

Vorlesung

Betriebssysteme

Teil 1 Einführung

Dozent

Prof. Dr.-Ing.

Martin Hoffmann

martin.hoffmann@fh-bielefeld.de

Raum D206

Ringstr. 94

Materialien bereitgestellt durch

- Prof. Dr. Heinz-Josef Eikerling
 - Hochschule Osnabrück
- Prof. Dr. Christian Müller-Schloer
 - Leibniz Universität Hannover
- Prof. Dr. habil. Jürgen Brehm
 - Leibniz Universität Hannover

Konstruktive Kritik ist herzlich erbeten!

Organisatorisches

- 2h Vorlesung + 2h Praktikum
- Vorlesung:
 - Vorbereitung zum Praktikum
 - Übungsaufgaben zur Klausurvorbereitung
- Prüfung:
 - Klausur
- Planung Praktikum:
 - Anwesenheit = (Übungsblatt vorlegen + Lösungsversuche zum vorherigen Aufgabenblatt)
 - maximal drei Fehltermine
 - Erstes Praktikum in D109 am
 - **G2 Mo 20.04. 9:45 Uhr**
 - **G3 Mo 20.04. 11:30 Uhr**
 - **G1 Mi 22.04. 12:00 Uhr**

Hinweise

- Vorgehen:
 - Präsentation der Inhalte über Folien
 - Folien, Aufgaben, weitere Unterlagen werden per Ilias bereit gestellt (PW gates)
 - **Testfragen** während der Vorlesung
- Praktikum:
 - Aufgaben werden in Eigenarbeit in Gruppen gelöst
 - Probleme sofort melden

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

Literatur

Bücher:

- **Andrew S. Tanenbaum: Moderne Betriebssysteme. Pearson Studium, 4., aktualisierte Auflage.**
- Mandl, Peter: Grundkurs Betriebssysteme - Architekturen, Betriebsmittelverwaltung, Synchronisation, Prozesskommunikation, 3. Aufl. 2013
- William Stallings: Operating Systems - Internals and Design Principles, 6th Ed., Pearson, 2007
- M. Garrelts: Bash Guide for Beginners. GNU Public Licence
<http://www.tldp.org/LDP/Bash-Beginners-Guide/html/Bash-Beginners-Guide.html>
- S. Grägert: POSIX-Programmierung mit UNIX. GNU Public Licence (in Ilias und graegert.de erhältlich)

Aufbau der Veranstaltung

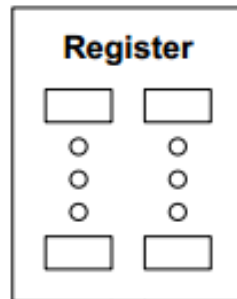
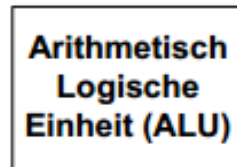
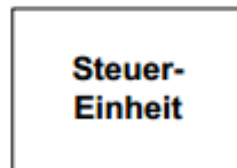
- Einführung
 - Aufgabe eines Betriebssystems
 - Aufbau von Betriebssystemen
 - Programmierung in C
- Prozesse
 - Prozesse und Threads
 - Prozessstatus
 - Scheduling
- Prozesskommunikation
 - Kommunikation
 - Synchronisierung
- Verklemmungen (Deadlocks)
 - Modellierung von Verklemmungen
 - Behebung und Vermeidung von Verklemmungen

Aufbau der Veranstaltung (2)

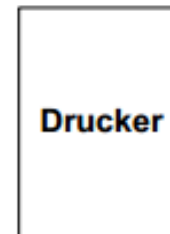
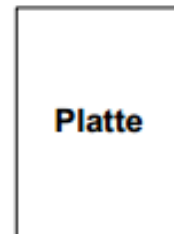
- Shell
 - Bedienung der Shell
 - Shell-Scripting
- Speicherverwaltung
 - Partitionierung
 - Virtueller Speicher
- Dateiverwaltung
 - Dateinamen
 - Dateisysteme
- Geräteverwaltung
 - Ein/Ausgabe
 - Treiber
- Sicherheit

Einführung Hardware: Von-Neumann Rechner

Zentrale Recheneinheit (CPU)



E / A – Geräte



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>

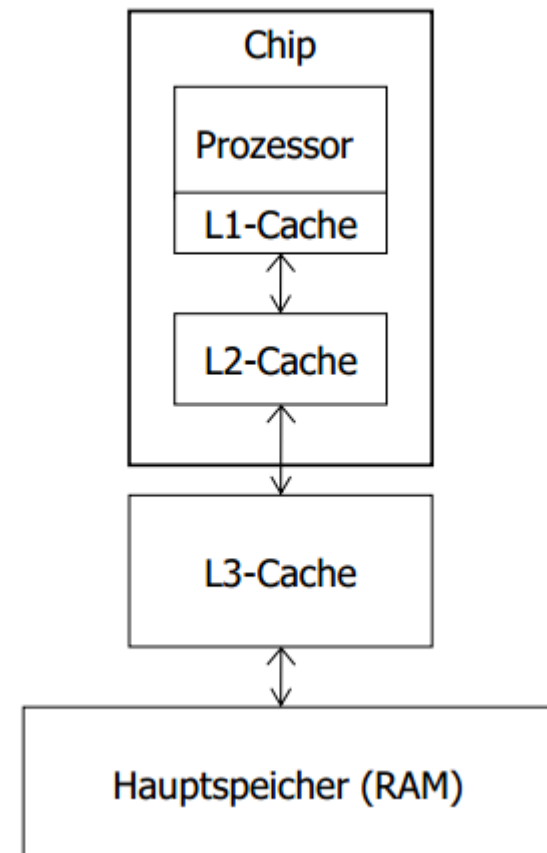
Table 3-2. Addressable General Purpose Registers

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

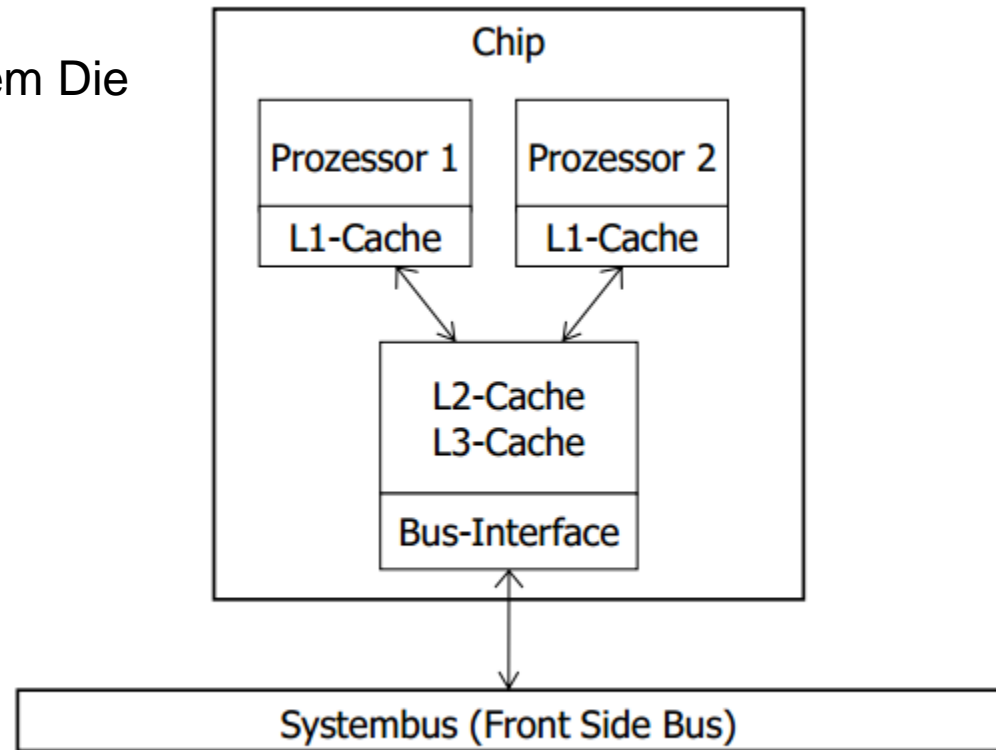
Speicherhierarchie

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



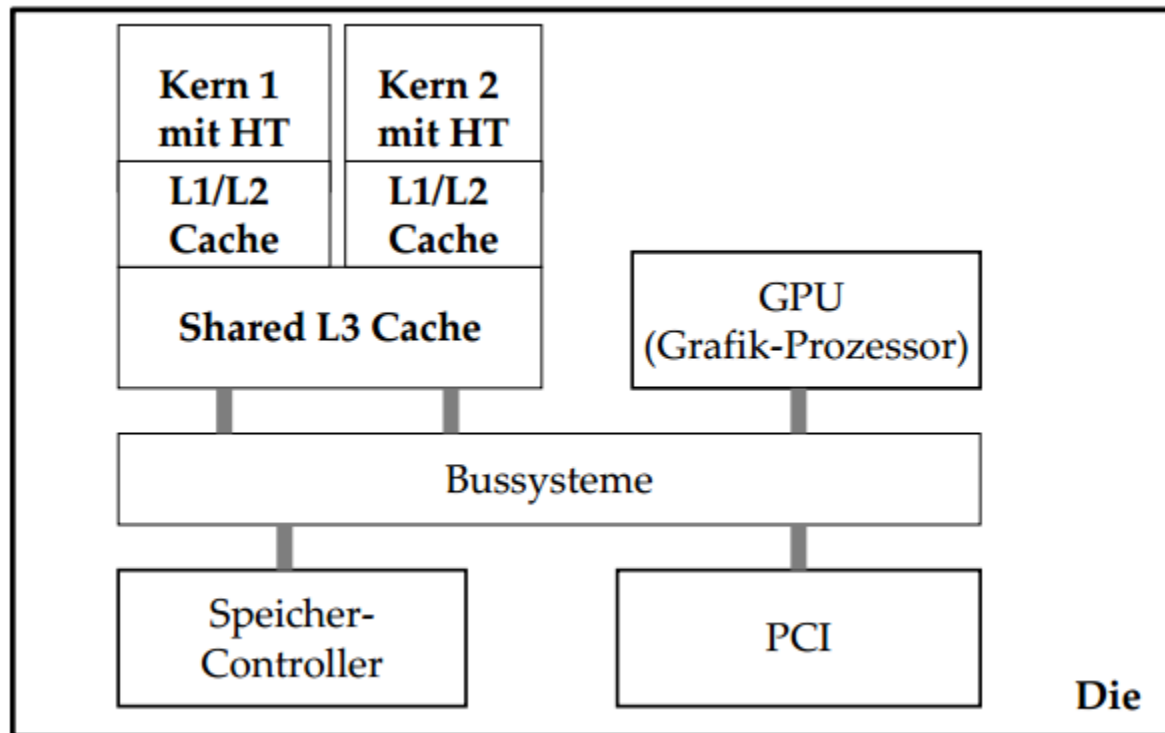
Mehrkern Chips

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



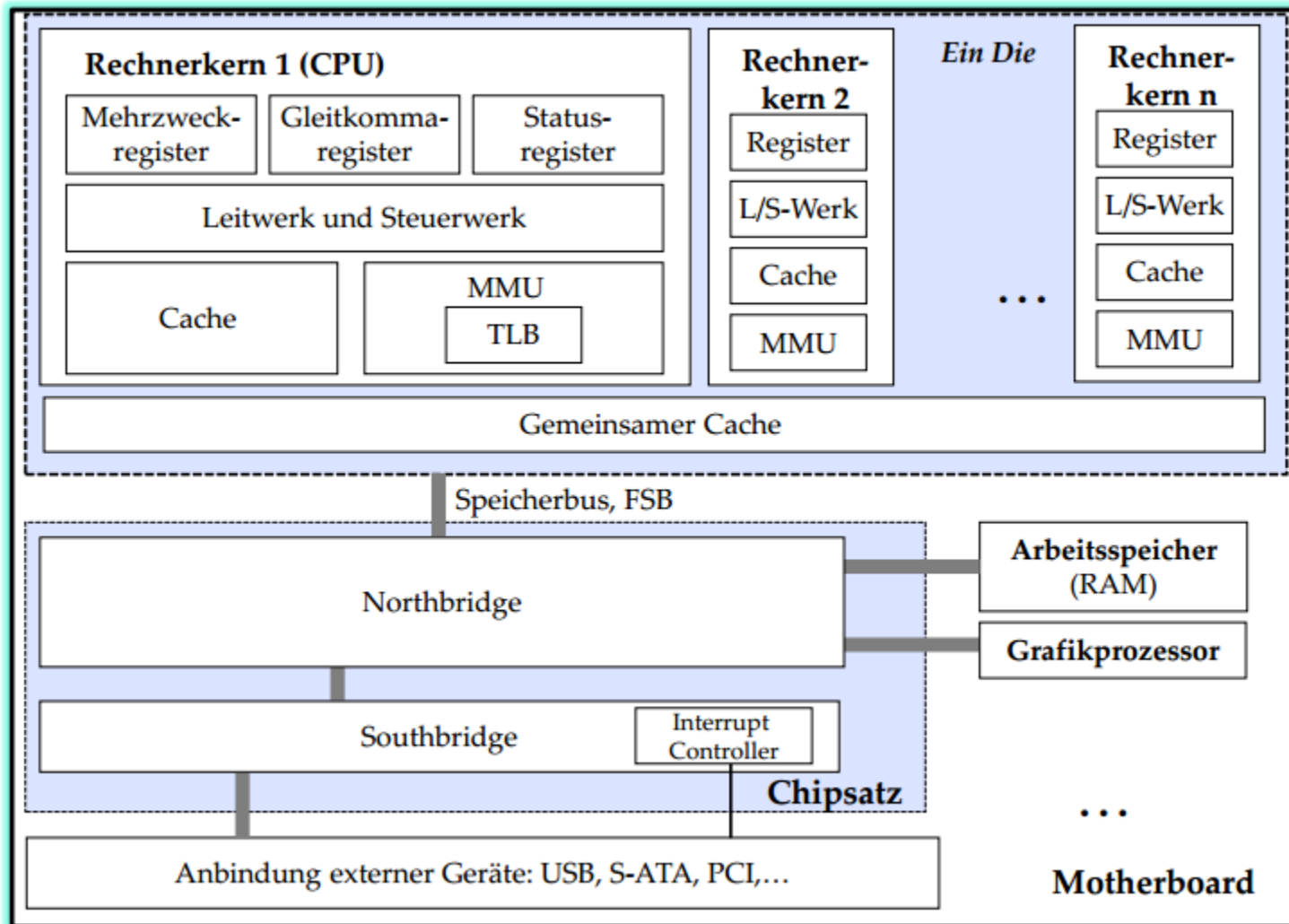
Quelle: *Böttcher, A.*: Rechneraufbau und Rechnerarchitektur, Springer-Verlag, 2006

Intel Core i5

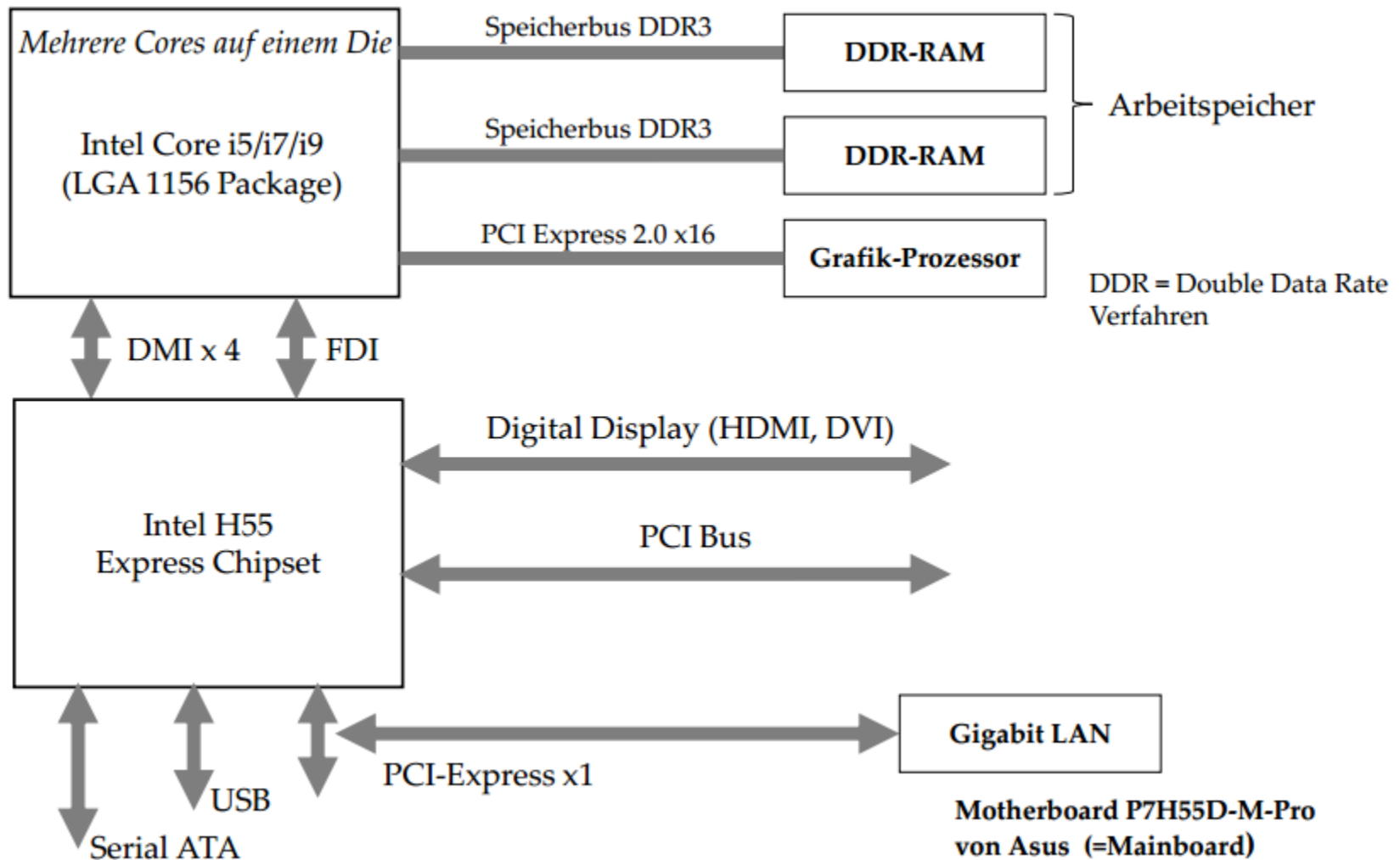




Hardwaremodell (vereinfacht)



Skizze: Motherboard Core-i-Serie von Intel



Einführung: Definition

Definitionen Betriebssystem:

- DIN 44300:

„Die Programme eines digitalen Rechensystems, die zusammen mit den Eigenschaften dieser Rechanlage die Basis der möglichen Betriebsarten des digitalen Rechensystems bilden und die insbesondere die Abwicklung von Programmen steuern und überwachen.“

- A. Silberschatz:

„An operating system is similar to a government... Like a government the operating system performs no useful function by itself.“

Testfrage: Aufgaben eines Betriebssystems

- Nennen Sie einige BS-Aufgaben!

Aufgaben des Betriebssystems (1)

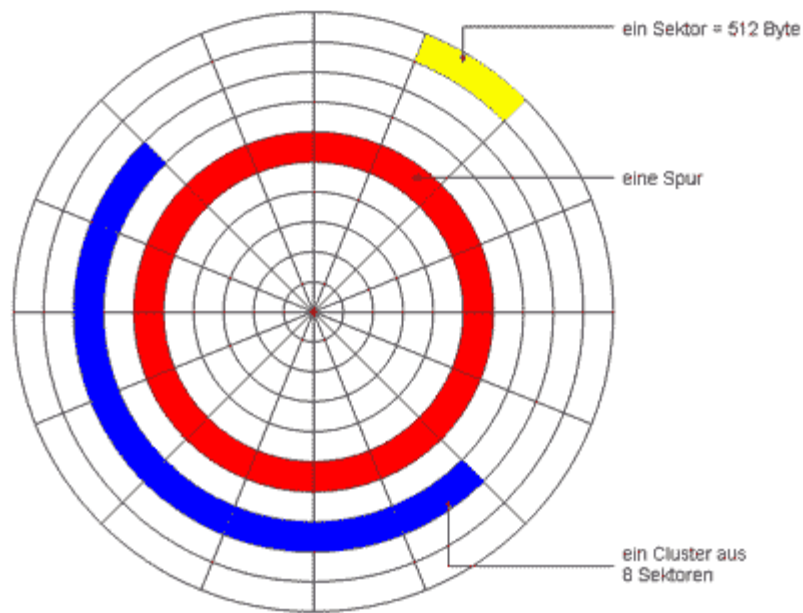
- Boot
- BS prüft **Zugriffsberechtigung** (login).
- BS stellt Verbindung zum **Benutzer** her: Kommandointerpretation, Shell, Benutzeroberfläche (Benutzer: Mensch, Programm).
- BS verwaltet **Daten** in Form von Dateien (Files): Zugriff auf Datenfiles auf Massenspeicher (Schreiben, Lesen), Kopieren, Löschen, Benennen, Ordnen. Falls Datei ein ausführbares Programm enthält: BS startet **Ausführung des Programms**.

Aufgaben des Betriebssystems (2)

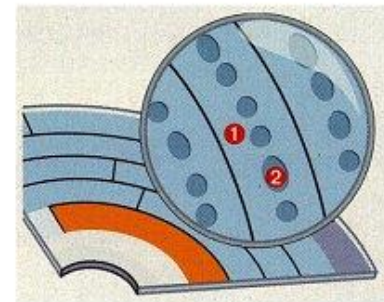
- BS verwaltet die **Ressourcen**: CPU, Speicherplatz, FD, HD, Netzausgang, Drucker, aber auch SW-Ressourcen: Programme, Prozesse, Tabellen.
- Beispiel Mainframe, 500 Terminals: BS teilt jedem Benutzer (Terminal) die CPU für eine begrenzte Zeit zu. "Gleichzeitige" Bearbeitung wird vorgetäuscht (**Time Sharing**)
- BS sammelt **Abrechnungsdaten** (bei Mehrbenutzersystemen)
- BS verbirgt Komplexität der HW-Maschine durch Vortäuschung einer komfortableren Maschine (**Abstraktion, Virtualisierung**).

Aufgaben des Betriebssystems (3)

- BS vereinfacht den **Gerätezugriff**.
- Beispiel: Massenspeicher



DVD-Recording



Die spiralförmige Spur trägt die Daten in einer Folge von reflektierenden und nicht reflektierenden Stellen (Land 1 / Pit 2).

Sie entsprechen den digitalen Werten 0 und 1.

DVD-Video (Standard)

Einführung: Aufgaben

Aufgaben eines Betriebssystems:

- **Verwaltung** der Betriebsmittel (Ressourcen)
 - *Prozessverwaltung* (erzeugen, löschen, zuteilen, synchronisieren)
 - *Speicherverwaltung*
 - Verwaltung des *Dateisystems*
 - Verwaltung von *Geräten*
 - Verwaltung der *Benutzer*
- **Abstraktion** von der Hardware
 - Die Eigenschaften der Hardware werden vor dem Benutzer verborgen.
 - Die Benutzung der Hardware wird durch eine einheitliche Schnittstelle gewährleistet.
 - Die Hardware stellt zusammen mit dem Betriebssystem eine *abstrakte Maschine* dar, auf der die Benutzerprogramme aufsetzen.

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



Vorlesung

**Vielen Dank für Ihre
Aufmerksamkeit**

Dozent

**Prof. Dr.-Ing.
Martin Hoffmann**
martin.hoffmann@fh-bielefeld.de

**Raum D206
Ringstr. 94**