

## Vorlesung

# Betriebssysteme

## Teil 2

### Einführung und Prozesse

## Einführung: Historie

### Historie von Betriebssystemen:

- erste Rechnergeneration (ca. 1945-1955):
  - **Kein** Betriebssystem („*Single Purpose Computers*“),
  - Programmierung über Steckbrett oder Lochstreifen... keine Programmiersprachen.
- zweite Generation (ca. 1955-1965):
  - **Stapelverarbeitung**, einfache Job Control.
  - Programmiersprachen: Assembler, Fortran...
- dritte Generation (1965-1980):
  - Mehrbenutzer- und **Multiprogrammbetrieb**, Steuerung über Terminal (Tastatur und Bildschirm).
  - Programmiersprachen: C, Pascal...
- vierte Generation (ab ca. 1975):
  - **Interaktive** Systeme mit grafischer Benutzeroberfläche, verteilte Betriebssysteme  
Netzwerkbetriebssysteme,
  - Multiprozessorsysteme. Objektorientierte Programmiersprachen

## Microsoft und Windows

- Gründung Microsoft 1975
- Erfolg: MS-DOS („the day Gary Kildall went flying“)
- 100.000 Mitarbeiter, 75 Mrd \$ Umsatz

**MICROSOFT**

- 95% Marktanteil auf PCs und Notebooks



## Unix



Bell Labs:

**Ken Thompson**

(1. Unix Version 1969,  
Sprache Assembler,  
später B),

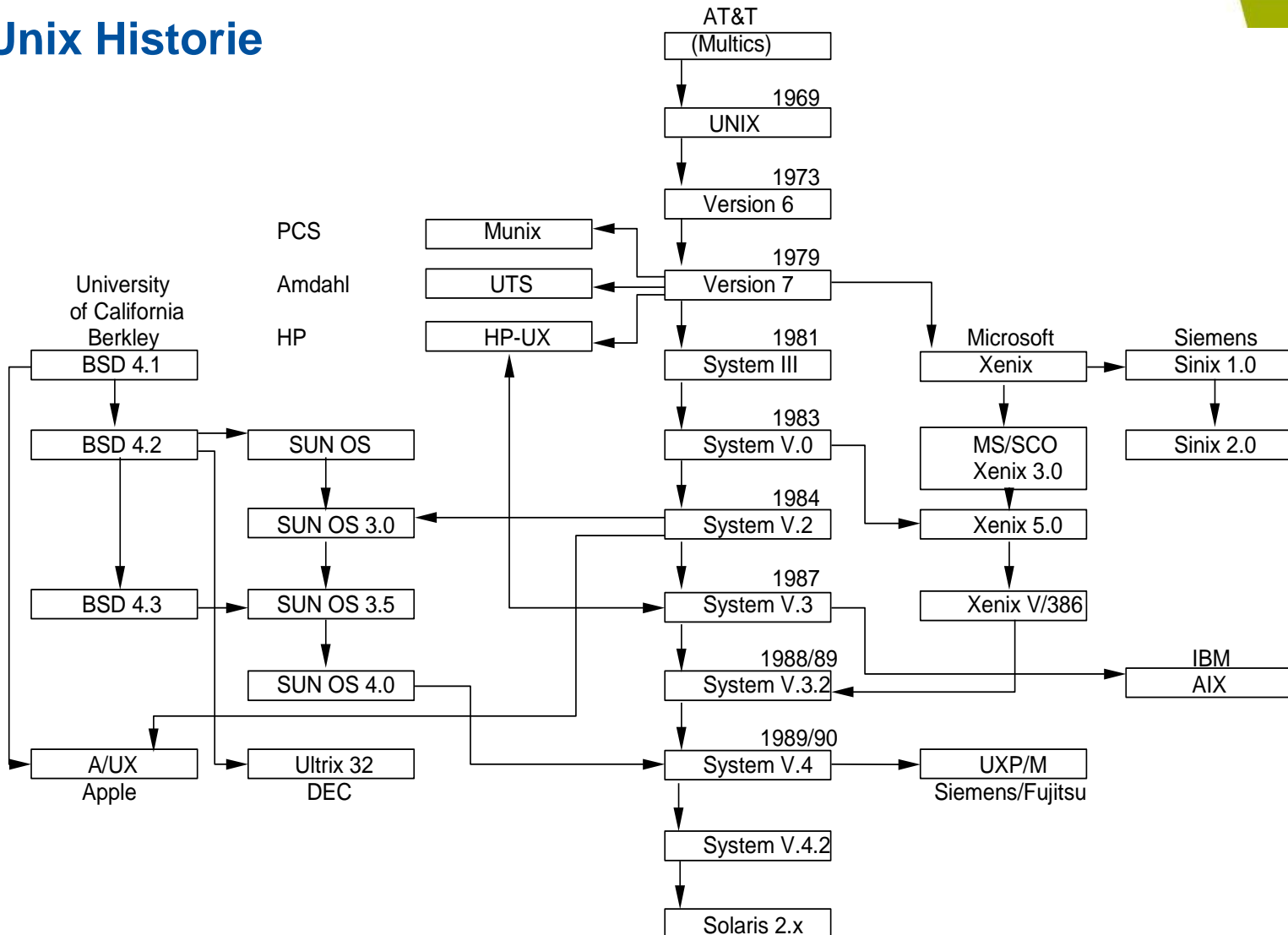
**Dennis Ritchie**

(Weiterentwicklung der  
Sprache B zu C),

**Brian Kernighan**

Bild: Dennis Ritchie (stehend)  
und Ken Thompson bei der  
UNIX-Portierung auf die  
PDP-11 an 2 Teletype 33  
Terminals (1970)

# Unix Historie



## Linux

- Linus Torvalds 1991
  - Unix Betriebssystem für IBM PC
  - Tanenbaums Minix
- Zukunftsperspektive für Linux
  - Webserver (Microsoft Anteil 20%)
  - Embedded Systems (Microsoft Anteil < ??%)



## Linux auf Embedded Systems



# Smartphones

## ■ Smartphones

### ■ Umfrage Betriebssysteme WS 2012/2013

■ Windows	2
■ Android	25
■ iOS	4
■ Symbian	0
■ Palm OS	0

### ■ Umfrage Betriebssysteme SoSe 2014

■ Windows	2
■ Android	22
■ iOS	8
■ Symbian	0
■ Palm OS	0

### ■ Umfrage Betriebssysteme SoSe 2018

■ Windows	0
■ Android	22
■ iOS	9
■ Symbian	0
■ Palm OS	0

### ■ Umfrage Betriebssysteme SoSe 2015

■ Windows	1
■ Android	30
■ iOS	4
■ Symbian	0
■ Palm OS	0

### ■ Umfrage Betriebssysteme SoSe 2016

■ Windows	3
■ Android	30
■ iOS	4
■ Symbian	0
■ Palm OS	0



## Einführung: Arten

### Einteilung von Betriebssystemen:

- Nach Betriebsart:
  - Stapelverarbeitung (batch processing)
    - Programme werden einzeln gestartet und verarbeitet.
    - Klassische Großrechnerbetriebssysteme.
  - Time-Sharing-Betriebssystem
    - Die Rechenleistung wird in Zeitscheiben aufgeteilt.
    - Interaktives Arbeiten (auch mehrerer Benutzer) ist möglich.
  - Echtzeitbetriebssystem
    - Garantierte Antwortzeiten können für Prozesse angegeben werden.
- Nach Rechnerarchitektur
  - Einprozessorsystem
  - Multiprozessorsystem
  - Verteiltes System (Rechenknoten haben eigene CPU und eigenen Speicher)

## Einführung: Arten (Forts.)

### Einteilung von Betriebssystemen (Forts.):

- Nach *Einsatzgebiet*
  - **Großrechnersysteme**
    - Bsp. z/OS (IBM)
  - **Server**
    - Solaris (SUN), Linux, BSD, Windows Server (Microsoft)
  - **Multiprozessorsysteme**
    - spezielle Varianten von Windows, UNIX (Solaris)
  - **Personalcomputer**
    - Windows (XP, Vista, 7, 10), MacOS, Linux
  - **Handheldcomputer / Smartphones:**
    - Android, iOS, Symbian, Blackberry, Windows Mobile, Palm OS
  - **Eingebettete Systeme:**
    - QNX, VxWorks
  - **Sensorknoten:**
    - TinyOS, Java VM

## Einführung: Modi

### Kern- und Benutzermodus:

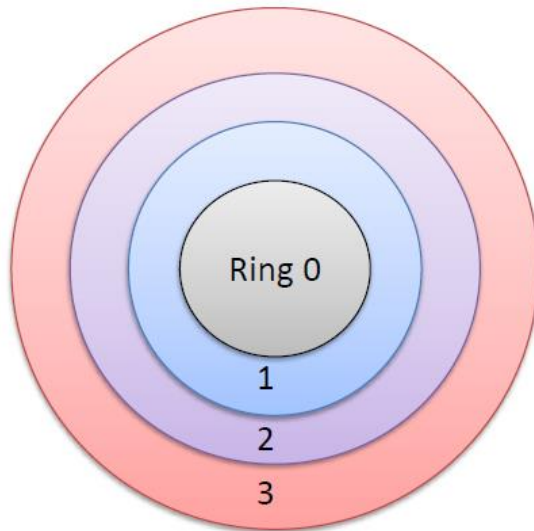
- Zum Schutz des Betriebssystems (und damit des Rechners) vor fehlerhaften (oder böswilligen) Applikationen und zum Schutz der Applikationen untereinander haben moderne leistungsfähige Prozessoren ein Privilegiensystem
  - Betriebssystem läuft im **Kernmodus** (*kernel mode, supervisor mode*),
  - die Applikationen im **Benutzermodus** (*user mode*)

Privileg	Modus	
	Benutzermodus	Kernmodus
Ausführbare Maschinenbefehle	begrenzt	alle
Hardwarezugriffe	nein	Vollzugriff
Zugriff auf MMU	nein	ja
Zugriff auf Systemcode bzw.Daten	Keiner oder nur lesend	ja

## Einführung: Schutzebenen

### Schutzebenen des Pentium:

- Die Schutzstufe eines Programms ist im **PSW** codiert.
- Wechsel zwischen den Schutzstufen (Ringen) nur mit CALLS möglich. Kein direkter Einsprung in niedrigeren Ring!



Schutzstufe:	Typische Nutzung:
Ring 0	Betriebssystems- kern ( <i>Kernel</i> )
Ring 1	Systemaufrufe
Ring 2	Shared Libraries
Ring 3	Benutzer- programme

## Einführung: Systemaufrufe

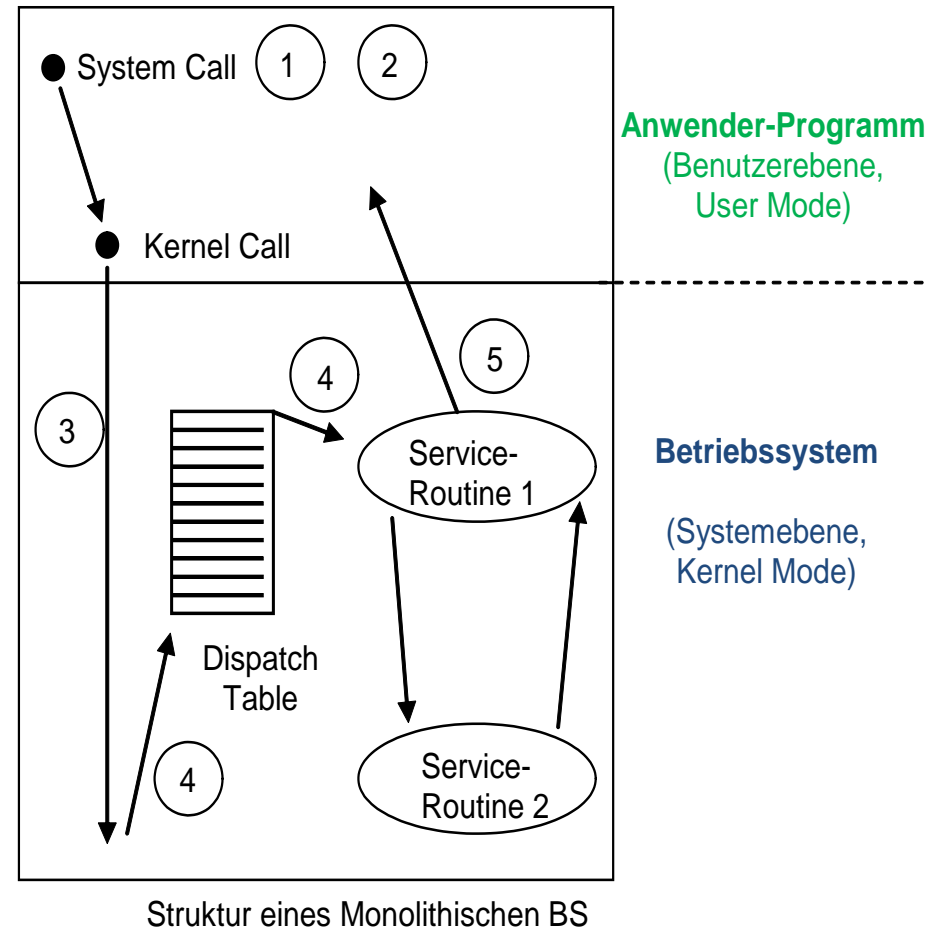
### Betriebssystemaufrufe

Um aus dem Benutzermodus in den Kernmodus zu gelangen, gibt es folgende Möglichkeiten:

- **Hardware Unterbrechung** (*HW Interrupt*) , z.B.
  - Echtzeit-Uhr (Timer)
  - Anforderung eines E/A Gerät
  - Hardware Fehler (z.B. Stromversorgung, Speicherfehler...)
- **Software Unterbrechung** (*SW Interrupt, Trap*)
  - System Aufruf (*System Call, Supervisor Call*)
  - Software Fehler (z.B. Zugriff auf ungültige Adresse, ungültiger Befehl, Division durch 0 ...)

## Einführung: Systemaufrufe (Forts.)

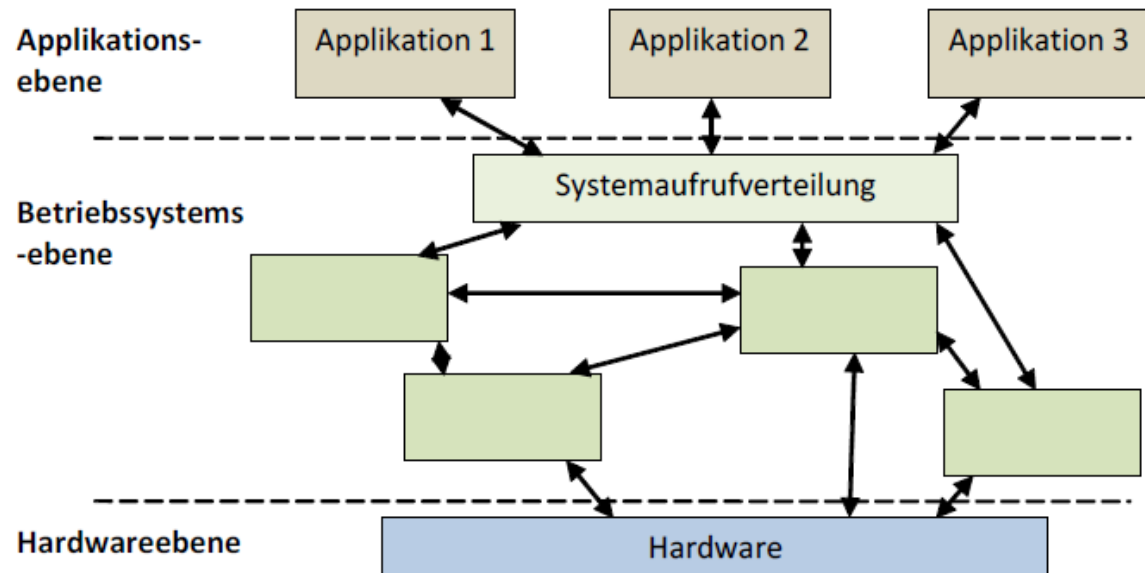
- Aufteilung in **Benutzerebene** (*user*) und **Systemebene** (*kernel*)
- Ablauf:
  1. Anwender-Programm benötigt einen BS-Service: **System Call**.
  2. Parameter werden in Übergabebereich platziert.
  3. Steuerung wird an den Systemkern übergeben: **Kernel Call** (auch: Supervisor Call).
  4. **Kernel**: identifiziert **Service-Routine** und ruft sie auf.
  5. **Service-Routine** läuft ab und gibt Ergebnis an den Auftraggeber (Anwender-Programm) zurück.
- Zweiteilung (*user* - *kernel*) ergibt ungenügende Strukturierung.



## Einführung: Aufbau (Forts.)

### Aufbau von Betriebssystemen:

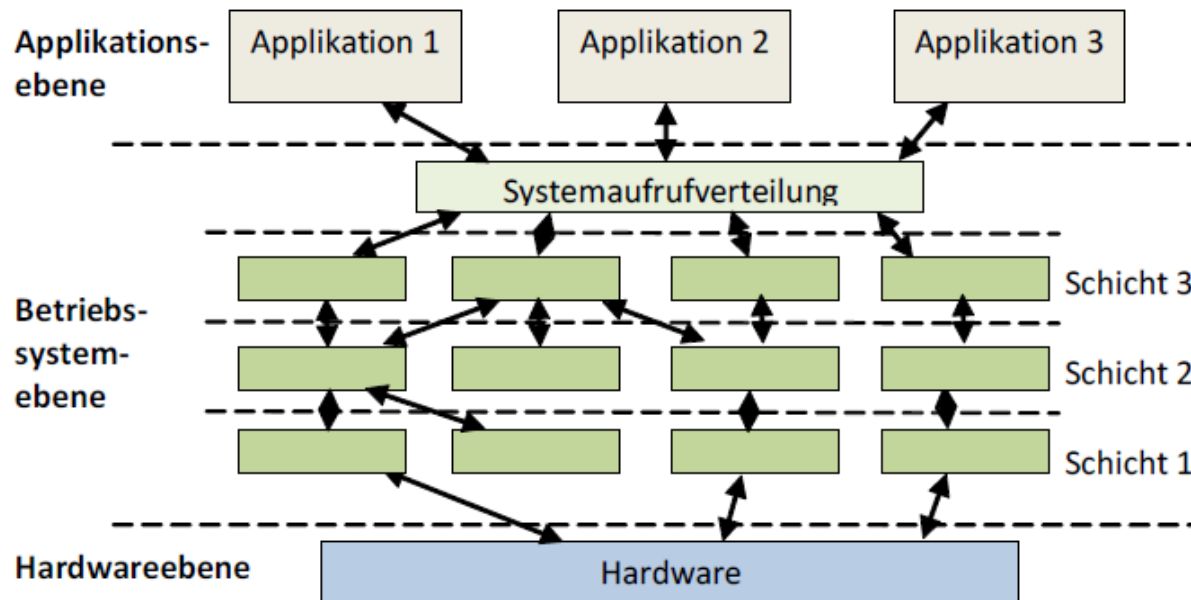
- *Monolithisches System:*
  - Das Betriebssystem ist eine Sammlung von Prozeduren, die sich gegenseitig aufrufen.
  - Jede Prozedur kann beliebige Datenstrukturen verändern.
  - Beispiele: MS-DOS, CMX-RTX (embedded RTOS), ältere Unix Varianten



## Einführung: Aufbau (Forts.)

### Aufbau von Betriebssystemen:

- *Geschichtetes System:*
  - Die Teilfunktionen sind hierarchisch gegliedert.
  - Eine horizontale Kommunikation ist nicht möglich.
  - Beispiele: OS/2, moderne Unix-Varianten





## Testfrage

- Zusammenarbeit zwischen Anwenderprogramm und BS: Welches sind die Ziele des Kernel-Call-Mechanismus?
- Mögliche Antworten:

Erhöhung der Geschwindigkeit

Vereinfachung der Programmierung

Schutz von Speicherbereichen

Einsparung von Energie

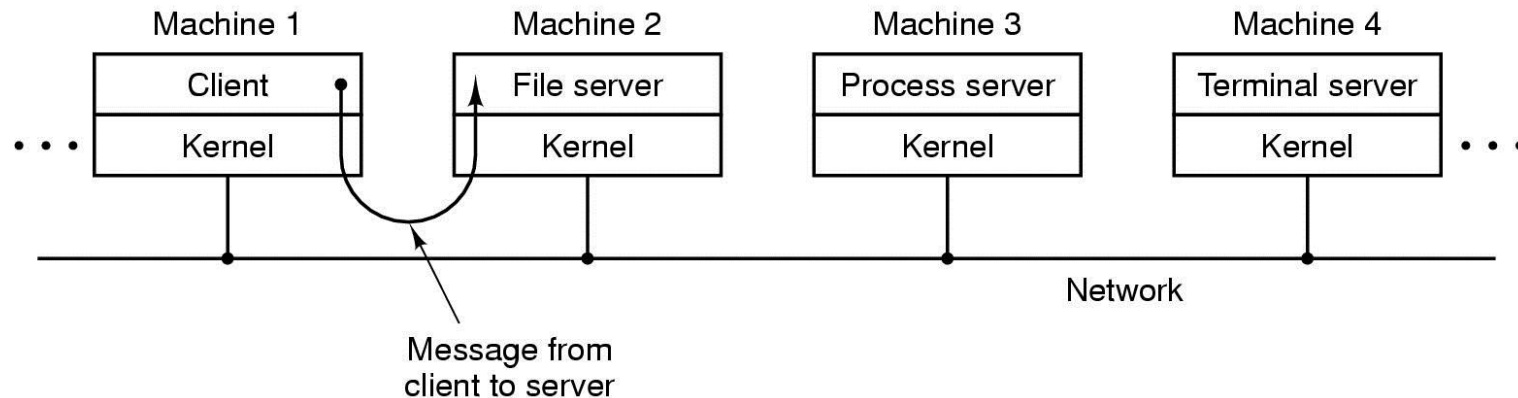
Senkung der Taktfrequenz

Entkopplung von User und System

## Einführung: Aufbau (Forts.)

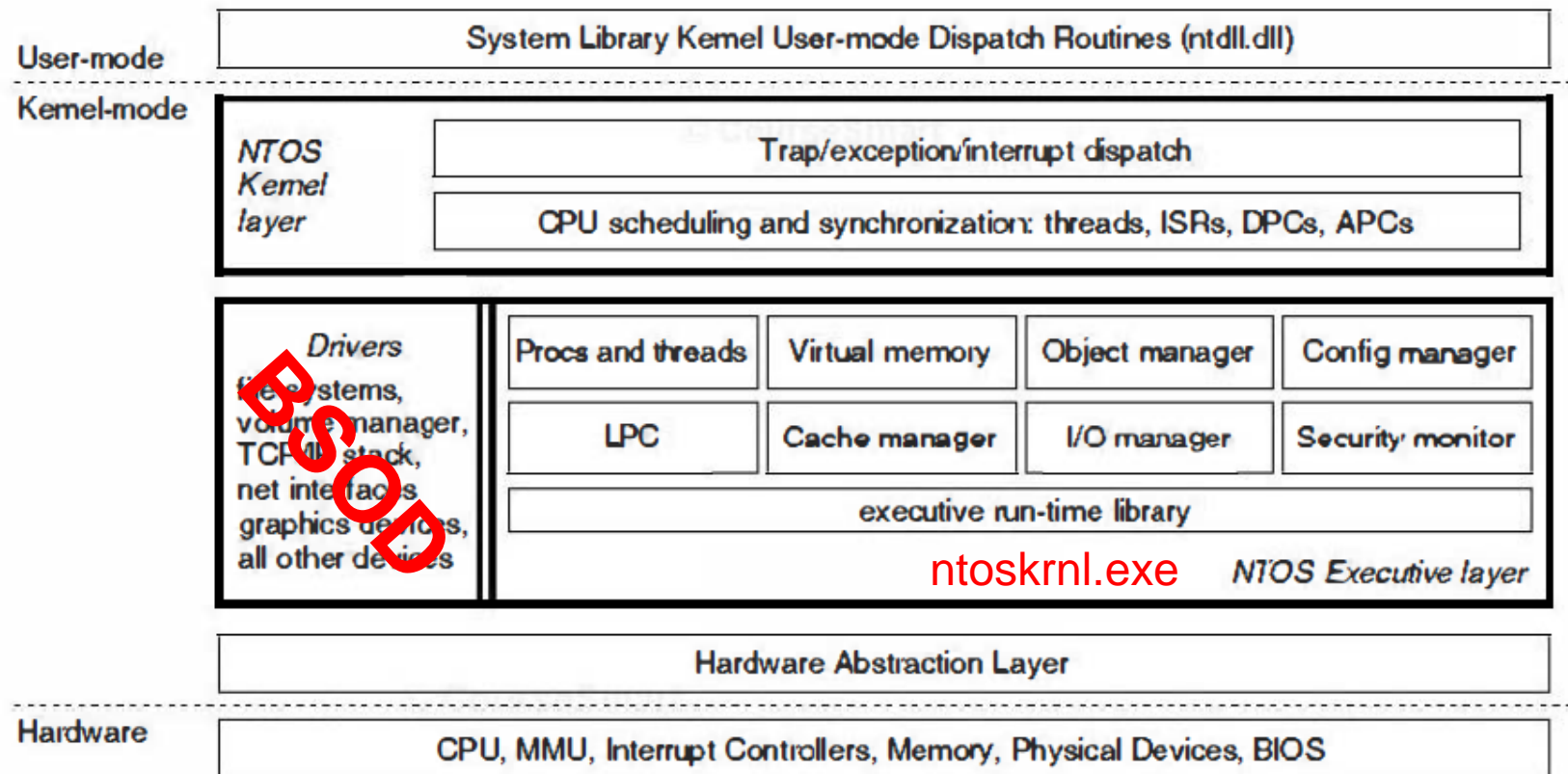
### Aufbau von Betriebssystemen

- Client-Server-Modell:
  - Server können auf verteilten Systemen realisiert werden.



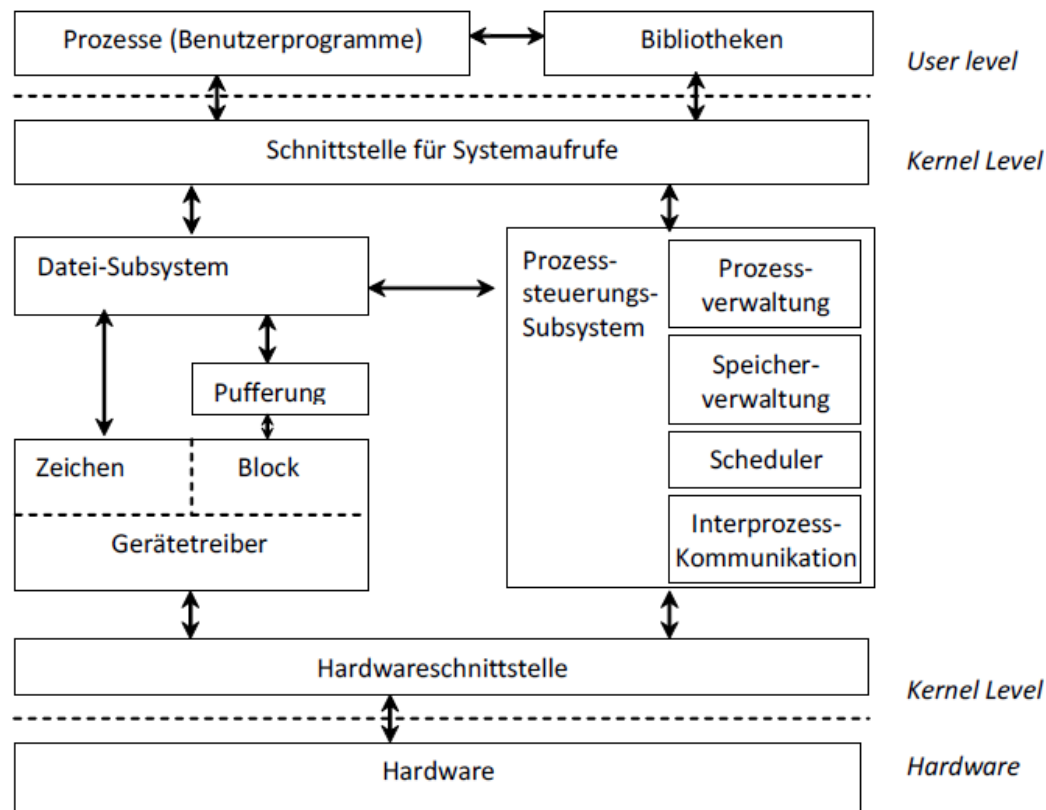
## Einführung: Aufbau (Forts.)

Aufbau von Windows NT/2000/XP/Vista (nach Tanenbaum):



## Einführung: Aufbau (Forts.)

### Aufbau von Unix (Kern des System V Release 3):



## Testfrage

- Nennen Sie die Basis- oder Kernelfunktionalitäten eines BS!
- Mögliche Antworten:

Prozessmanagement

Dateimanagement

Behandlung von Echtzeitereignissen

Bestimmung der Prozessreihenfolge

Userverwaltung

Verwaltung des virtuellen Speichers

Login

Nachrichtentransport

## Motivation & Ziele

### Motivation:

- Software für Rechensysteme wird unterteilt in zwei Gruppen:
  - **Systemprogramme** zum Betrieb des Computers.
  - **Anwenderprogramme** erfüllen die Anforderungen der Anwender.
- Das Betriebssystem (*operating system*) ist das **wichtigste Systemprogramm**.

### Ziele der Veranstaltung: Vermittlung von

- **Wissen:**
  - Aufbau von Betriebssystemen
  - Grundlegende Konzepte (u.a. nebenläufige Prozesses, Multi-Threading,...)
- **Praktischen Kenntnissen:**
  - Analyse und Konzeption von Betriebssystemkomponenten
  - Realisierung in **C**
  - Programmierung in der Shell
  - **Keine** Klickanleitung „wie installiere ich Windows“

## Grundvoraussetzungen

### Grundlegende Kenntnisse:

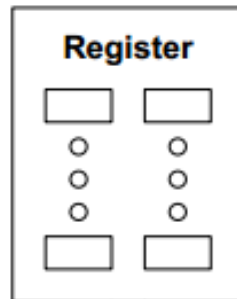
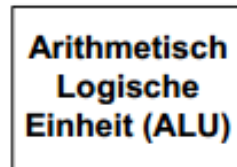
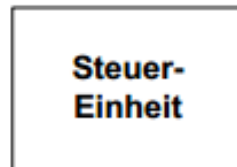
- Aufbau von Rechensystemen (CPU, Speicher, E/A,...)
  - Was ist ein Register? ALU? MMU?

### Programmierkenntnisse in C:

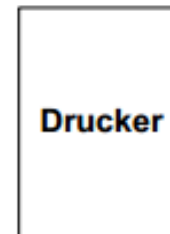
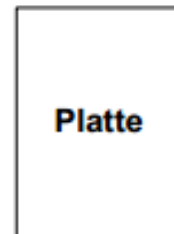
- Grundlegende Konzepte:
  - Datentypen, Kontrollstrukturen, Schleifen, Funktionen, Zeiger / Speicherverwaltung
  - Entwicklungsprozess (edit, compile, link, execute)
- Programmierung in der Unix (Linux) Umgebung
  - gcc, make, ...
  - Beispiel: Ubuntu 12.04
- Binärsystem
  - Bits und Bytes

## Einführung Hardware: Von-Neumann Rechner

Zentrale Recheneinheit (CPU)



**E / A – Geräte**



**BUS**



## Intel 8086 – Vier 16 Bit Register

### Allgemeine Arbeitsregister

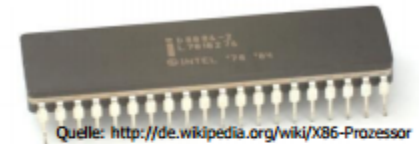
AX	AH	AL	Akkumulator
BX	BH	BL	Basisregister
CX	CH	CL	Zählerregister
DX	DH	DL	Datenregister

### Adress- und Indexregister

SP	Stapelzeiger
BP	Basiszeiger
DI	Zielindex
SI	Quellindex

### Segmentregister

CS	Codesegment
DS	Datensegment
ES	Extrasegment
SS	Stacksegment



Quelle: <http://de.wikipedia.org/wiki/X86-Prozessor>

IP	Befehlszeiger
SR	Statusregister (PSW)

## Intel Pentium – Acht 32-Bit Register

- Kompatibel zum 8086
- Acht Gleitkommaregister



Quelle: <http://www.chip.de>

	31	15	0	
EAX		AH	AL	AX
EBX		BH	BL	BX
ECX		CH	CL	CX
EDX		DH	DL	DX

Registerbezeichnungen:

[E]AX: Akkumulator

[E]BX: Basisregister

[E]CX: Zählregister

[E]DX: Datenregister

## AMD64 bzw. Intel 64 (Core-i Serie)

- Sechzehn 64 Bit Register
- Acht 64 Bit Gleitkommaregister
- Sechzehn 128 Bit Mediaregister



Quelle: <http://www.socket939.co.uk>

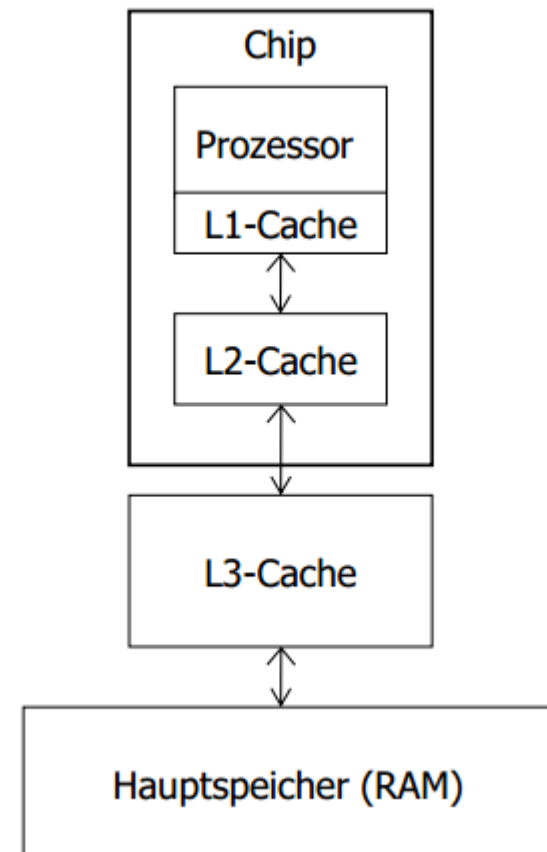
**Table 3-2. Addressable General Purpose Registers**

Register Type	Without REX	With REX
Byte Registers	AL, BL, CL, DL, AH, BH, CH, DH	AL, BL, CL, DL, DIL, SIL, BPL, SPL, R8L - R15L
Word Registers	AX, BX, CX, DX, DI, SI, BP, SP	AX, BX, CX, DX, DI, SI, BP, SP, R8W - R15W
Doubleword Registers	EAX, EBX, ECX, EDX, EDI, ESI, EBP, ESP	EAX, EBX, ECX, EDX, EDI, ESI, EBP, ESP, R8D - R15D
Quadword Registers	N.A.	RAX, RBX, RCX, RDX, RDI, RSI, RBP, RSP, R8 - R15

Quelle: Intel® 64 and IA-32 Architectures Software Developer's Manual

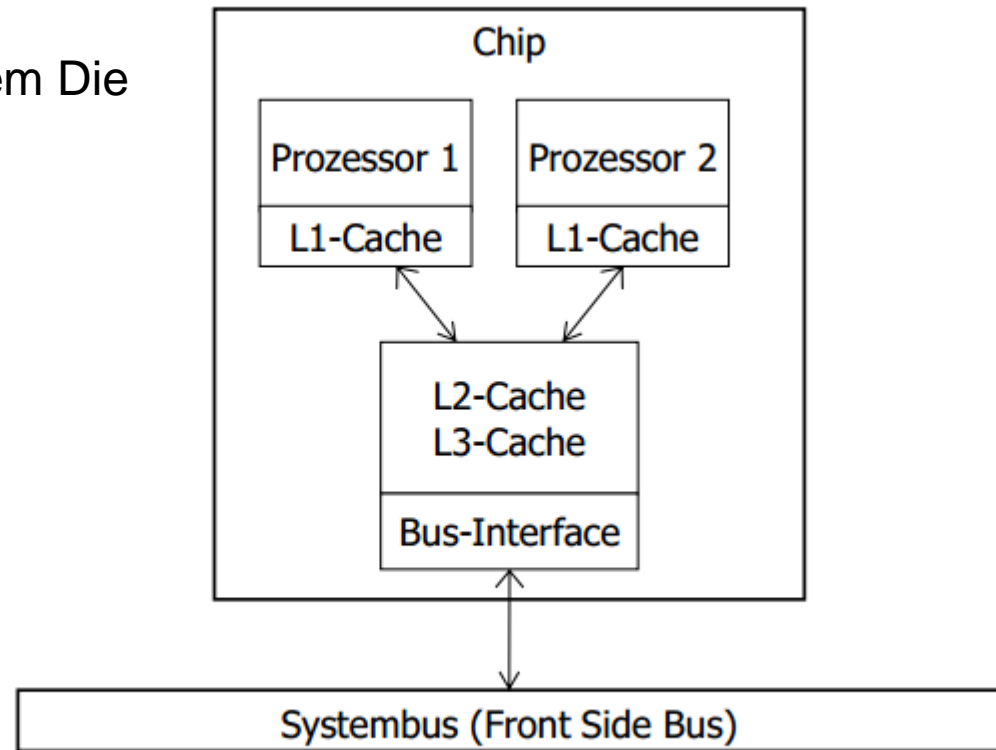
## Speicherhierarchie

- L1 ist kleiner und schneller als L2
- L2 ist kleiner und schneller als L3
- L3 ggf. außerhalb des Chip



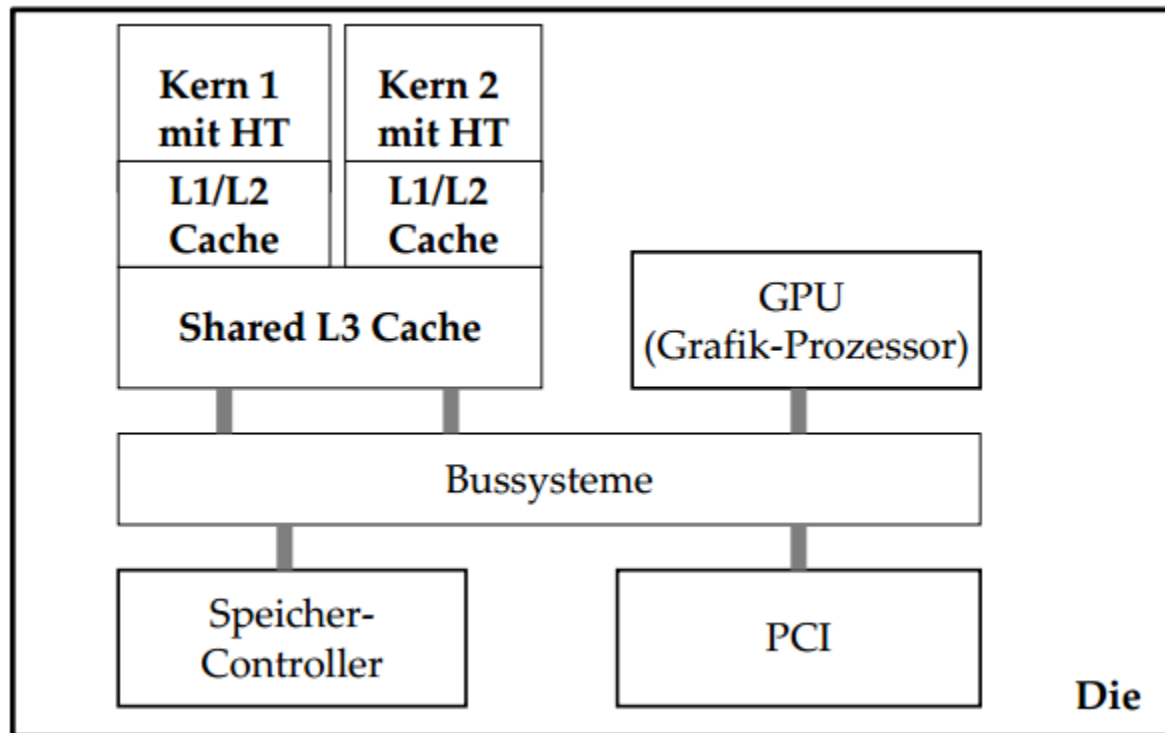
## Mehrkern Chips

- Zwei Prozessorkerne mit integriertem L1 Cache
- L2 und L3 Cache auf dem Die



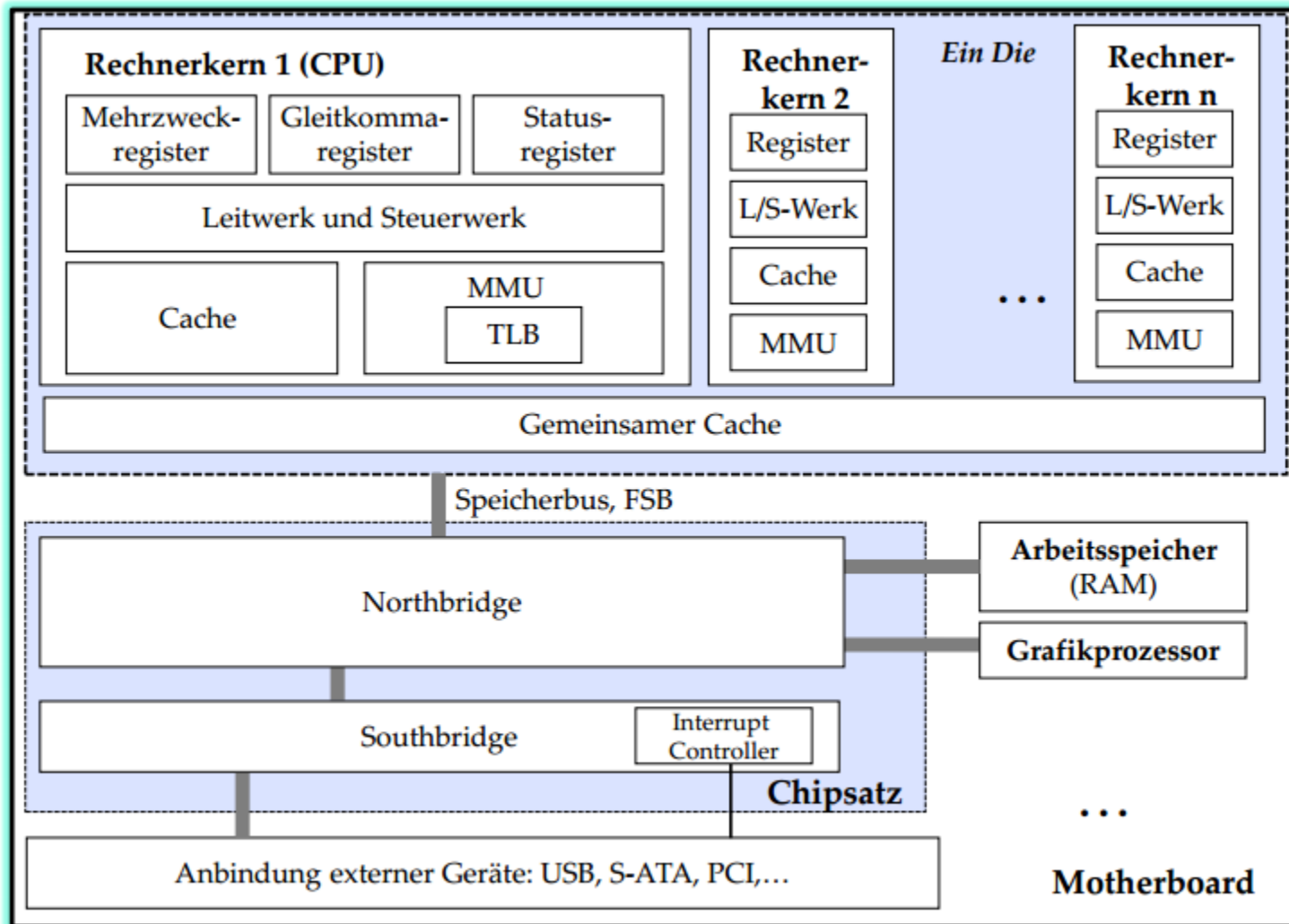
Quelle: *Böttcher, A.*: Rechneraufbau und Rechnerarchitektur, Springer-Verlag, 2006

## Intel Core i5

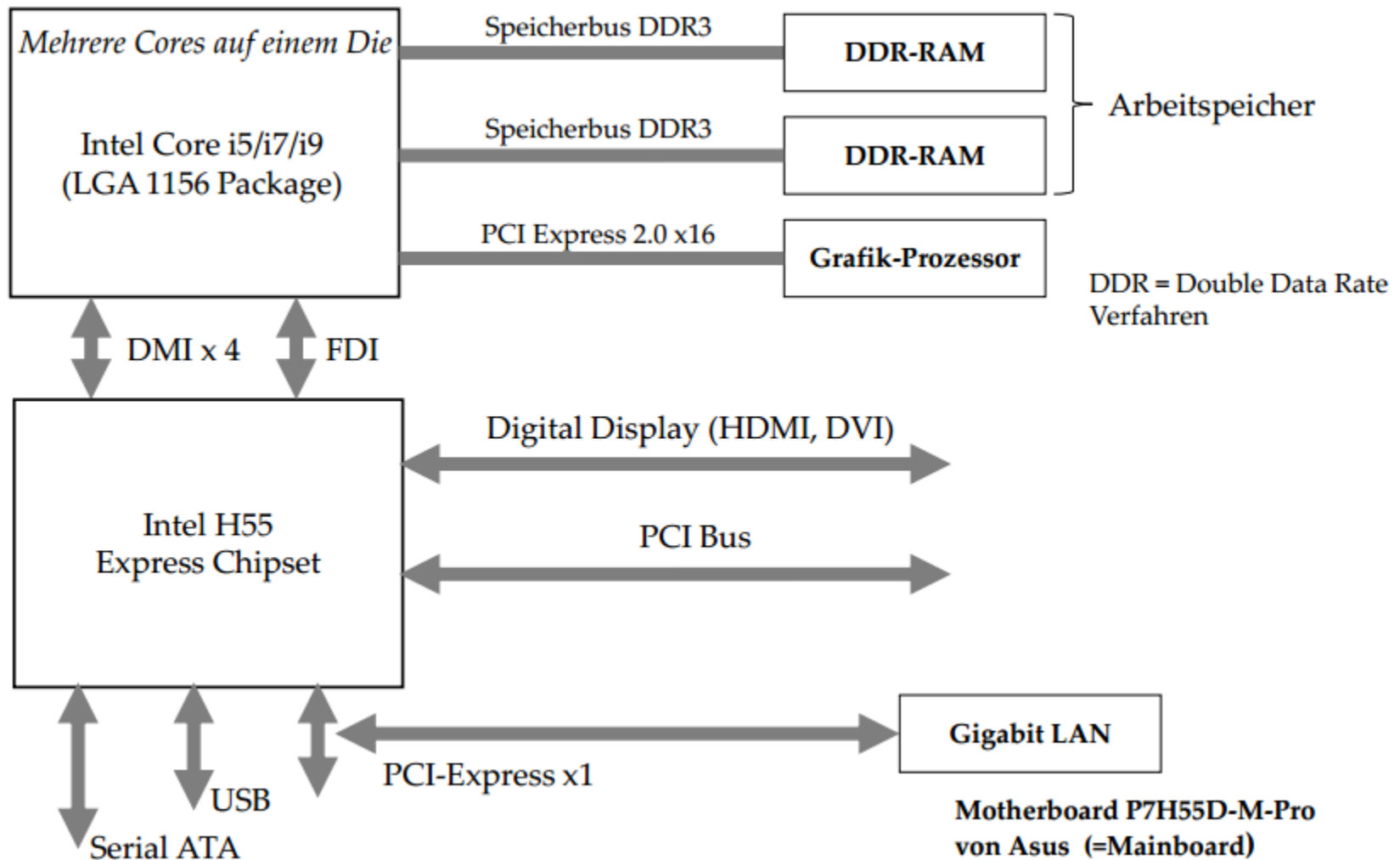




## Hardwaremodell (vereinfacht)



## Skizze: Motherboard Core-i-Serie von Intel



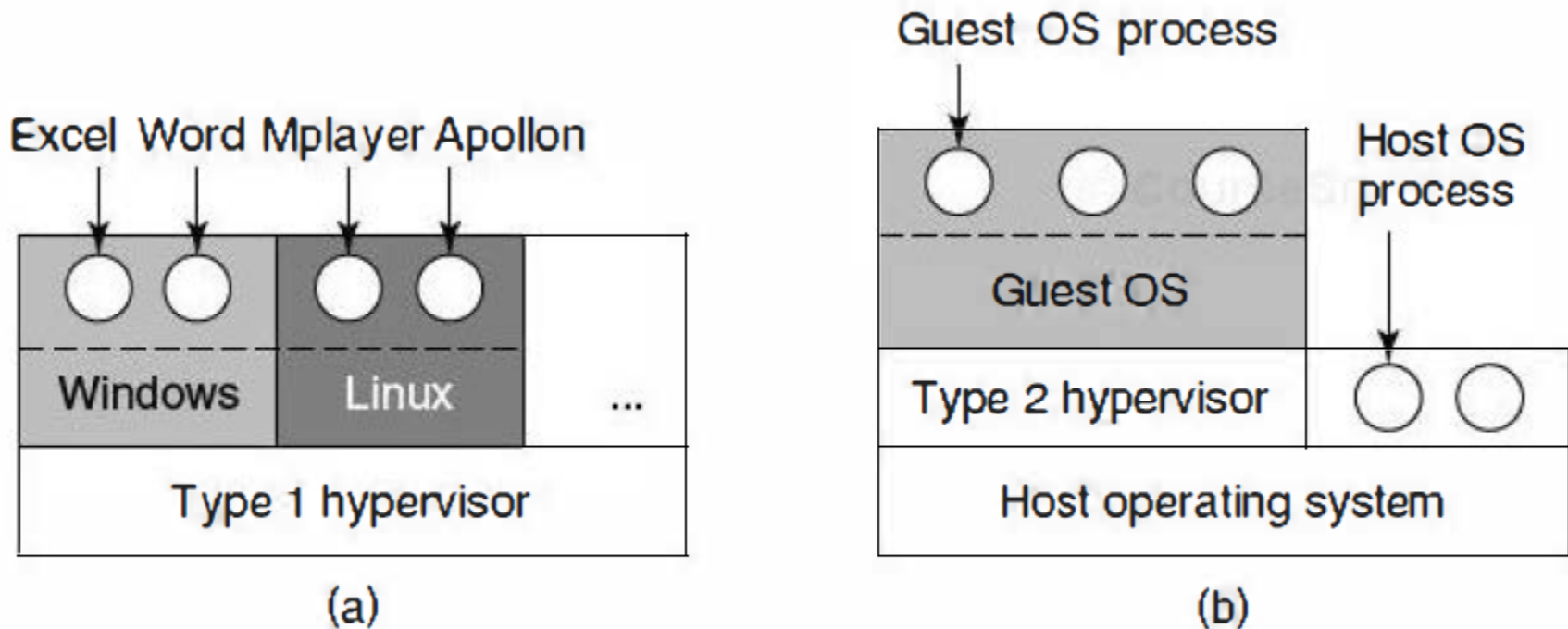


## Einführung: Virtualisierung

Techniken:

- **Betriebssystem**-Virtualisierung
- **Plattform**-Virtualisierung:
  - *Vollständige* Virtualisierung
    - Typ-I Hypervisor
    - Typ-II Hypervisor
  - *Para*-Virtualisierung

## Typ-I und Typ-II Hypervisor (Abb. aus Tanenbaum)



**Figure 1-29.** (a) A type I hypervisor. (b) A type 2 hypervisor.

**Vielen Dank für Ihre  
Aufmerksamkeit**