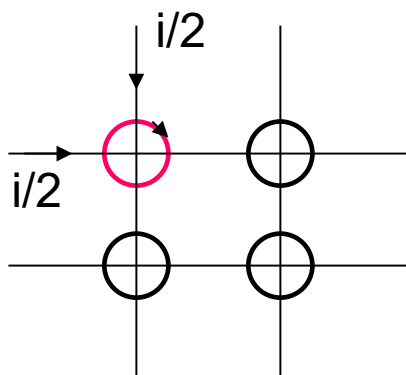
 - magnetisiert

0 ... keine Spannung, kein Strom, keine Ladung,

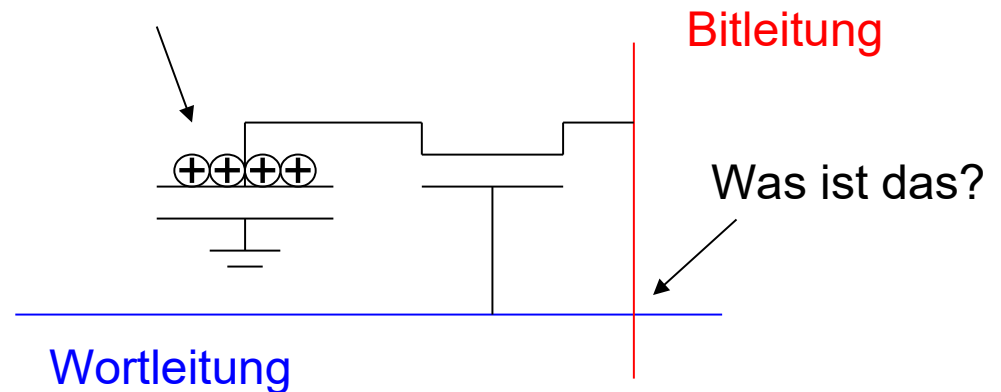
S
N

 - magnetisiert

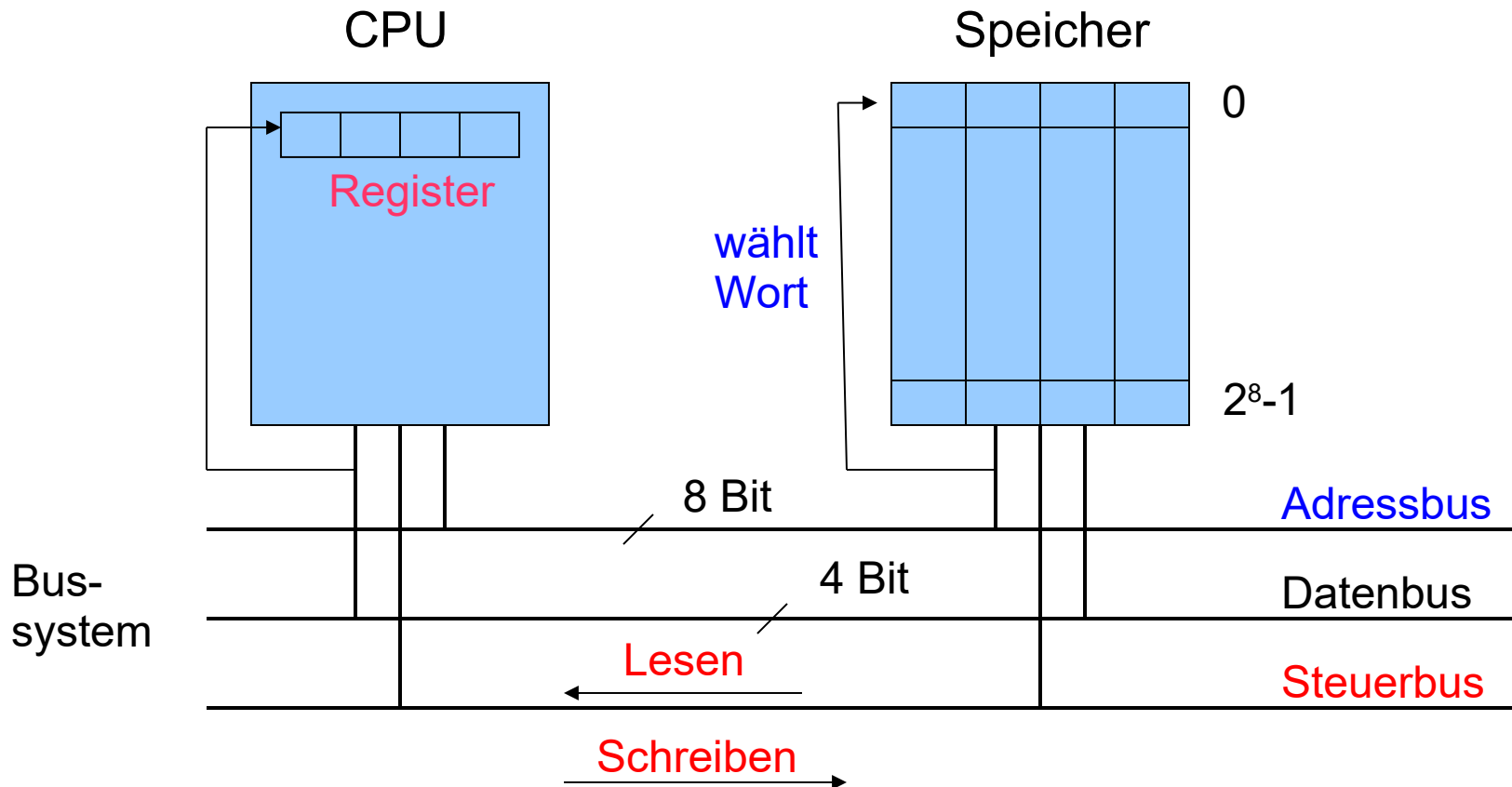
Kernspeicherprinzip



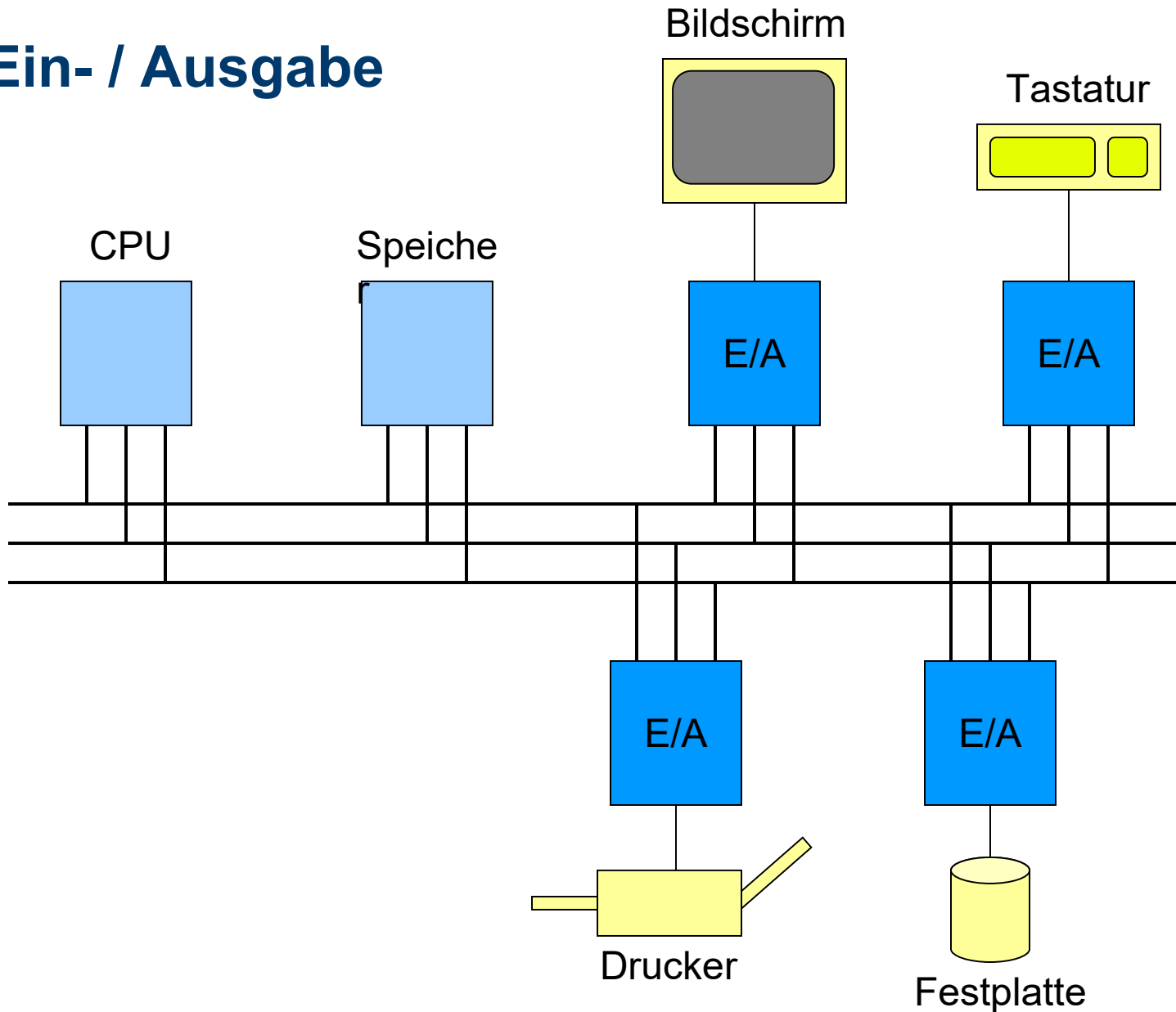
Speicherkondensator



Kommunikation Prozessor/Speicher

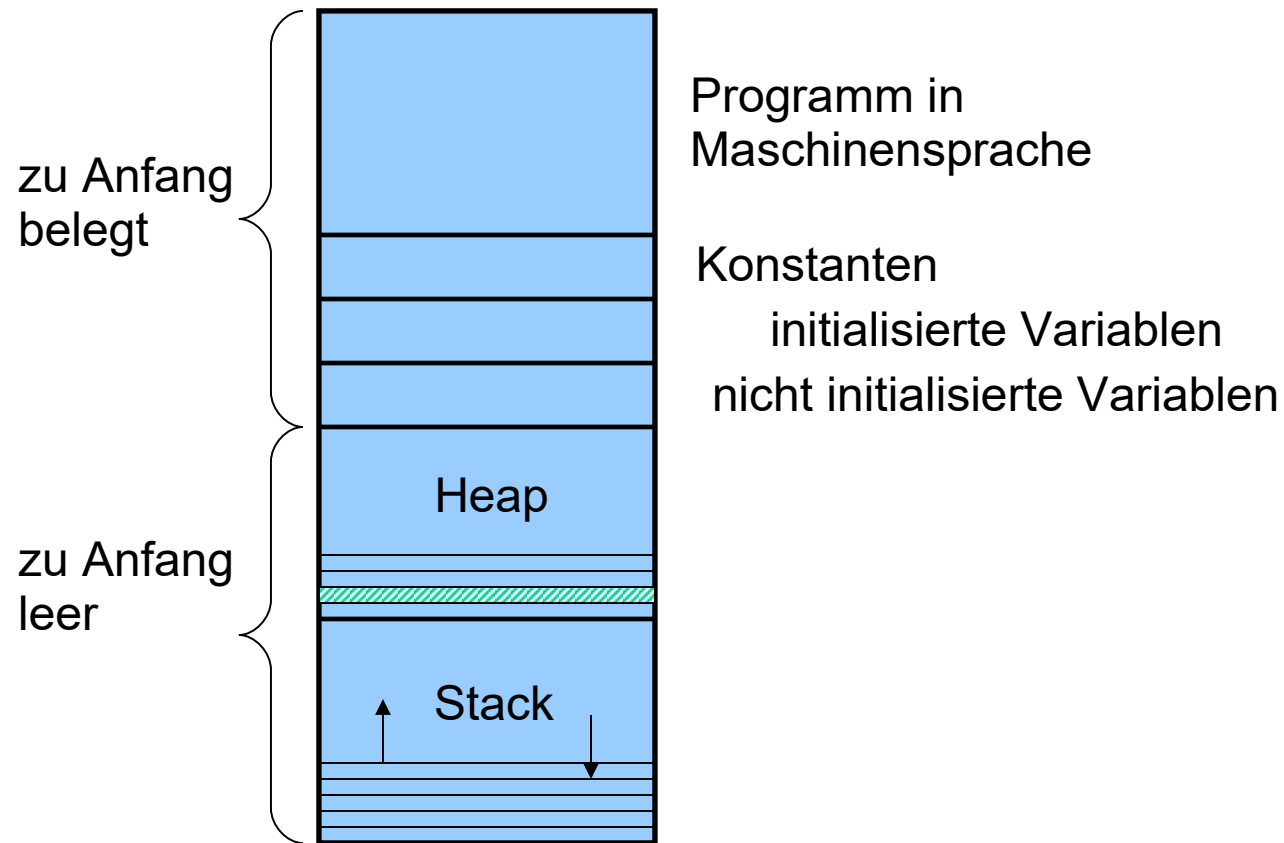


Ein- / Ausgabe



Von-Neumann Architektur

Programm + Daten liegen im gleichen Speicher



1.2 Was ist ein Betriebssystem?

A: Historisch gesehen enthält ein BS alle Programme, die nötig sind, einen Rechner für verschiedene Aufgaben zu betreiben.

vgl.

Heizungssteuerungssysteme auf Mikrokontrollern brauchen kein BS. Es werden Maschinenprogramme direkt für eine Aufgabe entworfen, die die Hardware steuern.

B: Das BS ist die Software, die für den Betrieb eines Rechners anwendungsunabhängig notwendig ist.

Ressourcenverwalter

C: Ein BS ist die Gesamtheit aller Programme, die die Benutzung von Betriebsmitteln steuern und verwalten.

Betriebsmittel (Ressourcen) sind:

- Prozessor
- Speicher
- E/A-Geräte, z.B. Tastatur, Monitor, Drucker
- logisch: Dateien, Programme

D: Schnittstelle zwischen Mensch und Maschine (virtuelle Maschine)

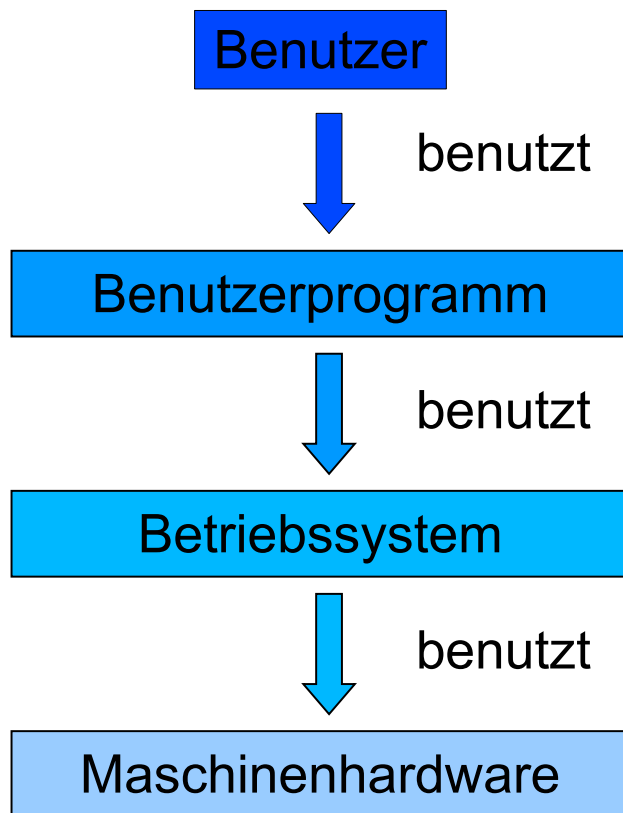
Der Programmierer nutzt:

write (dateinummer, textadresse, bytezahl);

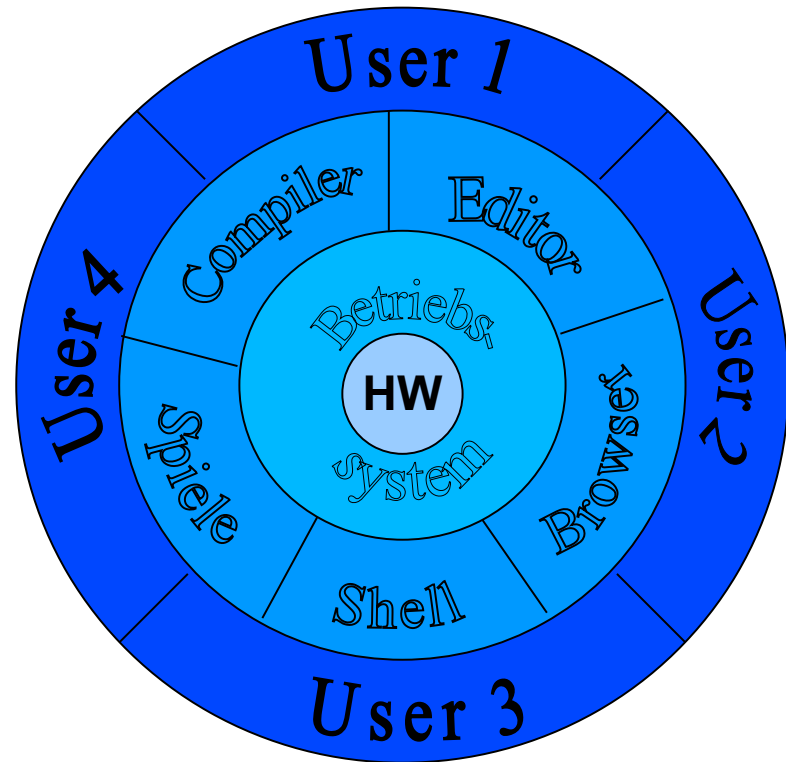
anstatt ???

1.3 Das Schichtenmodell

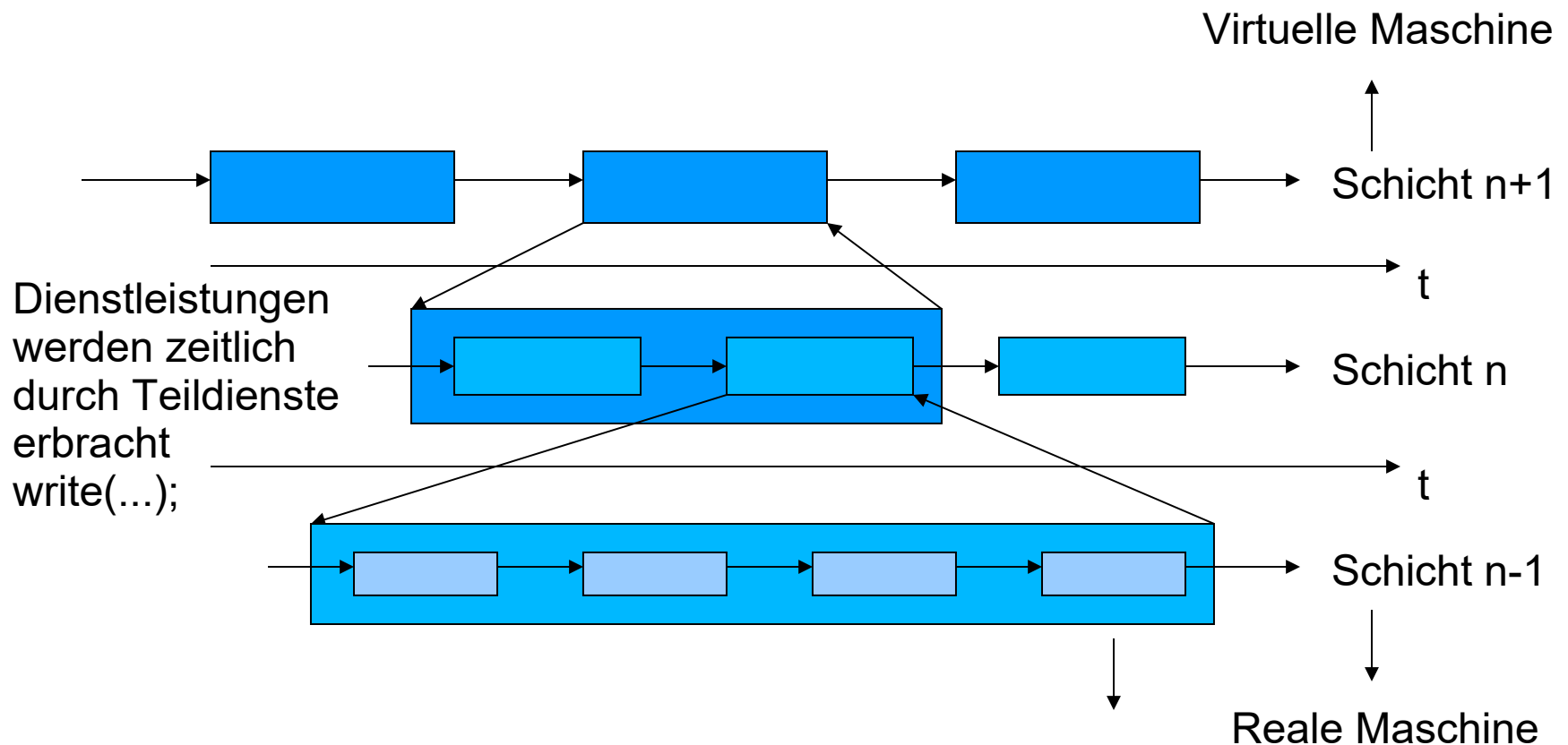
Einfaches Schichtenmodell



Schalenmodell



1.4 Schnittstellen und virtuelle Maschine



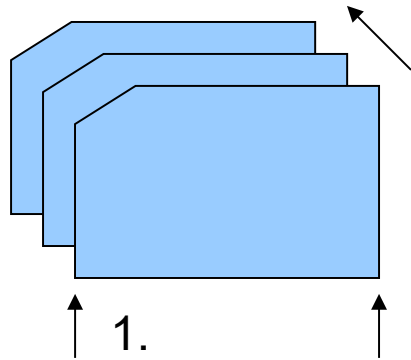
1.5 Die Geschichte von Betriebssystemen

Betriebssystem + Rechnerarchitektur → Generationen

0. Generation (1940 - 1950): Röhren und Steckbretter

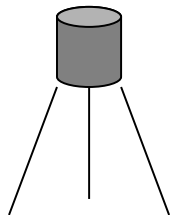
- **kein** Betriebssystem
- Programme in **Maschinensprache** (Prozessorbefehle)
- **Kabel umstecken** auf Steckbrettern
- **keine** Programmiersprachen

1. Generation (1950 - 1960): Lochkarten, Stapelsystem, Transistoren



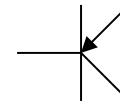
Stapel aus genormten **Papierkarten**
ersetzt das Kabelumstecken

Kodierung von bis zu **80 Zeichen**

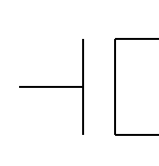


ersetzt Röhre

Schaltbild:



Bipolar



MOS

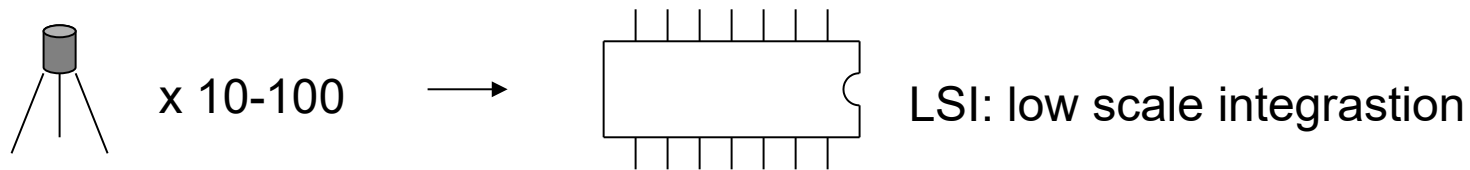
- Berufe: Operator
 Wartungspersonal
 Computerarchitekt
 Programmierer
- Programmierer stanzt Programm auf Lochkarte.
 Programmiersprachen: Assembler, Fortran, Cobol.
- Operator übernimmt Stapel, liest¹⁾ Programm ein, lässt berechnen²⁾,
 gibt Ausdruck an Programmierer zurück.

1) auf „billigem“ Computer

2) auf eigentlichem Computer

2. und 3. Generation (1960 - 1975): integrierte Schaltkreise ICs

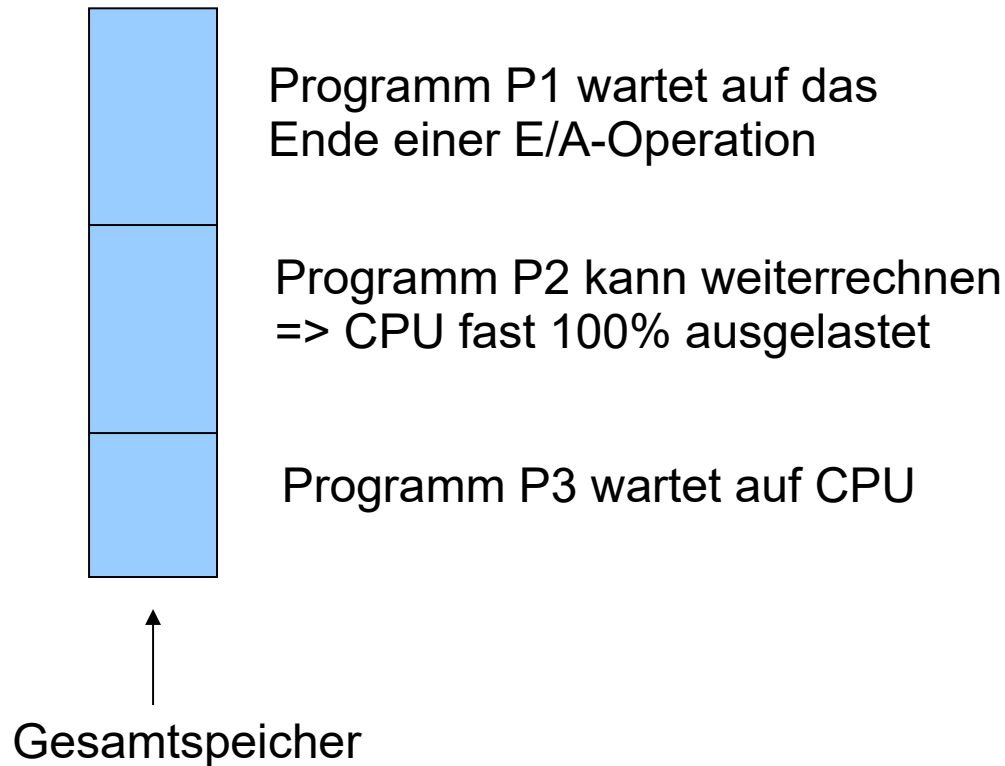
Multiprogramming, Spooling, Timesharing



IBM 360 → wissenschaftliche Datenverarbeitung
→ kommerzielle Datenverarbeitung

Betriebssystem: Millionen Zeilen aus Assemblercode

Multiprogramming



Spooling (Simultaneous peripheral operations online)

Jobs (Lochkartenstapel) werden direkt, wenn sie abgegeben werden auf Festplatte geladen. BS kann nun neuen Job schneller beginnen.

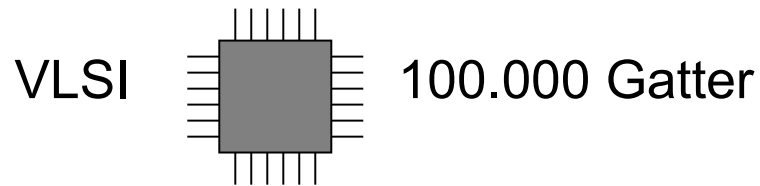
Timesharing

Neue Variante des Multiprogramming.

Jeder Benutzer hat sein eigenes **Terminal** (**Bildschirm, Tastatur** direkt zum Computer). Die Rechenzeit wird in Stücke (**Zeitscheiben**) aufgeteilt. Jeder Benutzer erhält der Reihe nach eine Zeitscheibe, in der er die CPU nutzen darf.

4. Generation (1975 - 2006): PCs und Computernetze

PCs mit den Betriebssystemen MS-DOS, MS-Windows, UNIX/LINUX



PCs preisgünstig

DOS: single user BS

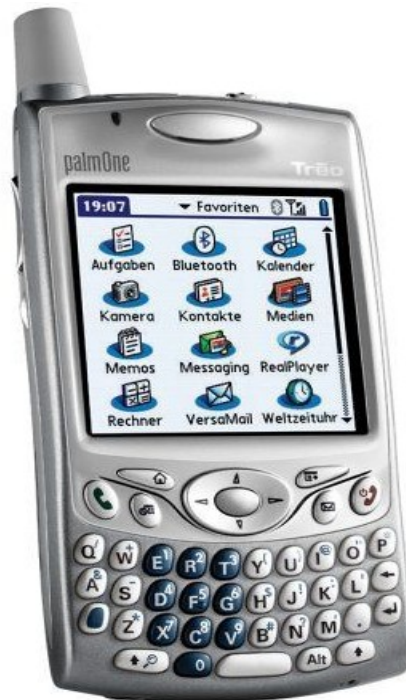
Windows: multitasking BS (task = Aufgabe)

Unix: multitasking/multiuser BS

Computernetze: - Kommunikation
- Steigerung der Rechenleistung

5. Generation (2006 – heute): PDAs, Smartphones, Tablet-PCs Embedded Systems

Palm OS, iOS, Android
Windows Embedded, Windows Mobile, Embedded Linux
Java ?





The Java™-based Middleware
for Measurement, Control and Automation

 suchen


Start | Neues | Über JControl | Forum | Wiki | Beispiele | Tutorials | Dokumentation | Downloads | Anbieter | Partner

