



# Einführung Betriebssystemen

# Einleitung – Definition Betriebssystem

- **Verwaltungssicht:** Gesamtheit der Programme, die Systemressourcen verwalten und Anwendungsprogrammen Zugriff darauf ermöglichen
- **Anwendersicht:** Gesamtheit der Programme, die für den Betrieb eines Rechners notwendig sind
- **Abstraktionssicht:** Virtuelle Maschine, die dem Benutzer Abstraktionen der Hardware zur Verfügung stellt, die wesentlich einfacher zu benutzen sind

## Rechnergenerationen:

1. Generation (1945-1960): Röhrenrechner
2. Generation (1960-1970): Transistorrechner
3. Generation (1970-1980): Mikrorechner mit niedrigem Integrationsgrad
4. Generation (1980-1995): Mikrorechner mit hohem Integrationsgrad
- 4.5 Generation (bis heute): Mehrkernsysteme
5. Generation (?): Intelligente Prozessoren

# 1. Generation (1945-1960)

*H. Aiken (Harvard), J. v. Neumann (Princeton) u.a. (USA)*  
*Konrad Zuse (Deutschland)*

- Entwurf, Bau, Programmierung, Betrieb und Wartung eines Systems durch Expertengruppe
- Programmierung durch Verdrahtung, Steckkontakte, Steckkarten, keine höheren Programmiersprachen
- Nutzung für Anwendungen der numerischen Mathematik
- Ab Anfang der 50er Jahre Einlesen per Lochkarten
- Betriebssysteme unbekannt

## 2. Generation (1960-1970)

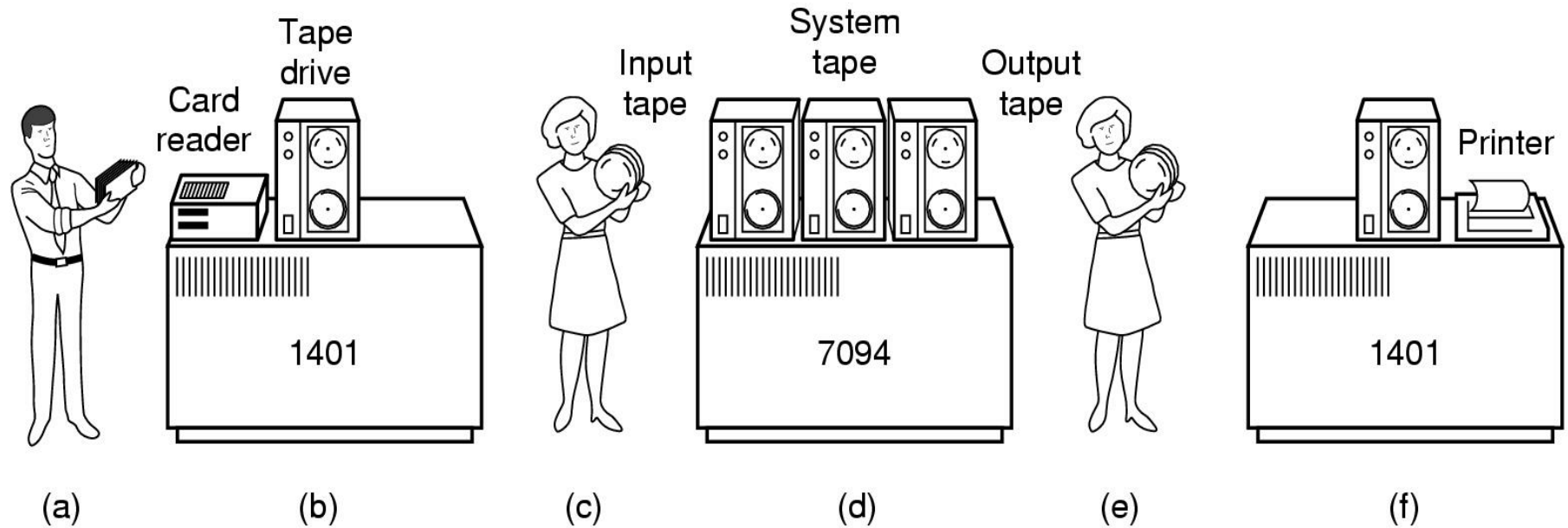
- Nach Einführung von Transistoren wurden Rechner zuverlässiger, Betrieb wurde möglich.
- Unterscheidung zwischen Entwicklern, Herstellern, Operateuren, Programmierern und Wartungspersonal.
- Zunächst Ausführung einzelner Jobs in Form von Lochkartenstapeln mit hohem Anteil an manuellen Arbeiten.
- Rationalisierung durch Einführung des Stapelbetriebs (Batch-System), in dem mehrere ähnliche Jobs gruppiert und abgearbeitet wurden.

# Batchsysteme

- Nutzten oft höhere Programmiersprachen wie FORTRAN.
- Jobs wurden nach Art der verwendeten Programmiersprache gruppiert.
- Anwender geben Lochkartenstapel mit auszuführenden Programm an (menschlichen) Operator. Dieser gruppiert die Jobs und startet den Einlesevorgang.
- Ein fest installiertes Programm, der *Resident Monitor*, steuert die Ausführung des Jobstapels.

➤➤➤➤ **Vorläufer heutiger Betriebssysteme!**

# Batchsysteme

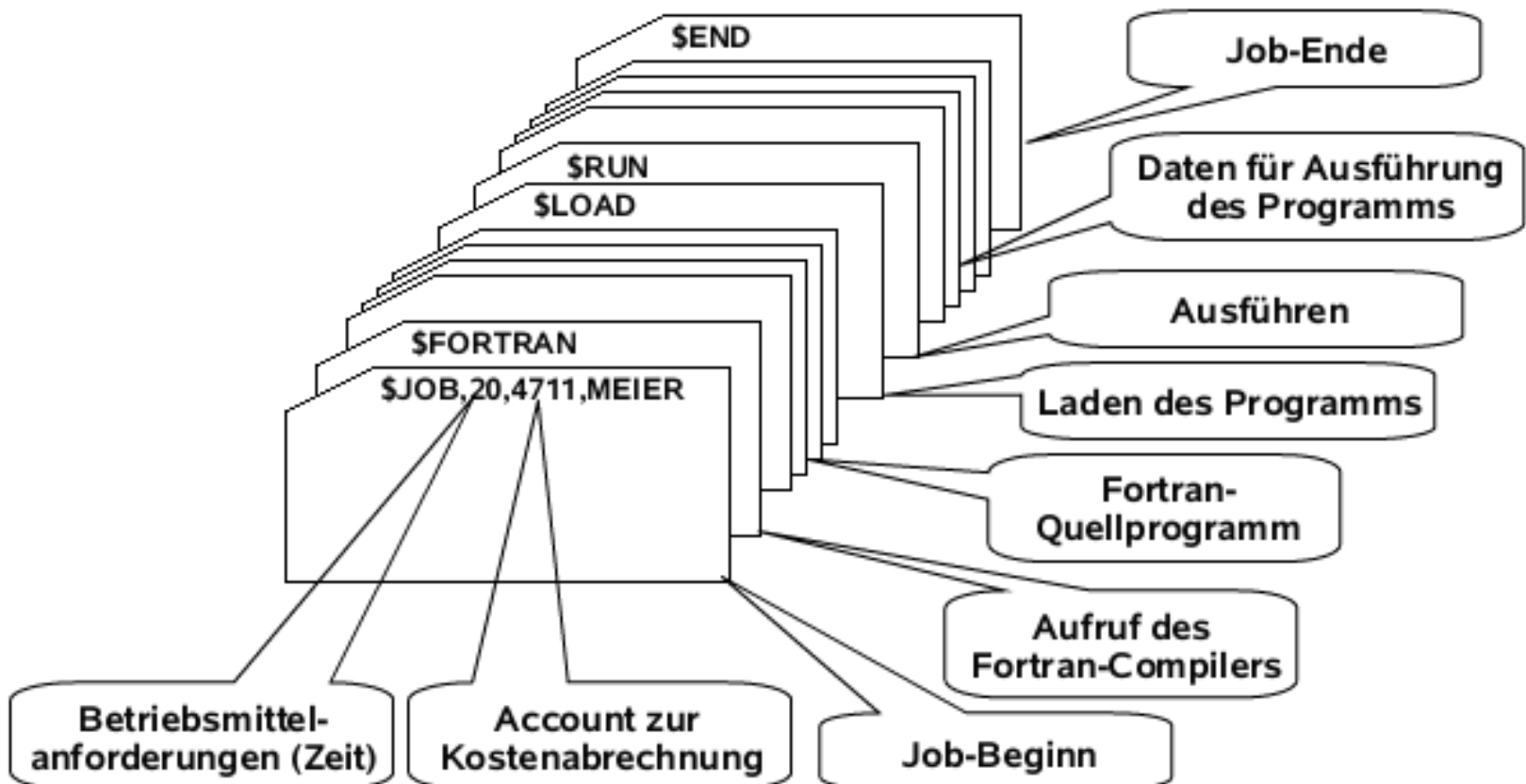


## Batch-System (Beispiel)

- IBM 7094 als Hauptrechner, IBM 1401 für Ein- und Ausgabe
- Eingabe per Lochkarten, Transfer auf Magnetband
- Später ersetzt durch direktgekoppeltes System 7094/7044

# Batchsysteme

- Eigene Kommandosprache (Job Control Language) zur Beschreibung und Steuerung eines einzelnen Jobs
- Vorläufer heutiger Kommandozeileninterpreter





### 3. Generation (1970-1980)

- Sequentielle Jobausführung ineffizient, sobald Ein-/Ausgabeoperationen durchgeführt werden
- **Lösung:** Mehrere Jobs parallel im Speicher vorhalten. Sobald einer dieser Jobs auf Ein-/Ausgabe wartet, wird der nächste Job bedient (Multiprogramming/ Multitasking).
- Betriebssystem organisiert unter anderem
  - Umschaltung zwischen den Jobs
  - Speicherzuteilung



**Speicher-  
organisation**

## 4. Generation (1980 – 1990)

- Vom Mainframe zum Personal Computer
- Lokale Netzwerke zur Kommunikation und Kooperation
- Graphische Ein-/Ausgabe führt zu bedienfreundlicheren Benutzerschnittstellen

Mit diesen Möglichkeiten wächst auch der Aufgabenbereich von Betriebssystemen stetig

## 4.5 Generation (bis heute)

### **Steigende Komplexität und neue Funktionalitäten:**

- Verteiltheit (Client/Server),
- Heterogenität,
- Skalierbarkeit
- Mehrkernsysteme,
- Sicherheit (Datenschutz, Integrität)
- Fehlertoleranz / Robustheit
- Multimedia

# Aktuelle Entwicklungsrichtungen

- Verteilte Betriebssysteme
- Massives Multiprocessing
- Standardisierung von Schnittstellen
- Virtualisierung
- Konfigurierbarkeit
- Sicherheit

# Klassifikation nach Aufgaben

(Beispielbetriebssysteme in Klammern)

- **Mainframe**-Betriebssysteme (*OS2200*)
- **Server**-Betriebssysteme (*SCO OpenServer*)
- **Endanwender**-Betriebssysteme (*Windows 8*)
- **Smartphone**-Betriebssysteme (*Android*)
- **Echtzeit**-Betriebssysteme (*IOS*)<sup>1</sup>
- Betriebssysteme für **eingebettete Systeme** (*Nucleus*)
- **Chipkarten**betriebssysteme (*TCOS*)

---

<sup>1</sup>) *Internet Operating System* von Cisco, nicht zu verwechseln mit iOS von Apple

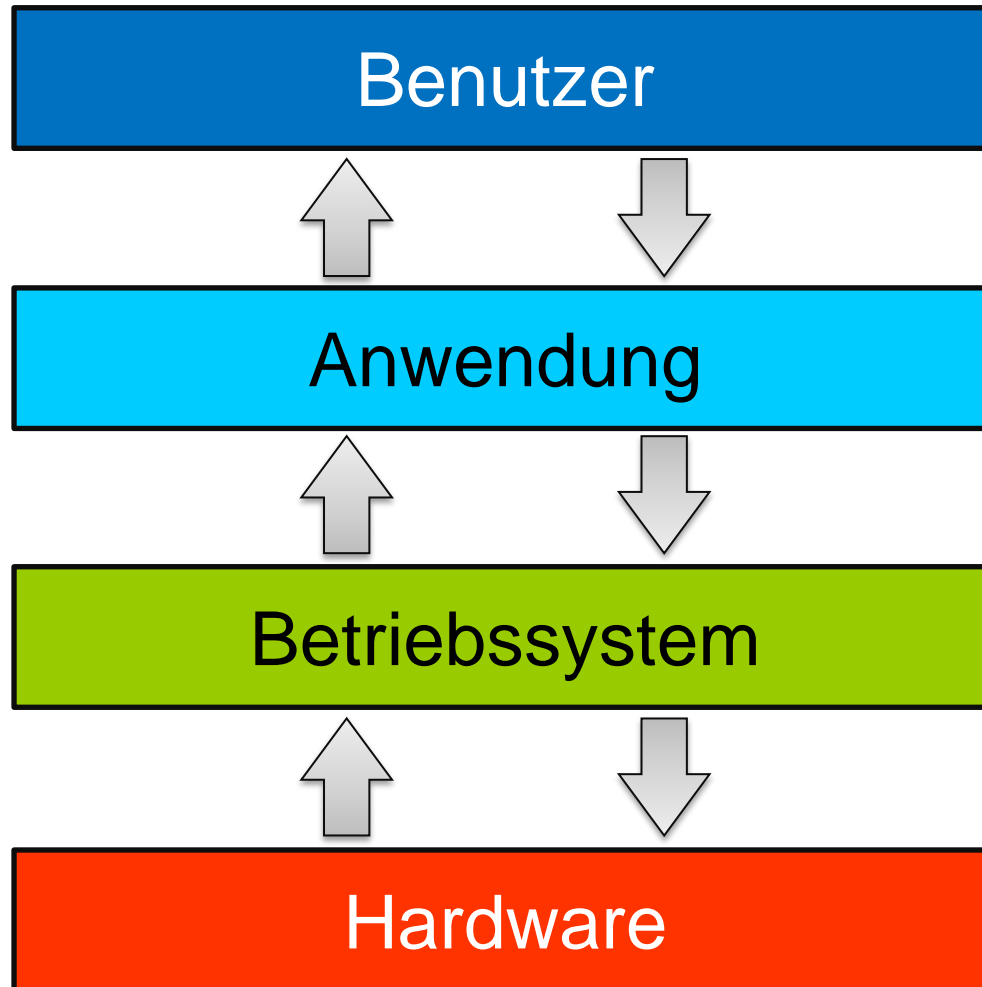
# Zurück zur Einleitung – Definition Betriebssystem

**Verwaltungssicht:** Gesamtheit der Programme, die Systemressourcen verwalten und Anwendungsprogrammen Zugriff darauf ermöglichen

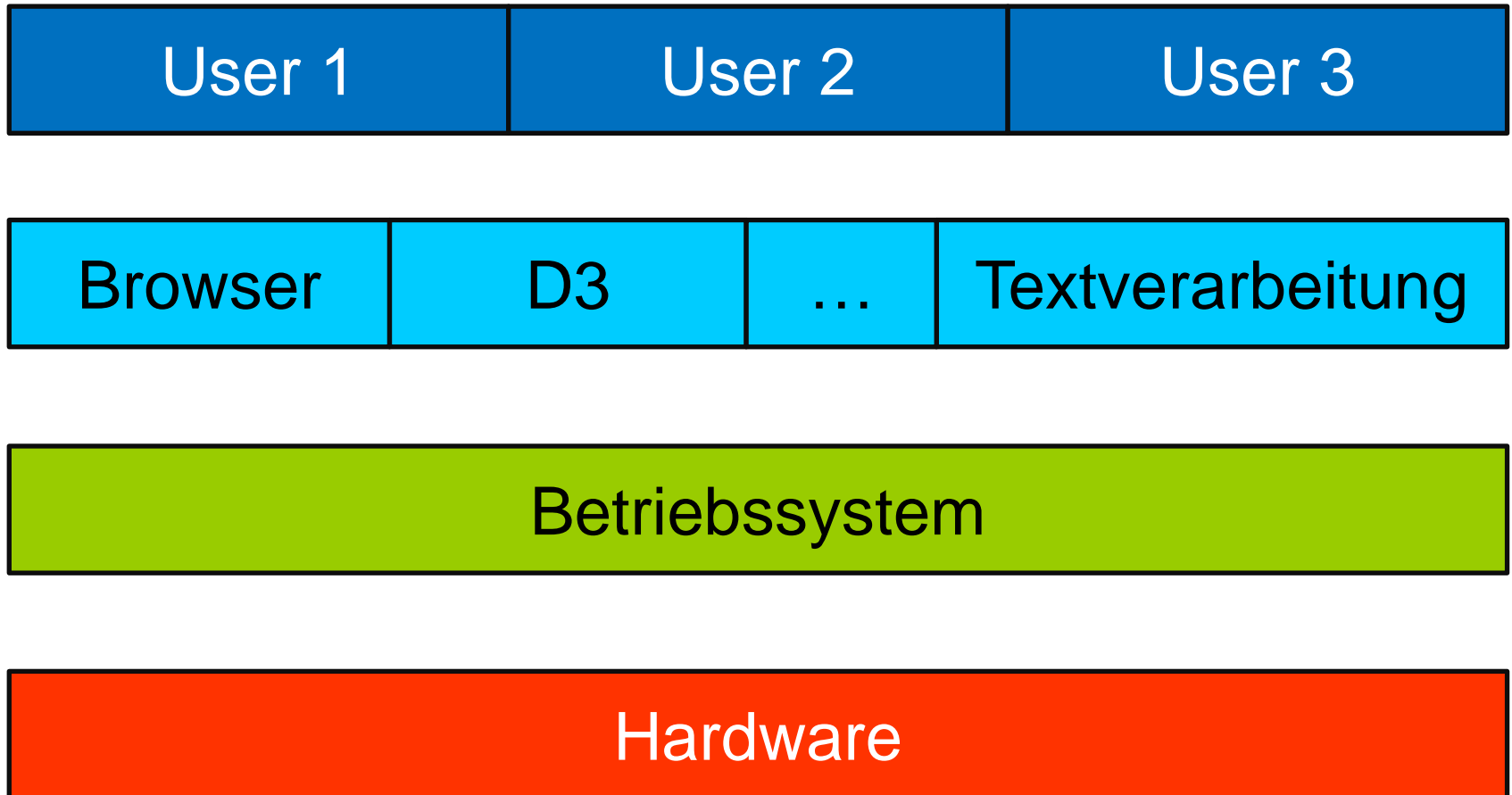
**Anwendersicht:** Gesamtheit der Programme, die für den Betrieb eines Rechners notwendig sind

**Abstraktionssicht:** Virtuelle Maschine, die dem Benutzer Abstraktionen der Hardware zur Verfügung stellt, die wesentlich einfacher zu benutzen sind

# Schichtenmodell

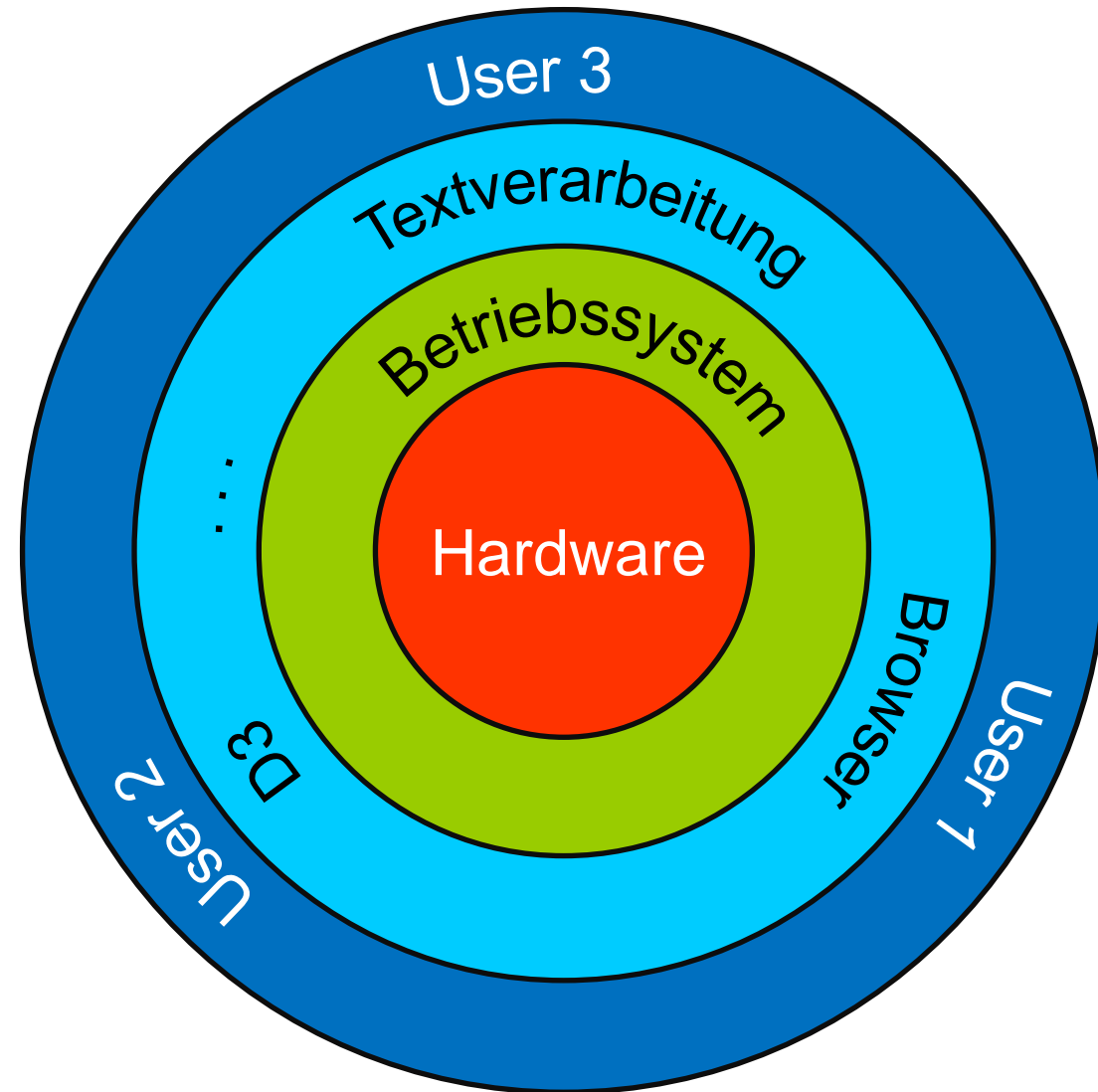


# Schichtenmodell





# Zwiebelschalenmodell

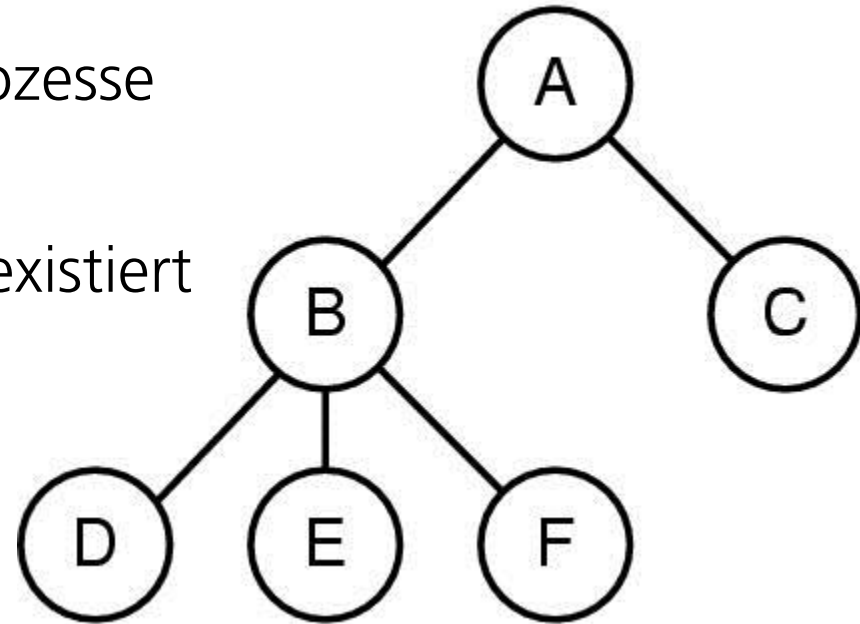


- **Schichtenmodell** stellt Schnittstellen in den Vordergrund
- **Zwiebelschalenmodell** bringt Kapselung zum Ausdruck

- Ein **Prozess** ist ein eindeutig identifizierbarer *Ablauf eines Programms*, d.h. ein Programm in Ausführung mit allen dazu notwendigen Daten.
- Der **Prozessor** (engl.: Central Processing Unit, CPU) führt die Anweisungen eines Prozesses nacheinander aus. Er kann aus mehreren Kernen bestehen, je Kern kann genau ein Prozess ausgeführt werden.

# Kindprozesse und Prozessbaum

- Prozesse können selbst neue Prozesse erzeugen (Kindprozesse).
- Beim Start des Betriebssystems existiert nur ein einziger Prozess (*root process*).
- Dieser erzeugt schrittweise Kindprozesse, die wiederum selbst Kindprozesse erzeugen können, es entsteht ein Prozessbaum<sup>\*)</sup>.



<sup>\*)</sup> Windows kennt keine solche Prozesshierarchie

- Die zentralen Komponenten des Betriebssystems werden als Betriebssystem**kern** (engl.: Kernel) bezeichnet.
- Die Aufgaben des Kerns umfassen im Wesentlichen:
  - Direkte Kommunikation mit der **Hardware** (durch Gerätetreiber)
  - **Reihenfolgeplanung** der Ausführung (Scheduling) und **Umschalten** (Dispatching) von Prozessen
  - **Speicherverwaltung**

# Priviligierungsstufen: Kernel- und Benutzermodus

Die Verwendung von Betriebssystemen kann durch die Architektur des Rechners unterstützt werden. So bietet bspw. die **CPU** verschiedene **Priviligierungsstufen** für laufende Prozesse. Mindestens sind dies

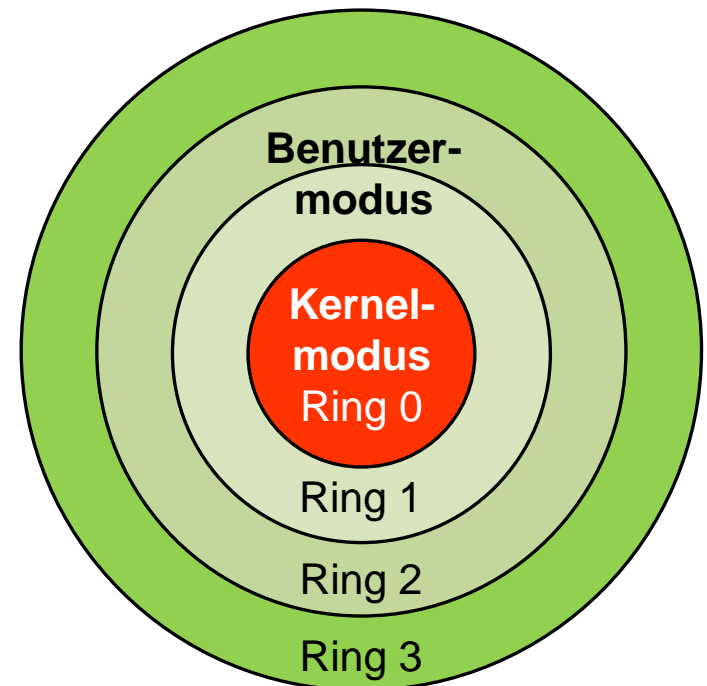
- **Kernelmodus** (*kernel mode, system mode*): Modus mit Zugriff auf privilegierte Befehle, reserviert für die Kernelkomponenten des Betriebssystems.
- **Benutzermodus** (*user mode*): Gilt für alle anderen Prozesse. Ihnen steht nur ein eingeschränkter Befehlssatz zur Verfügung.

# Priviligierungsstufen: Beispiel x86 Architektur

X86-kompatible CPUs (also faktisch alle gängigen PCs) besitzen insgesamt **4 Priviligierungsstufen** (Ring 0 bis 3) mit absteigendem Umfang des verfügbaren Befehlssatzes.

**Ring 0** entspricht dem Kernelmodus (*supervisor mode*, voller Befehlssatz), **Ring 1-3** sind dem Benutzermodus zugeordnet.

In der Praxis werden Ring 1 und 2 selten genutzt (Ausnahme: Virtualisierung).



# Grundlegende Kernelarchitekturen (Grafik: Wikipedia)

auf monolithischem Kernel  
basierte Betriebssysteme

Anwender

Anwendungen



VFS, System-Aufrufe

IPC, Dateisystem

Scheduler, Virtual Memory

Geräte-Treiber, Dispatcher, ...

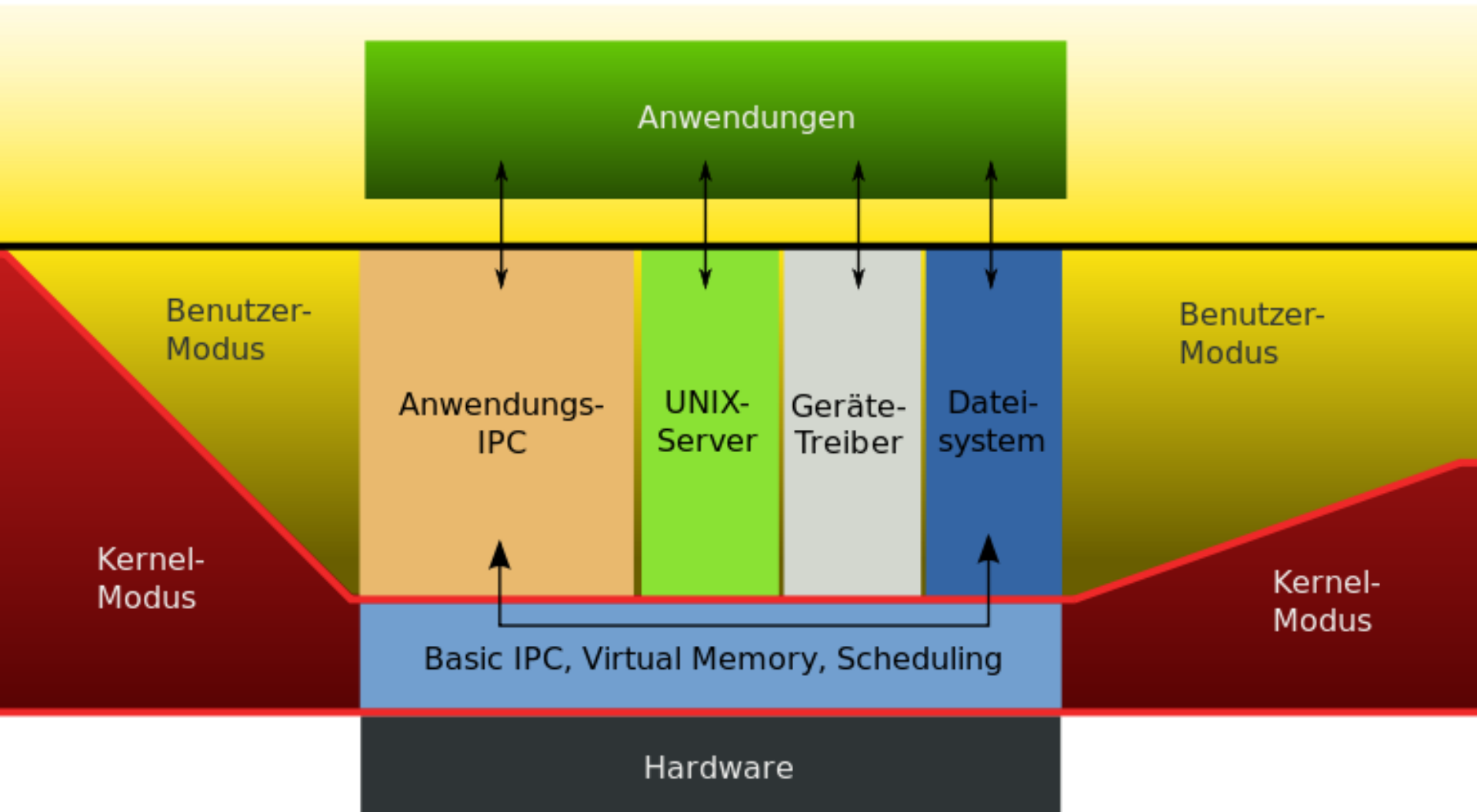
Kernel-  
Modus

Hardware

Betriebssystem

# Grundlegende Kernelarchitekturen (Grafik: Wikipedia)

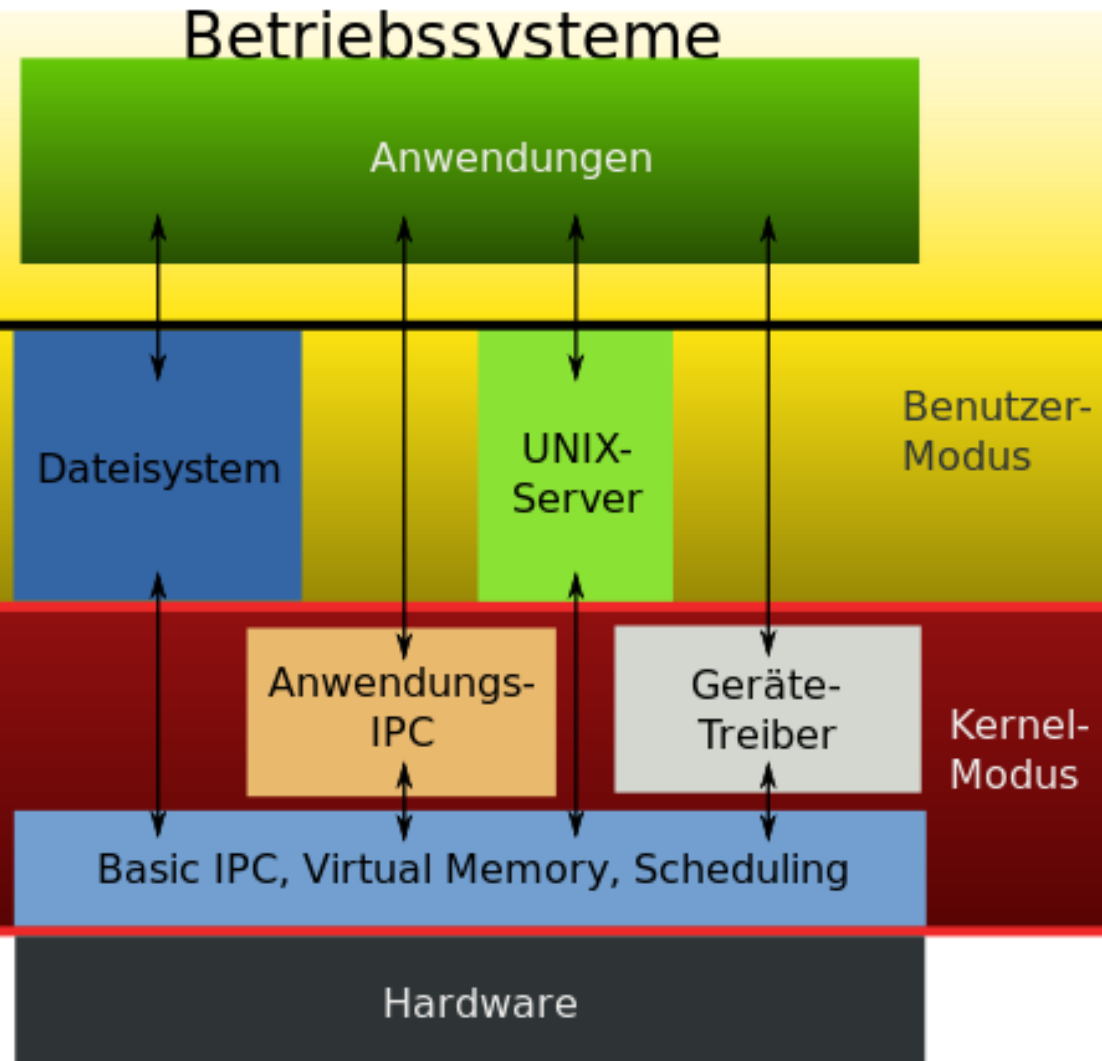
## Mikrokernel-basierte Betriebssysteme



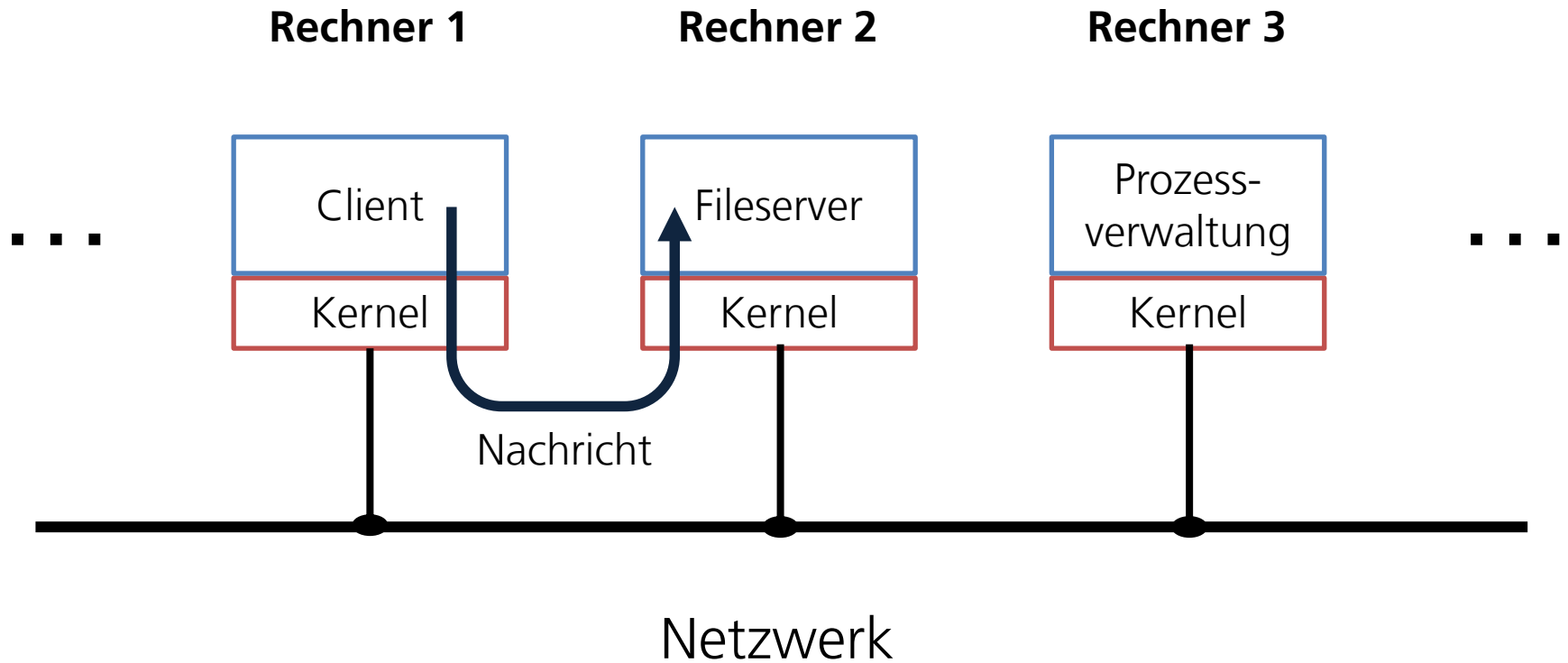


# Grundlegende Kernelarchitekturen (Grafik: Wikipedia)

## Hybridkernel-basierte



# Netzwerkbetriebssystem



# Erneut zur Definition Betriebssystem

**Verwaltungssicht:** Gesamtheit der Programme, die Systemressourcen verwalten und Anwendungsprogrammen Zugriff darauf ermöglichen

**Anwendersicht:** Gesamtheit der Programme, die für den Betrieb eines Rechners notwendig sind

**Abstraktionssicht:** Virtuelle Maschine, die dem Benutzer Abstraktionen der Hardware zur Verfügung stellt, die wesentlich einfacher zu benutzen sind

## 1. Benutzerschnittstelle

grafische Oberflächen, Kommandointerpreter

## 2. Dienstprogramme

Konfigurationsprogramme, Editoren, Dateizugriffe etc.  
aber bspw. auch Compiler zur Programmierung

## 3. Bibliotheken

Interaktion von Programmen mit Betriebssystem,  
unter anderem auch für grafische Oberflächen

# Erneut zur Definition Betriebssystem

**Verwaltungssicht:** Gesamtheit der Programme, die Systemressourcen verwalten und Anwendungsprogrammen Zugriff darauf ermöglichen

**Anwendersicht:** Gesamtheit der Programme, die für den Betrieb eines Rechners notwendig sind

**Abstraktionssicht:** Virtuelle Maschine, die dem Benutzer Abstraktionen der Hardware zur Verfügung stellt, die wesentlich einfacher zu benutzen sind

# Schnittstellen

**(Eine) Definition:** Standardisierter Austausch von Daten und/oder Anweisungen zwischen Prozessen und/oder Komponenten.

- **Passive Schnittstelle:** Nur Datenaustausch zwischen den beteiligten Prozessen/Komponenten.
- **Aktive Schnittstelle:** Schnittstelle führt selbst Aktionen aus, bspw. beim Übergang von Soft- zu Hardware.

**Ziel<sup>\*)</sup>: Abstraktion** von Implementierungsdetails.

---

<sup>\*)</sup> Es gibt noch weitere Ziele, die wir an dieser Stelle aber nicht diskutieren

# Abstraktion

Schnittstellen verbergen Implementierungsdetails vor dem Aufrufenden (Kapselung). Dies ermöglicht Standardisierung des Aufrufs und Abstraktion von Komponenten.

## Beispiel: Datenträger

- Abstrakte Funktionalität: Lesen und Schreiben von adressierbaren Daten
- Schnittstelle zum Lesen benötigt vom Aufrufenden nur Informationen darüber, **welche** Daten gelesen werden sollen (Adresse. Umfang). Die Details der Implementierung des Gerätes bleiben verborgen.

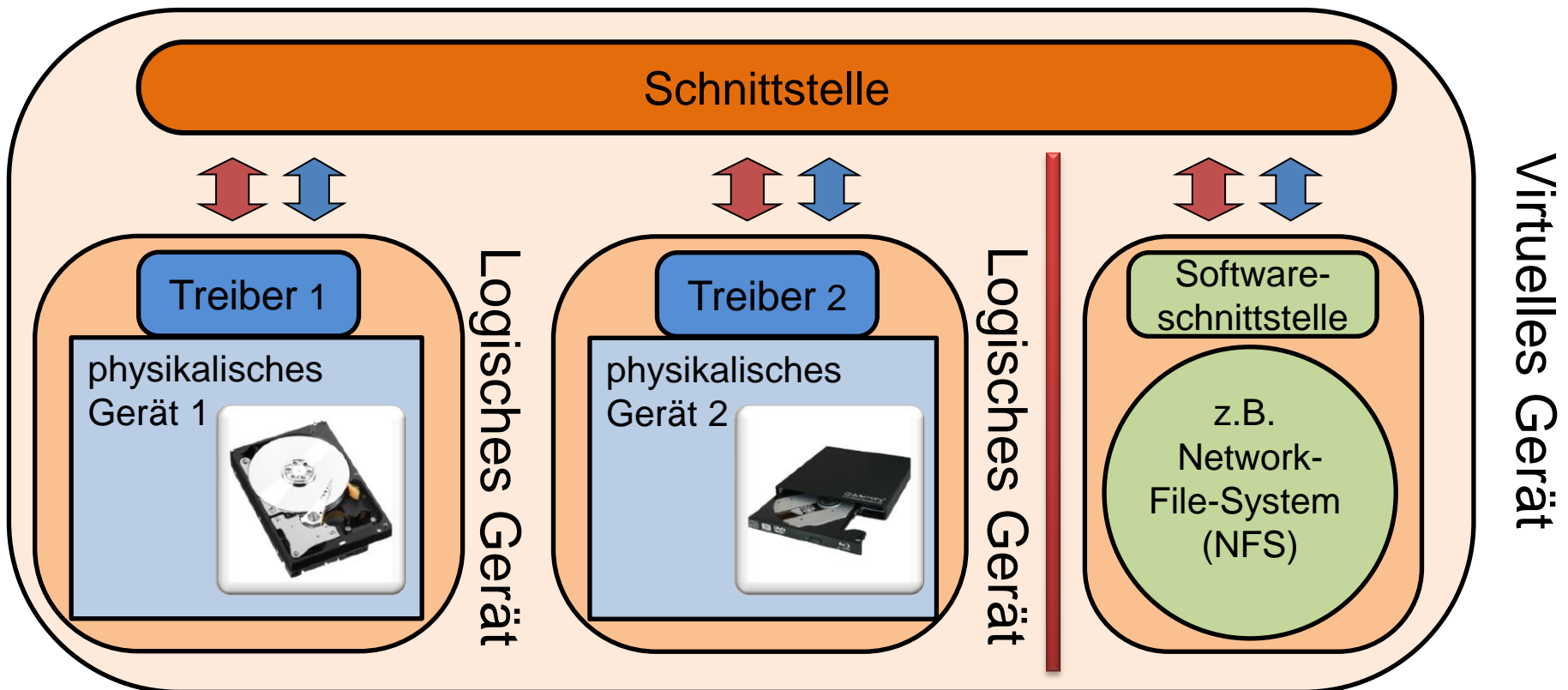
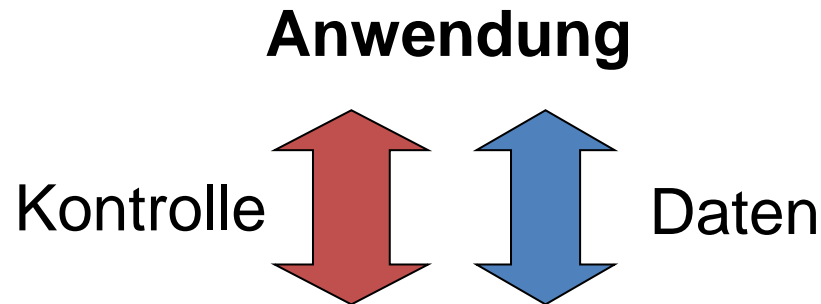
Schnittstelle zu einem Gerät wird durch sogenannten **Gerätetreiber** als Teil des Betriebssystems bereitgestellt. Dieser kennt die interne Implementierung des Gerätes.

## Abstraktionsstufen:

- **Logisches Gerät** – Schnittstellenrepräsentation eines realen (dinglichen) Gerätes. Kommunikation über Gerätetreiber
- **Virtuelles Gerät** – Abstrakte Schnittstellenrepräsentation einer **Funktionalität**, unabhängig davon, ob diese dinglich oder virtuell implementiert ist. Kommunikation über Softwareschnittstelle.



# Abstraktion



# Zusammenfassung

- Betriebssysteme übernehmen Aufgaben der Verwaltung, der Interaktion mit dem Benutzer und dem Schutz von Hard- und Software.
- Je nach Fokus können die Aufgaben des Betriebssystems aus Verwaltungs-, Anwender- und Abstraktionssicht klassifiziert werden.
- Laufende Programme werden als Prozesse bezeichnet und können ggf. selbst weitere Prozesse erzeugen.
- Der Kern des Betriebssystems läuft üblicherweise im privilegierten Modus und hat als einziger volle Kontrolle über Hard- und Software.