

# Grundlagen der Informatik

## – Rechnergrundlagen 2 –

Prof. Dr. Bernhard Schiefer

basierend auf Unterlagen von Prof. Dr. Duque-Antón  
einige Abbildungen aus [Herold,Lurz,Wohlrab: GDI, 2012)]

[bernhard.schiefer@fh-kl.de](mailto:bernhard.schiefer@fh-kl.de)  
<http://www.fh-kl.de/~schiefer>



# Inhalt

---

- Hardware
- Betriebssystem

# Hardware

---

- Bei Computersystemen werden häufig folgende Varianten unterschieden:

- ⇒ Personal Computer (PCs)
- ⇒ Workstations (ähnlich PC, leistungsfähiger und robuster)
- ⇒ Mainframes (Zentralrechner)
- ⇒ Super-Computer (schnellste Rechner der Welt)

-> S. neuen Foliensatz "Ubiquitous Computing"

- Viele weitere Bezeichnungen für Varianten

# von-Neumann-Architektur

---

- Die von-Neumann-Architektur ist ein Referenzmodell für Computer:
  - ⇒ ein gemeinsamer Speicher für Befehle und Daten
  - ⇒ Bus-Systeme zur Verbindung aller Komponenten
- Geschichte / Bedeutung
  - ⇒ Die entsprechende Architektur wurde bereits 1945 von John von Neumann (1903-1957) veröffentlicht.
  - ⇒ Mit dem Ansatz lassen sich Programme für alle prinzipiell lösbaren Probleme formulieren – vorherige Ansätze basierten auf fest verdrahteter Logik
  - ⇒ Alles was von einer Turingmaschine berechnet werden kann, kann auch mit diesem Rechner grundsätzlich gelöst werden.

# von-Neumann Architektur

## ■ Ein von-Neumann-Rechner enthält folgende Komponenten:

- ⇒ Rechenwerk: führt Rechenoperationen und logische Verknüpfungen durch. Auch Prozessor, Zentraleinheit oder ALU (Arithmetic Logical Unit) genannt.
- ⇒ Steuerwerk : interpretiert die Anweisungen eines Programms und steuert die Befehlsabfolge. Auch Control Unit oder Leitwerk genannt.
- ⇒ Speicherwerk (auch Memory genannt): speichert sowohl *Befehle/Programme* als auch *Daten*, welche für das Rechenwerk zugänglich sind.
- ⇒ Ein-/Ausgabewerk: steuert die Ein- und Ausgabe von Daten zum Anwender (Tastatur, Bildschirm) oder zu anderen Systemen (Schnittstellen). Auch I/O-Unit genannt.

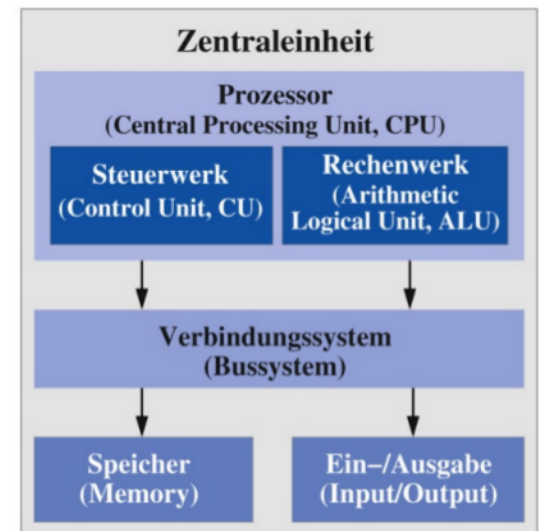
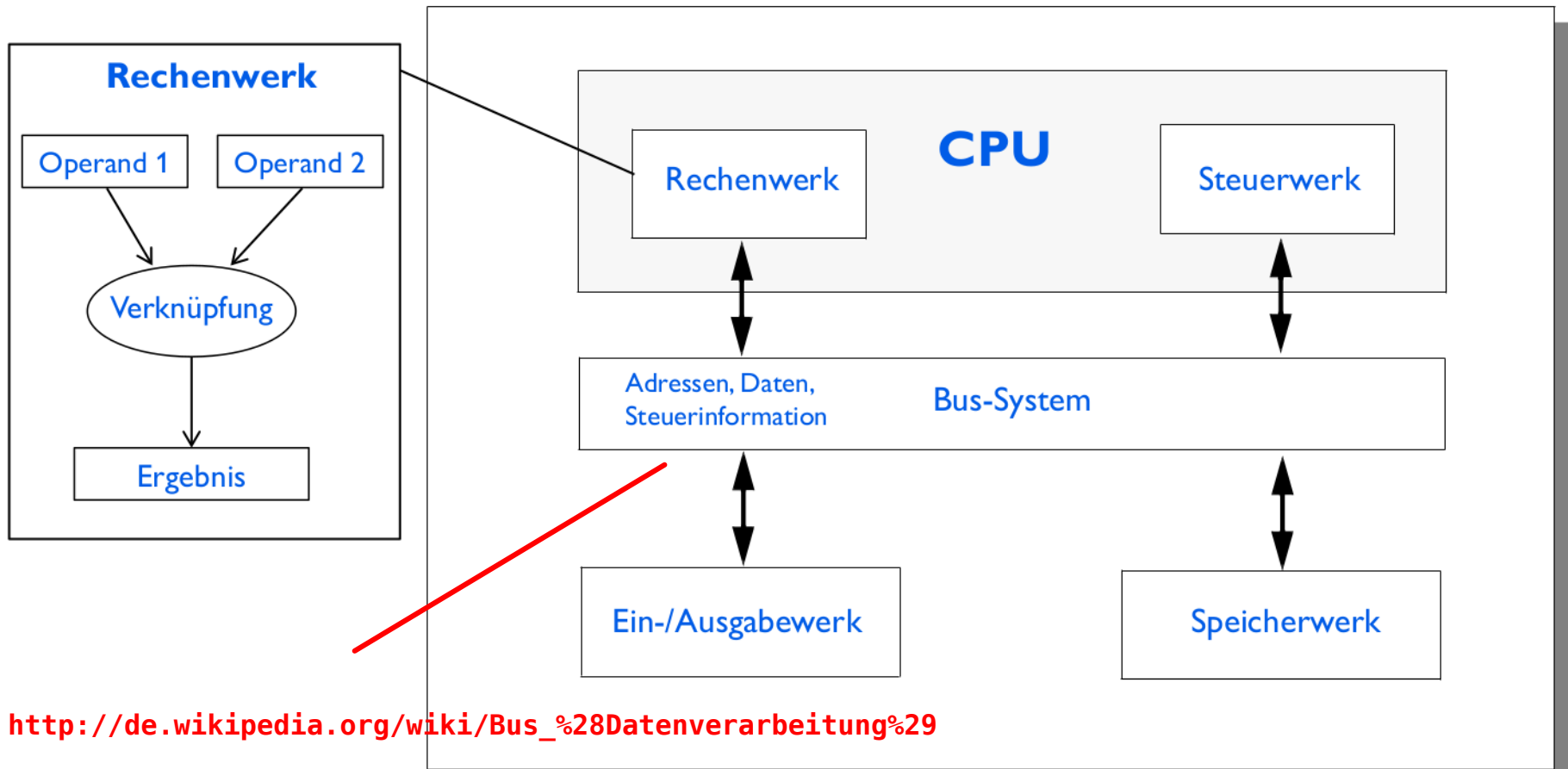


Abbildung 2.10: Das von-Neumann'sche-Rechnermodell

# Komponenten: von-Neumann-Rechner



[http://de.wikipedia.org/wiki/Bus\\_%28Datenverarbeitung%29](http://de.wikipedia.org/wiki/Bus_%28Datenverarbeitung%29)

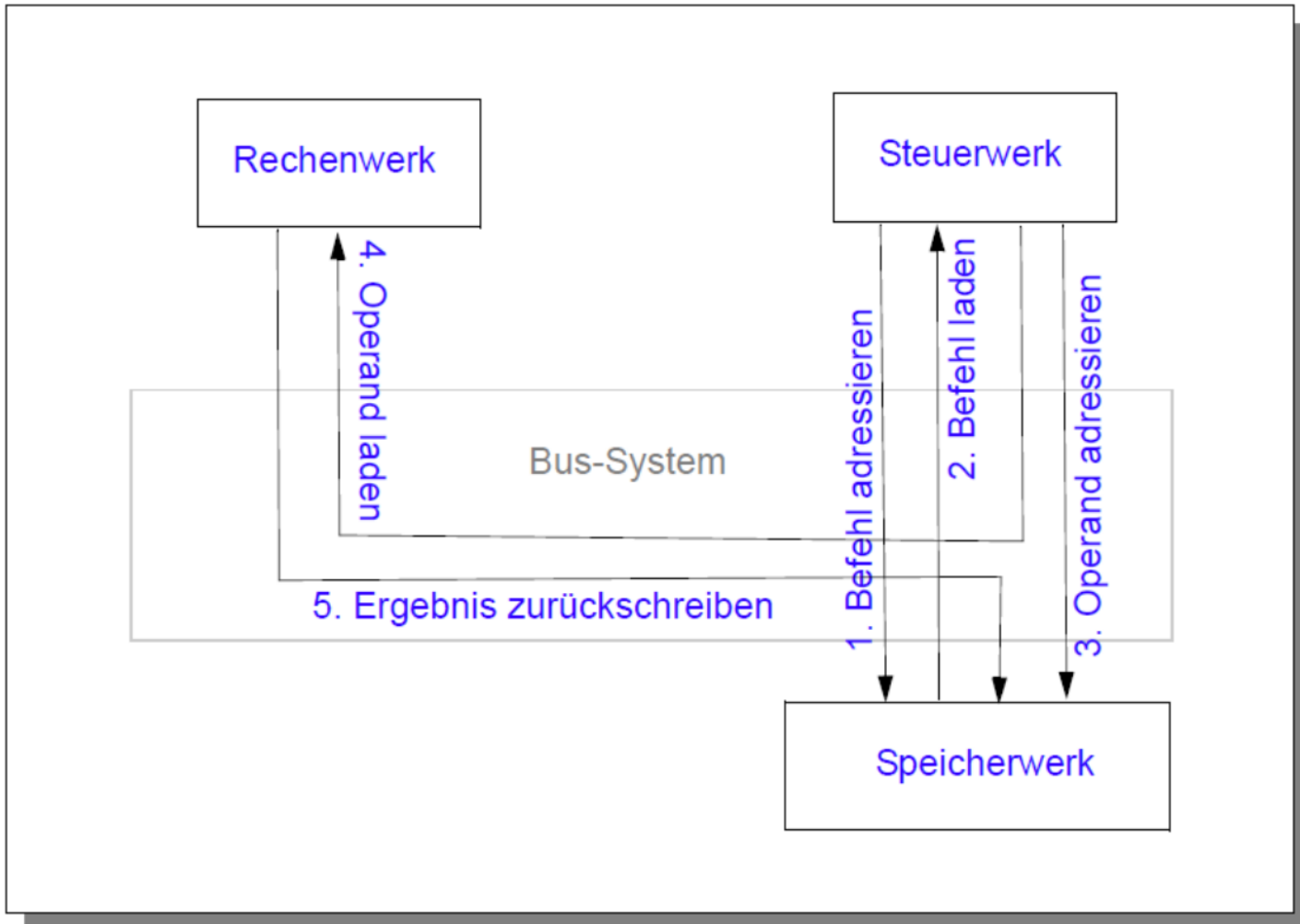
# Beispiele

---

- **Load <4>** // Hole den Inhalt der Speicherzelle 4 ins Rechenwerk:
- Dazu werden vom Steuerwerk die folgenden Operationen durchgeführt:
  - ⇒ Über die Adressleitung wird die Adresse 4 ausgegeben.
  - ⇒ Über die Steuerleitung zum Speicher wird das Signal „Lesen“ gegeben.
  - ⇒ Über die Steuerleitung zum Rechenwerk wird das Signal „Schreiben in das erste Operanden-Register“ gegeben.
- **Add <5>** // Addiere Inhalt der Speicherzelle 5 zum Inhalt d. Speicherzelle 4:
- Dazu werden vom Steuerwerk die folgenden Operationen durchgeführt:
  - ⇒ Über die Adressleitung wird die Adresse 5 ausgegeben.
  - ⇒ Über die Steuerleitung zum Speicher wird das Signal „Lesen“ gegeben.
  - ⇒ Über die Steuerleitung zum Rechenwerk wird das Signal „Schreiben in das zweite Operanden-Register“ und „Addition als Verknüpfung“ gegeben.

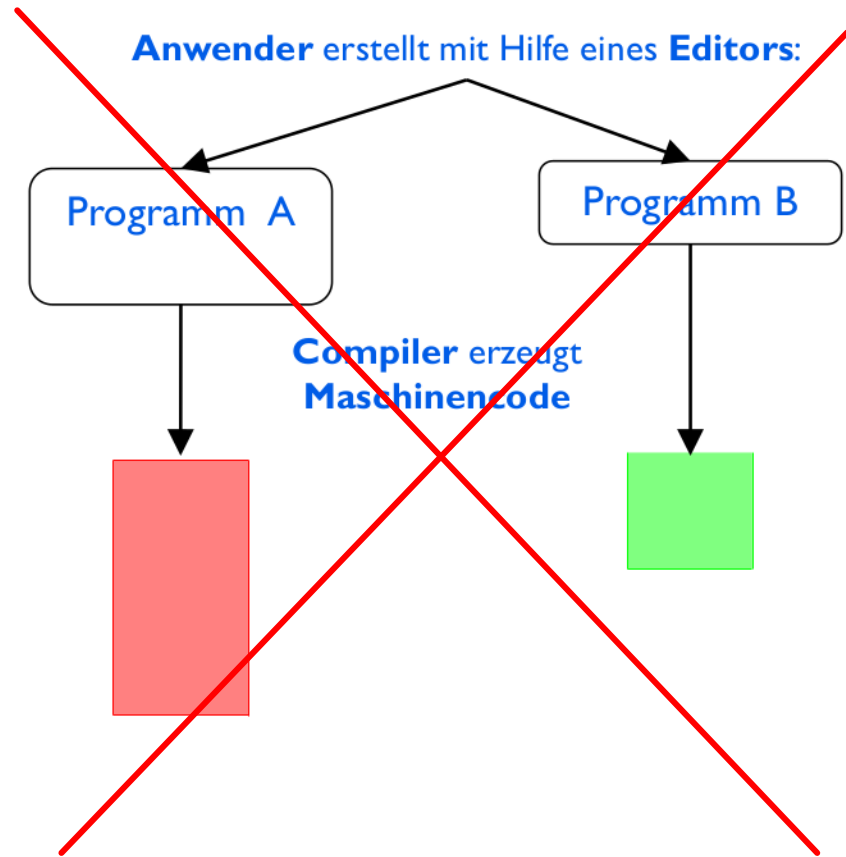
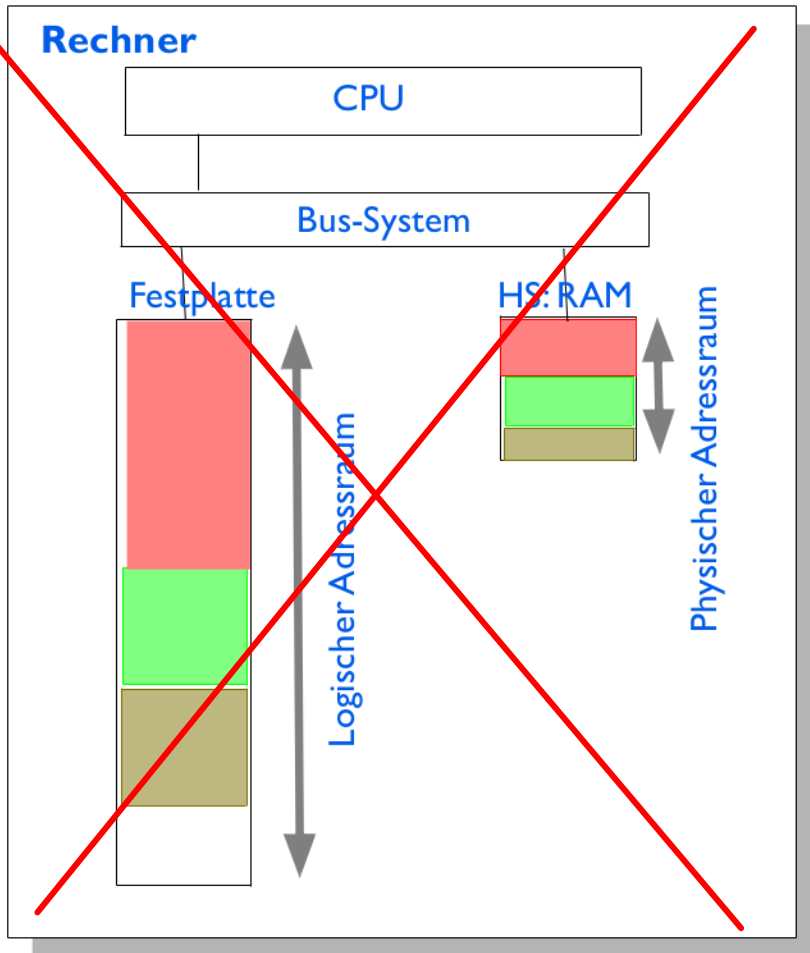
# Verarbeitung eines Datenwortes

Weglassen





# Zusammenwirken der Komponenten



# Weitere Komponenten

## Flüchtiger Speicher (RAM) vs. Festspeicher (Disk)

### (RAM)

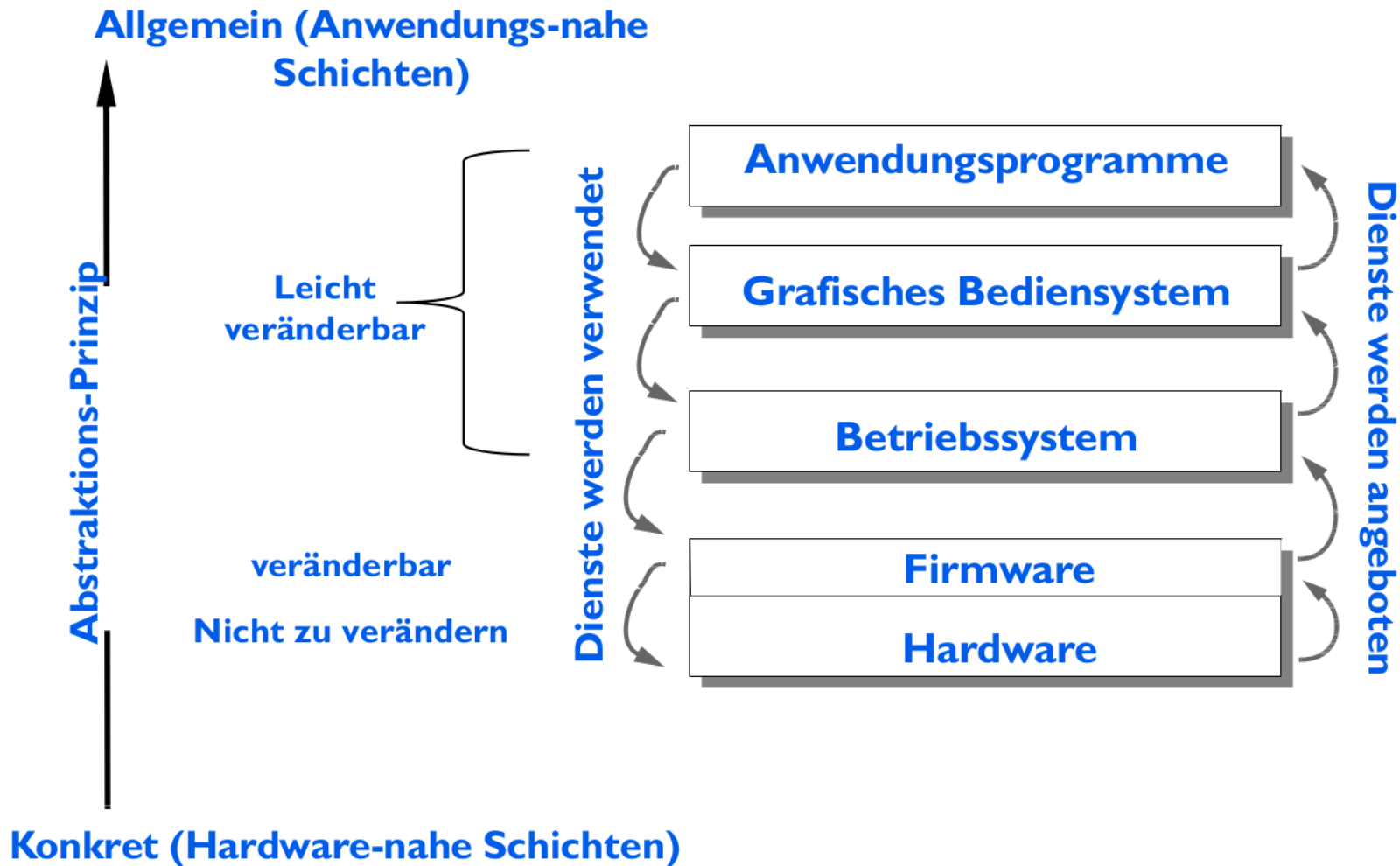
- Der **Arbeitsspeicher** eines Rechners verliert seinen Inhalt, wenn er nicht in regelmäßigen Abständen (z.B. alle 15  $\mu$ s) “aufgefrischt” wird. Insbesondere gehen beim Abschalten alle Daten verloren.
- Zur *langfristigen Speicherung* werden daher andere Speichertechnologien benötigt. Ihre Kenngrößen sind Speicherkapazität und Zugriffszeit
  - ⇒ Der wichtigste nichtflüchtige Massenspeicher ist die **Festplatte**. Ihre Kapazität hat innerhalb der letzten 15 Jahre um das Zehntausendfache zugenommen und nimmt weiter zu.
  - ⇒ Prinzipiell sind Festplatten und Disketten sehr ähnlich aufgebaut.
  - ⇒ Optische Platten wie **CD** und **DVD** schreiben Bits mithilfe von Löchern (pits), die beim Schreiben eingebrannt werden.
- Daneben gibt es weitere Ein- u. Ausgabegeräte wie ISDN-Karte, Netzwerkkarte, externe Festplatten, Mikrofone, Lautsprecher, Drucker und vieles mehr. Einige davon sind bereits im Inneren eines Rechnergehäuses fest eingebaut.
- Der ~~von-Neumann~~ Flaschenhals der von-Neumann-Architektur bezeichnet den Sachverhalt, dass das Bus-System zum Engpass zwischen dem Prozessor und dem Speicher wird. Da die CPU-Taktraten wesentlich schneller ansteigen als die verwendeten Speicherbausteine, wird der Hauptspeicher ebenso zum Flaschenhals.

# Betriebssystem

---

- Bisher haben wir die Hardware und die Möglichkeiten der Datenrepräsentation diskutiert. Ein solcher „blanker“ Rechner bzw. CPU kann nicht viel mehr als
  - ⇒ Speicherinhalte in Register laden, Registerinhalte in Speicher ablegen,
  - ⇒ Registerinhalte logisch oder arithmetisch verknüpfen,
  - ⇒ mit Ein- oder Ausgabebefehlen Register in Peripheriegeräten lesen und schreiben.
- Um einen Rechner auf dieser sehr niedrigen Hardware-Ebene bedienen zu können, muss man alle technischen Details kennen.
- Auf Benutzer/Anwender Ebene will man eher folgende Dinge erledigen:
  - ⇒ Briefe editieren und drucken,
  - ⇒ E-Mail bearbeiten und versenden,
  - ⇒ Fotos und Grafiken bearbeiten,
  - ⇒ Simulationen ausführen u.v.m.
- Die Lücke zwischen (intuitiven) Anwender und den Fähigkeiten eines Rechners wird mit Hilfe von Zwischenschichten geschlossen, die über geeignete Schnittstellen in einem Schichtenmodell miteinander kommunizieren.

# Hierarchisches Schichtenmodell



# Aufgaben des Betriebssystems

---

- Ein Rechner mit seinen Peripheriegeräten stellt Betriebsmittel (Ressourcen) zur Verfügung, auf die Anwender-Programme zugreifen können:
  - ⇒ CPU (Rechenzeit),
  - ⇒ Hauptspeicher,
  - ⇒ Plattenspeicherplatz,
  - ⇒ externe Geräte wie Drucker, Modem oder Scanner.
- Zur Verwaltung dieser Betriebsmittel müssen viele Benutzerprogramme gleichzeitig auf diese Ressourcen zugreifen. Ein Betriebssystem muss daher die folgenden zentralen Aufgaben lösen:
  - ⇒ Prozess verwalten,
  - ⇒ Speicher verwalten und
  - ⇒ Dateien verwalten.

# Prozessverwaltung

---

- Zur Lösung der Programmieraufgabe müssen Programme auf einem Rechner ausgeführt werden.
  - ⇒ Solche (dynamischen) Codeausführungen nennt man *Prozesse*.
- Häufig führt schon der Aufruf eines Programms zu vielen gleichzeitig und unabhängig voneinander laufenden (Teil-) Prozessen.
- Ein Prozess ist also ein eigenständiges Stück Programmcode mit eigenem Speicherbereich, der vor dem Zugriff durch andere parallel laufende Prozesse geschützt werden muss.
- In der Regel laufen auf einem Rechner (eine oder mehrere CPUs) viele Prozesse gleichzeitig (Time Sharing).
  - ⇒ Das Betriebssystem muss also alle nebenläufigen Prozesse möglichst fair verwalten.
  - ⇒ Ebenso muss die Kommunikation zwischen den Prozessen realisiert werden und zwar so, dass sich die Prozesse nicht gegenseitig beeinträchtigen oder sogar zerstören.  
**Was bedeutet "zu viele"?**
- **Vorsicht: Zu viele Prozesse machen das System langsam!**

# Speicherverwaltung

---

- In Analogie zu den Prozessen muss auch der Hauptspeicher verwaltet werden, in dem die Daten der vielen Prozesse gespeichert werden:
  - ⇒ Neuen Prozessen muss freier Hauptspeicher zugewiesen werden und
  - ⇒ der Speicher terminierter Prozesse muss wiederverwendet werden.
  - ⇒ Die Speicherbereiche verschiedener Prozesse müssen vor gegenseitigen Zugriff geschützt werden.
  - ⇒ Also: Das Betriebssystem
- I.d.R. ist der Bedarf an Arbeitsspeicher größer als der verfügbare physische Speicher.
  - ⇒ Dabei geht man von der Tatsache aus, dass zu einem beliebigen Zeitpunkt nur auf wenige Speicherplätze tatsächlich zugegriffen wird, während die anderen nur für einen späteren Zugriff (der u.U. nie erfolgt) bereitstehen.
  - ⇒ Die Grundidee ist also eine Erweiterung der Speicherkapazität unter Zuhilfenahme externer Massenspeicher, von denen bei Bedarf die benötigten Informationen in den Arbeitsspeicher geladen werden.

-> Virtual Memory, das Betriebssystem kümmert sich um den Speicher, nicht die Programme selbst

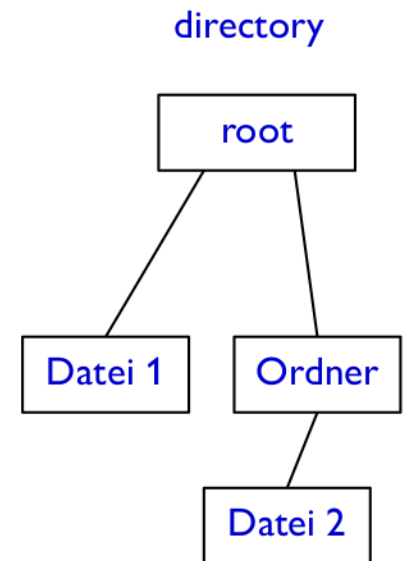
# ~~Dateiverwaltung~~

- ~~Das~~ <sup>as</sup> ~~Dateiverwaltung~~ <sup>system</sup> übernimmt die Aufgabe, Dateien auf die konkreten Gegebenheiten der Speichermedien abzubilden:
  - ⇒ In welche Sektoren bzw. auf welche Spuren und Köpfe wird eine gerade geschriebene Text-Datei auf die Platte geschrieben, oder
  - ⇒ wo ist die Version des Textes, das gestern gespeichert wurde?
- In diesem Sinne stellt das Betriebssystem das Konzept der Datei als Behälter für Daten aller Art zur Verfügung.
- Moderne Dateisysteme sind hierarchisch aufgebaut
  - Verzeichnishierarchie (Windows: zusätzlich "Laufwerksbuchstaben", Linux: Durchgängig ein Dateisystembaum)



# Verzeichnisstrukturen

- Mehrere Dateien können zu einem Ordner zusammengefasst werden
  - ⇒ Übliche Benennung: Verzeichnis (engl. directory)
  - ⇒ Da Ordner sowohl normale Dateien als auch andere Ordner enthalten können, entsteht eine baumähnliche Struktur mit einem Wurzelordner (engl. root) an der Spitze.
- Jede Datei erhält einen Namen, unter der sie gespeichert und wiedergefunden werden kann.
- Zusätzlich werden Erweiterungen verwendet, welche die Dateiinhalte spezifizieren und Attribute.



# DOS und Windows

---

- Frühe Betriebssysteme für PCs waren in erster Linie *Dateiverwaltungssysteme*. Wichtigster Vertreter war DOS (Disk Operating System):
  - ⇒ In diesem Fall kann immer nur ein Programm nach dem anderen ausgeführt werden.
  - ⇒ Die Schnittstelle zum Benutzer (Bediensystem) ist die Kommandozeile
  - ⇒ Erweiterungen (Prozess- und Speicherverwaltung): Windows 3.1, Windows 95, 98, ME, NT, 2000, XP, Windows 7, ...
- DOS ist ein Beispiel für ein Bediensystem, welches *Kommandozeilenorientiert* ist.
  - ⇒ Der Benutzer tippt ein Kommando ein, das vom Betriebssystem sofort ausgeführt wird und zwar über den sog. Kommandointerpreter (shell).
  - ⇒ Um z.B. die Namen aller Dateien im aktuellen Verzeichnis anzuzeigen, gibt man *dir* ein oder zur
  - ⇒ Umbenennung einer Datei mit *ren alt.doc neu.doc*
  - ⇒ Wird in Windows über cmd.exe aufgerufen oder über Menue.

Eine Kommandozeile, ähnlich wie `command.com` bei DOS. \_

# Linux

- Linux ist an UNIX angelehnt und wurde von dem finnischen Studenten Linus Torvalds entworfen und wird seitdem von tausenden Programmierer weltweit weiter entwickelt.

⇒ Der Quellcode ist frei zugänglich.

Wirklich? Effizient ist jedenfalls korrekt.

⇒ Es gilt als effizienter, schneller und robuster als Windows.

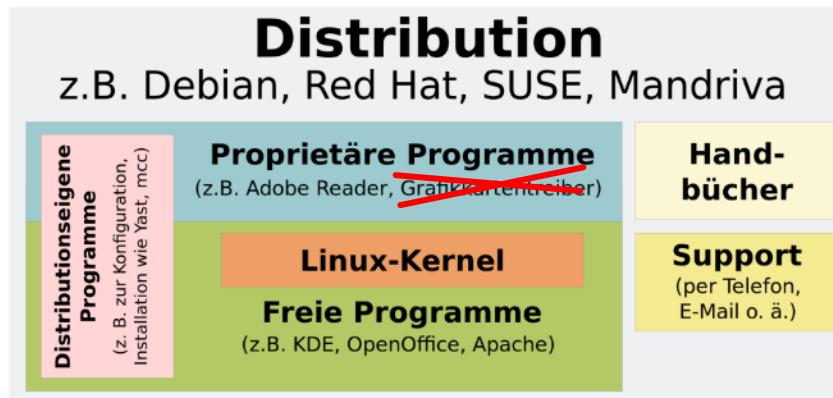
Das Entscheidende ist weniger die Technik, als die Nutzungs- und Entwicklungsbedingungen.

⇒ ~~Heute ist es genau wie Windows~~ sehr einfach zu bedienen. Der Benutzer kommuniziert über eine GUI (Graphical User Interface) mit dem System.

oder alternative Benutzerinterfaces (z.B. ADRIANE)

## ■ Linux-Distributionen

⇒ Knoppix, Ubuntu, Debian, SUSE, fedora, CentOS, redhat, ...



Folien aus "Betriebssysteme"-Vorlesung einfügen



# Schnittstellen und Treiber

---

- Damit eine CPU mit den Endgeräten (z.B. Laufwerk) verschiedener Hersteller zusammenarbeiten kann, muss man sich vorher auf eine gemeinsame Schnittstelle verständigt haben.
- Eine Schnittstelle ist eine Konvention, die eine Verbindung verschiedener Bauteile festlegt. -> Gibt es aber auch als Software-Schnittstelle / Bridge
  - ⇒ Man kann sich diesen Sachverhalt am Beispiel der elektrischen Steckdose verdeutlichen.
- Die Schnittstellen in der Informatik definieren nicht nur die räumlichen Ausmaße, sie können auch die Reihenfolge und Konvention des Signal- und Datenaustausches festlegen.
- Treiber unterstützen die Schnittstellen-Problematik auf der Komponenten-Seite:
  - ⇒ Treiber sind allgemeine Übersetzungsprogramme zur Ansteuerung einer Software- oder Hardware-Komponente.
  - ⇒ Treiber ermöglichen einem Anwendungsprogramm die Benutzung einer Komponente ohne den detaillierten Aufbau zu kennen.