

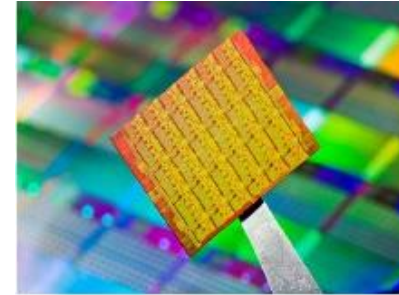
Betriebssysteme

Kapitel 1 Einführung

Aufbau der Vorlesung

- **Einführung**
 - Historischer Überblick
 - Betriebssystemkonzepte
- Prozesse und Threads
 - Einführung in das Konzept der Prozesse
 - Prozesskommunikation
 - Scheduling von Prozessen
 - Threads
- Speicherverwaltung
 - Einfache Speicherverwaltung
 - Virtueller Speicher
 - Segmentierter Speicher
- Dateien und Dateisysteme
 - Dateien
 - Verzeichnisse
 - Implementierung von Dateisystemen
- Input/Output
 - Grundlegende Eigenschaften der I/O-Hardware
 - Festplatten
 - Terminals
 - I/O-Software
- Deadlocks/Verklemmungen
- Virtualisierung und Cloud
- IT-Sicherheit
- Multiprozessor-Betriebssysteme

- Immer mehr Mehrkernprozessoren
 - Dezember 2009:
 - Intel 48 Core Prozessor
 - 24 „Kacheln“ mit 2 Cores
 - 24-Router Mesh Netzwerk
 - 4 DDR3 memory controller
 - Hardware Support für Nachrichtenaustausch (Message-Passing)
 - „Single-Chip-Cloud-Computer“



DDR4



DDR3



- Parallelität auch auf dem Chip
 - Wie programmiert man 48, 64, 512 Kerne?
 - 1 Kern für Word, 1 Kern für Browser, 2 Kerne für Audio/Video, ...
 - 44 für Antivirus??

Aufgabe/Frage

Fragen

- Was ist der Unterschied zwischen Core und CPU? Wo spielt diese Unterscheidung eine ganz wichtige Rolle?
- Was bedeutet DDR3 und DDR4 und erklären Sie den Unterschied?

Bedingungen:

→ Zeit: ad hoc



Ad hoc

iPhone 8:

- Apple A11 Bionic Chip
- CPU-Kerne 2
- Maximaler CPU-Takt 2.39 GHz
- Grafik-Chip Apple A11 Bionic GPU

iPhone 13:

- Apple A15 Bionic Chip
- 6-Core CPU mit 2 Performance-Kernen und 4 Effizienz-Kernen
- Neue 5-Core GPU
- Neue 16-Core Neural Engine
- Maximaler CPU-Takt 3 GHz

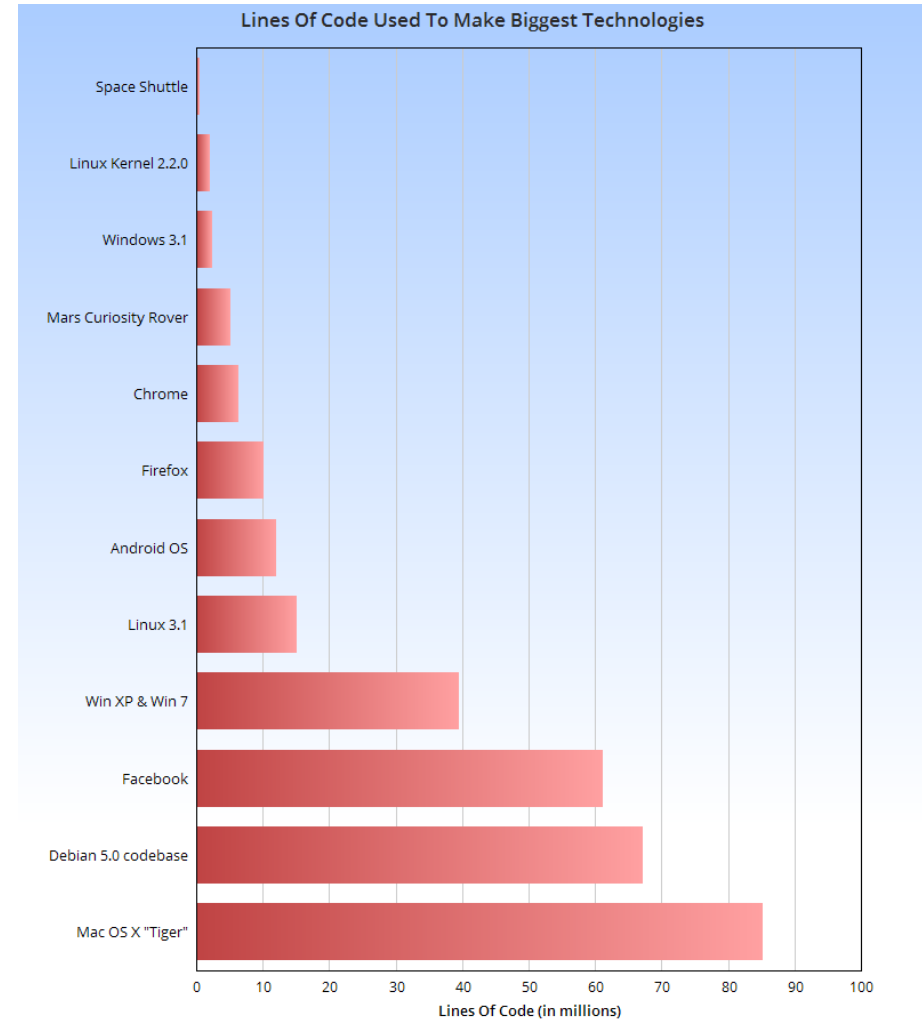
Samsung S21:

1 x Cortex-X1 bis 2,9 GHz
3 x Cortex-A78 bis 2,8 GHz
4 x Cortex-A55 bis 2,2 GHz
5nm ARM-Design

- Our World
in Data

6

- Dies spiegelt sich auch in Software wieder



<https://www.informationisbeautiful.net/visualizations/million-lines-of-code/>

SOURCE: NASA, QUORA, WIKIPEDIA
INFORMATION IS BEAUTIFUL



- Herausforderungen
 - Wie organisiert man das Management von Komplexität in heterogenen IT-Umgebungen?
 - Wie können die Anwendungen/Computersysteme ihre Aufgabe in Zukunft erfüllen?
 - Wie unterstützen Computer den Menschen/das Business?
 - Welche neuen Herausforderungen/Anwendungen kommen in der Zukunft auf uns zu?

- Rechensysteme sollen Probleme lösen!!
 - Wettervorhersagen
 - Steuerung eines Kraftwerks
 - Berechnungen von Ingenieuraufgaben
 - Informationen aus dem Internet besorgen
 - Email/Informationen austauschen
 - Textverarbeitung
 - Gehaltsabrechnung
 - Cognitive
 - Machine Learning
 - usw.
- Rechensysteme sind kein Selbstzweck
 - Business: Unterstützung einer Wertschöpfungskette
 - Privat: Unterhaltung

- Was wissen Sie über Betriebssysteme?
- Was machen Betriebssysteme?

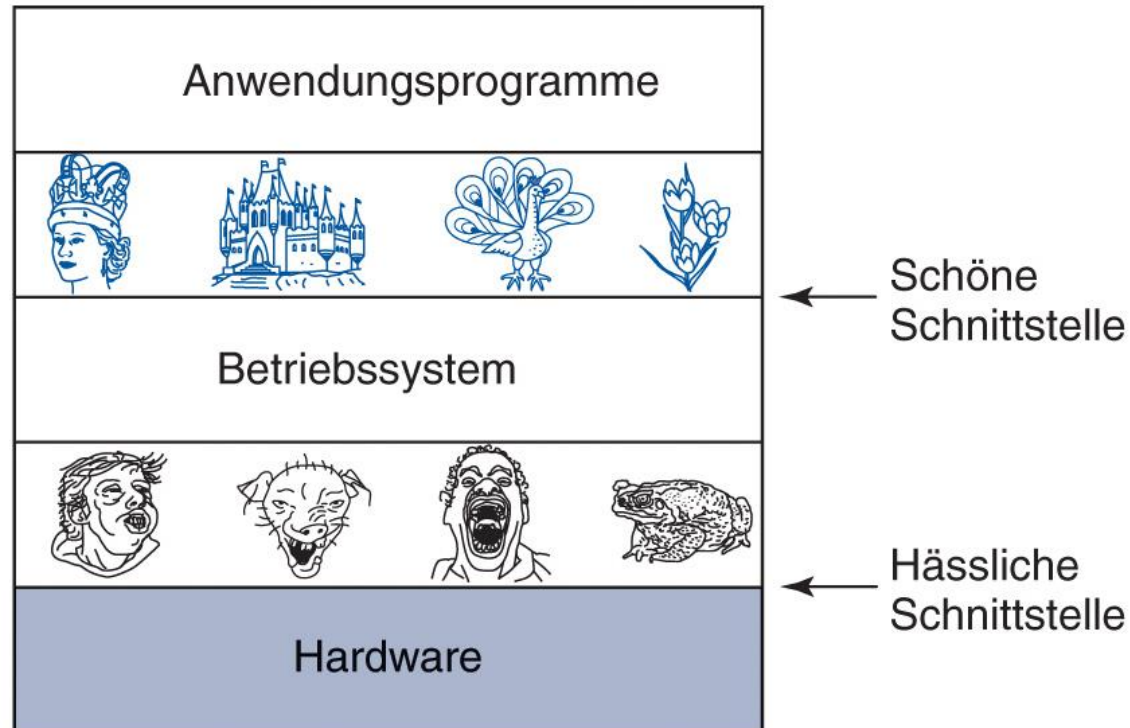
Frage/Diskussion

Fragen:

- Was wissen Sie über Betriebssysteme?
- Was machen Betriebssysteme?
- Schuhe sind vielfältig.
Können Sie Ähnlichkeiten zwischen der Welt von Schuhen und Betriebssystemen finden?



Ad hoc

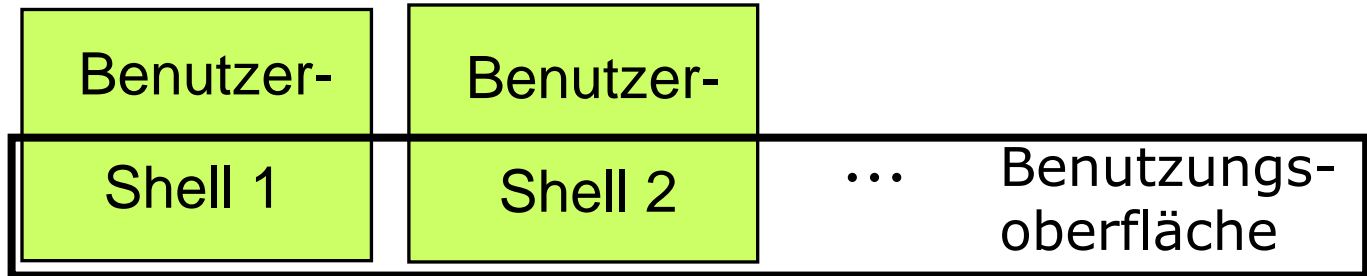


- Betriebssystem soll dem Anwendungsprogrammierer ein „**einfaches und klares Modell**“ eines Computers zur Verfügung stellen

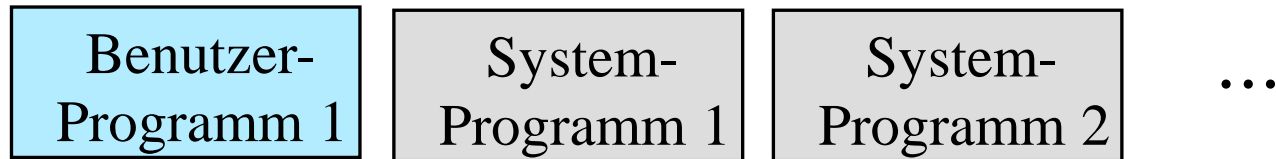


Erkennungsmerkmal: **Betriebssystem läuft im Kernmodus (Kernel Mode)**

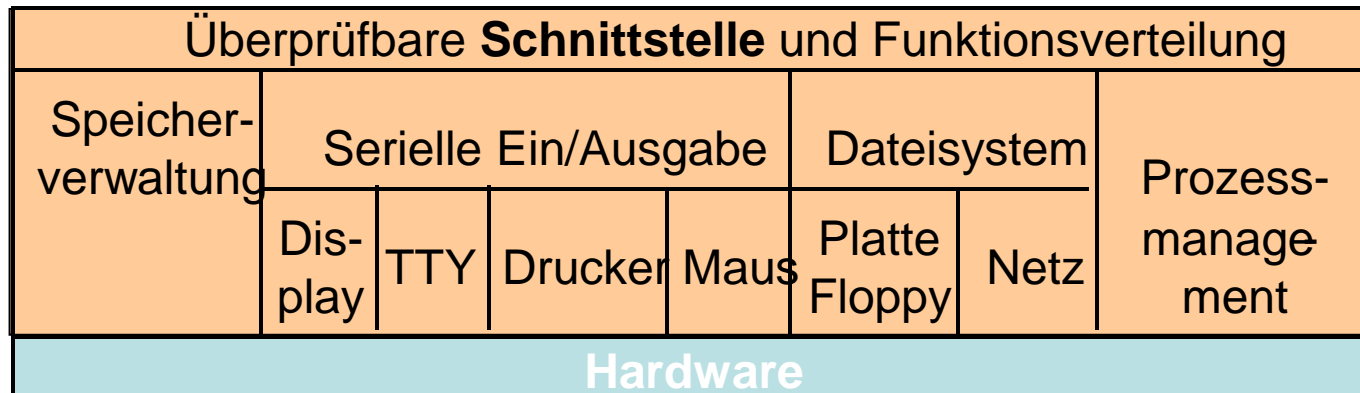
*Multi-
User*



user mode



kernel mode



Aufgabe/Frage

Fragen:

- Was bedeutet TTY?

Bedingungen:

→ Zeit: ad hoc



Ad hoc

Aufgaben des Betriebssystems

1. Abstraktion der Hardware

- Hardware beschränkt sich auf notwendige Funktionen (günstig)
 - **Betriebssystem stellt Funktionen bereit, die Anwendungsprogramme nutzen können**
 - Bsp.: Festplatte
- Trotz ähnlicher Architektur unterscheiden sich die Rechner im Detail z.B. in Speicher, Controller, ...
 - **Betriebssystem realisiert eine einheitliche Sicht für Anwendungen**
 - Bsp.: Dateien auf externen Speichermedien (kein Unterschied zwischen Digitalkamera und CD)

Aufgaben des Betriebssystems

2. Verwaltung der Ressourcen

- Anwendung braucht Ressourcen um ausgeführt zu werden, z.B. CPU, Speicher, Platte, Netzwerk, ...
- Leistungsfähige Rechner laufen im **Mehrprozess-** und **Mehrbenutzerbetrieb**, d.h. mehrere Anwendungen laufen „gleichzeitig“
 - **Betriebssystem verteilt die Ressourcen gerecht und sichert die Anwendungen und Benutzer gegeneinander ab**
- Multiplexing
 - des Prozessors
z.B. zeitlich bei CPU, Platte,... nach dem Prinzip **“Einer nach dem anderen“**
 - z.B. quantitativ bei Arbeitsspeicher nach dem Prinzip **„ein Teil für dich, ein Teil für mich“**

Geschichte der Betriebssysteme

1. Generation (-1945)

- Technologie: Elektronenröhren
- Manuelle Programmierung
 - Teilweise durch feste Verdrahtung
 - Einfache numerische Berechnungen waren möglich

2. Generation (1955 – 1965)

- Technologie: Transistoren (Großrechner)
- Trennung von Entwicklern, Operatoren, Wartungspersonal
- Lochkarten mit Programmcode (z.B. Assembler, Fortran)
- Betriebssystem
 - startet Übersetzer und Programm
 - nimmt Eingabe entgegen
 - gibt Ausgabe auf Drucker aus

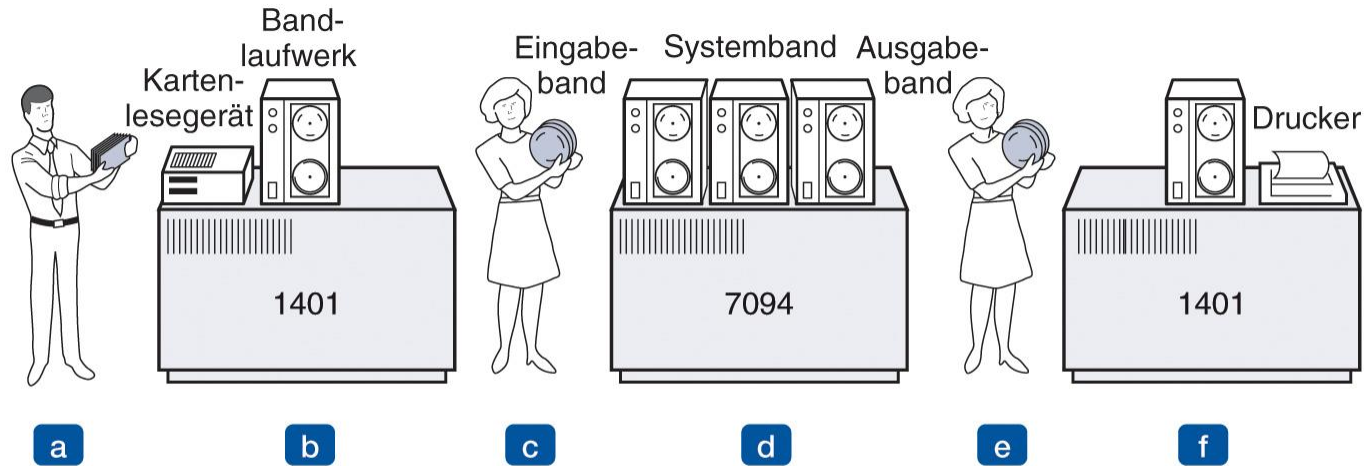
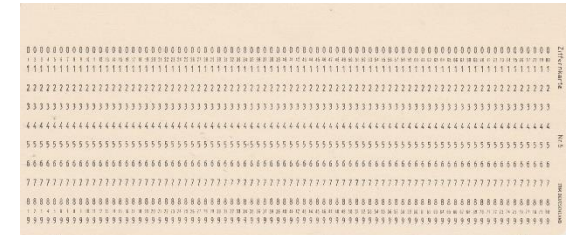
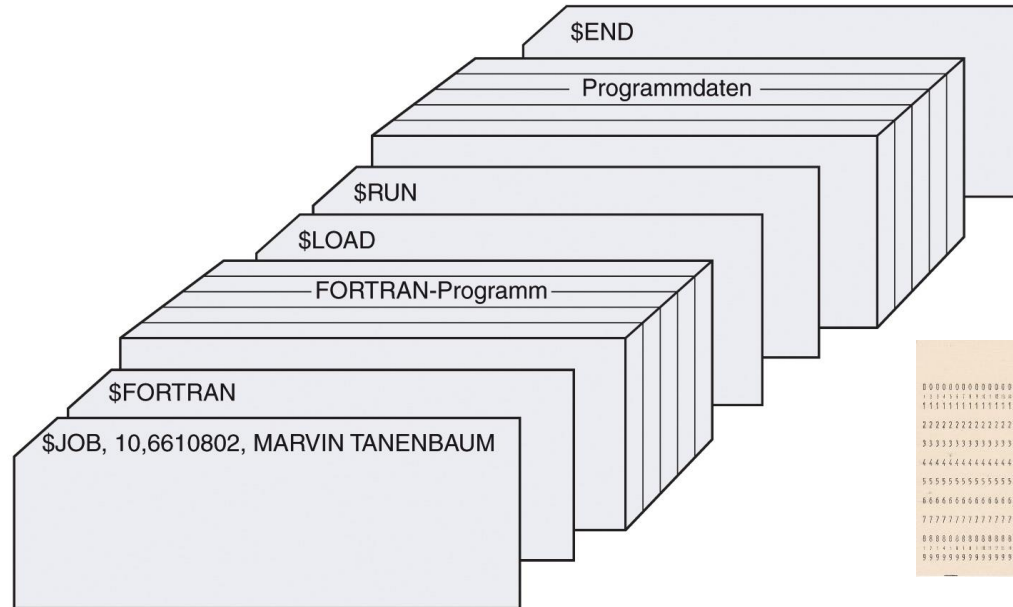


Abbildung 1.3: Ein frühes Stapelverarbeitungssystem. (a) Die Programmierer bringen die Stapel zur 1401. (b) Die 1401 liest den Stapel von Jobs auf ein Band. (c) Ein Operator trägt das Eingabeband zur 7094. (d) Die 7094 führt die Berechnung durch. (e) Ein Operator trägt das Ausgabeband zur 1401. (f) Die 1401 druckt die Ausgabe.

– Typische Systeme: FORTRAN Monitor System (FMS)

- Einführung von Jobs



- auch mehrere Jobs nacheinander
 - vom Magnetband
- Stapelverarbeitung
 - Arbeitsprinzip einer automatischen, vollständigen und meist sequentiellen Abarbeitung von Kommandos oder Daten

3. Generation (1965-1980)

- Technologie: Integrierte Schaltungen
- Einführung von Rechnerfamilien
 - Gleicher Befehlssatz
 - Unterschiedliche Leistung
 - Portabilität von Programmen möglich

Bsp.: IBM System/360 mit Produkten 370, 4300, 3080, 3090, Heute zSeries

→ Betriebssystem abstrahiert die Unterschiede der Rechner/Geräte!

- Einführung des Mehrprogrammbetriebs
 - CPU wartet oft (80%-90% der Zeit) auf Eingabe/Ausgabe-Geräte
 - statt zu warten wird ein anderer Job aktiviert
- Betriebssystem verwaltet die Geräte und Ressourcen!



3. Generation (1965-1980) ff

- Interaktive Nutzung der Rechner durch Timesharing
 - Terminals statt Lochkarten und Drucker
 - Mehrere Benutzer gleichzeitig, damit wurden auch Sicherheitsmechanismen notwendig

Bsp.: MULTICS (Multiplexed Information and Computing System)

 - Viele Innovationen, aber nur geringer wirtschaftlicher Erfolg
- Vision:
 - Zentralisierte Rechner verwendbar wie das Stromnetz
 - Ähnlichkeiten mit Internet und Cloud-Computing
- Verbreitung von Minicomputer
 - Bsp.: DEC PDP-1 bis PDP-11
 - MULTICS wurde angepasst → **Ursprung von UNIX**

4. Generation (1980-heute)

- Technologie: Hochintegrierte Schaltkreise (Mikroprozessoren)
 - Billige Hardware
- Zurück zu Einbenutzersystemen (DOS, Windows, ...)
- Von der Kommandozeile zur Graphischen Benutzeroberflächen (GUI)
 - Apple Macintosh
- Zunehmende Vernetzung der Rechner
 - Client/Server-Systeme: mehrere Benutzer
 - UNIX, Linux, Windows NT, ...
 - Virtualisierung
- Virtualisierung und verteilte Betriebssysteme
 - Ganz aktuell: Wie ist die Cloud aufgebaut?

Aufgabe/Frage

Fragen:

- Definieren Sie was Cloud Computing ist?
- Welche Cloud-Komponenten kennen Sie? Wie ist eine Cloud aufgebaut?
- Welche Servicemodelle gibt es?

Bedingungen:

→ Zeit: ad hoc



Ad hoc

Geschichte der Betriebssysteme

Juni 2021

Vorhang auf: Fraunhofer und IBM weihen Quantencomputer ein

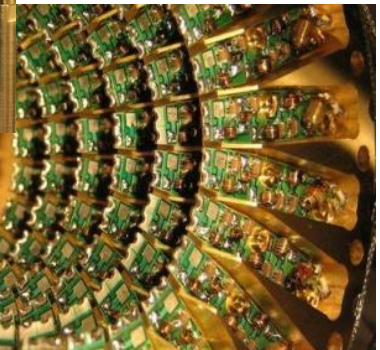
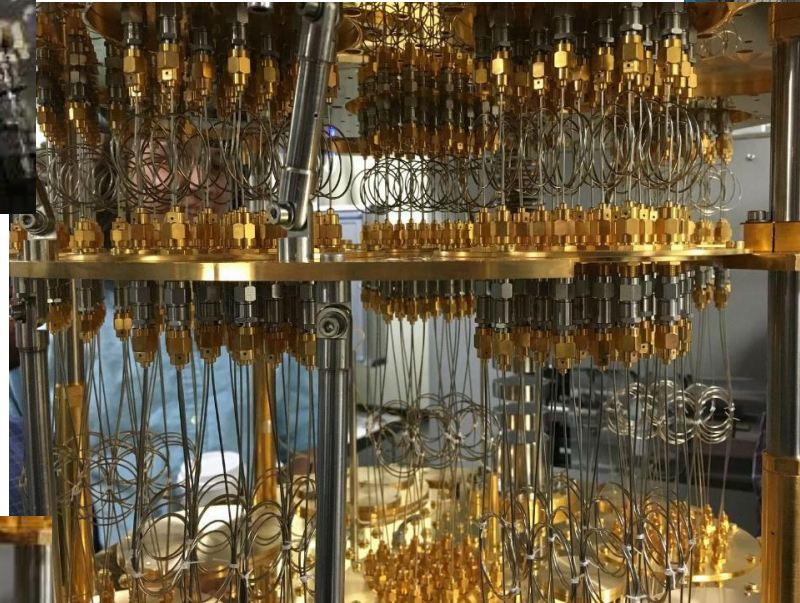
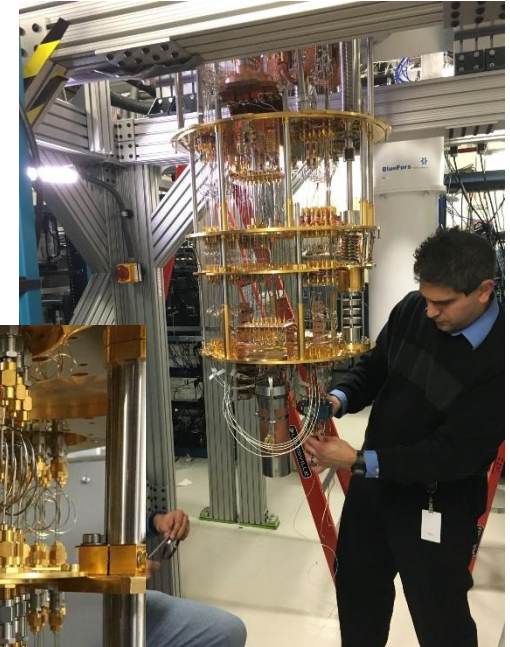
Presseinformation / 15. Juni 2021

Im digitalen Beisein von Kanzlerin Dr. Angela Merkel und Bundesforschungsministerin Anja Karliczek sowie dem baden-württembergischen Ministerpräsidenten Winfried Kretschmann enthüllen Experten von Fraunhofer und IBM am 15. Juni den ersten IBM Quantum System One in Deutschland für die Öffentlichkeit. Mit 27 Qubits ist es das derzeit leistungsstärkste System in Europa.



Mögliche Anwendungsgebiete für die Technologie sieht IBM in der Medikamenten- und Materialforschung, die Optimierung von Lieferketten und Logistikabläufen, neuartige Analysen von Finanzinformationen, Teilaspekte der künstlichen Intelligenz wie Machine Learning sowie die Verbesserung der Sicherheit von Daten in Cloud-Umgebungen.

5. Generation (Zukunft) Q System



<https://www.youtube.com/watch?v=o-FyH2A7Ed0>

5. Generation (Zukunft) Q System One

<https://www.youtube.com/watch?v=o-FyH2A7Ed0>

<https://www.youtube.com/watch?v=VN8DGUDRKqY>

Geschichte der Betriebssysteme

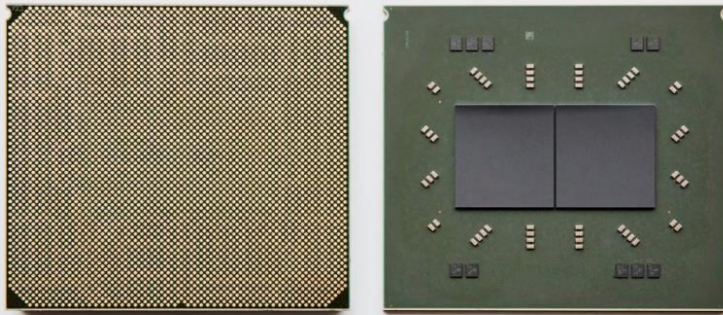
Announcement August 2021

BIG IRON —

A brief overview of IBM's new 7 nm Telum mainframe CPU

A typical Telum-powered mainframe offers 256 cores at a base clock of 5+GHz.

JIM SALTER - 9/3/2021, 12:45 AM



Enlarge / Each Telum package consists of two 7nm, eight-core / sixteen-thread processors running at a base clock speed above 5GHz. A typical system will have sixteen of these chips in total, arranged in four-socket "drawers."

- Chip mit 22,5 Milliarden Transistoren und 30,6 Kilometer Leitungen auf 17 Metallschichten.
- Optimierung Core-Performance zu durch 256 MByte Cache auf einem Chip
- d.h. 2 GByte Cache in einem mit vier Sockets bestückten Drawer mit vier Dual-Chip-Modulen.
- Jedem Core können 32 MByte „privater“ Level2-Cache zugewiesen werden, der mit einer Load/Use-Latenz von 19 Zyklen (~3,8 Nanosekunden, inklusive Zugriff auf den „Translation Lookaside Buffer“) aufwarten kann.

Betriebssystemfamilie

- **Großrechner (Mainframe)**
 - Hohe Ein-/Ausgabe-Leistung, viele Prozesse, Transaktionen,
- **Server**
 - Viele Benutzer über ein Netzwerk
- **Multiprozessorsysteme**
 - Parallelrechner
- **Personal Computer**
 - Linux, FreeBSD, Windows Vista, Windows 10
 - Oberfläche ist nicht das Betriebssystem
- **Handheld-Computer**
 - PDA, iPad, Android-Phones, iPhones ...

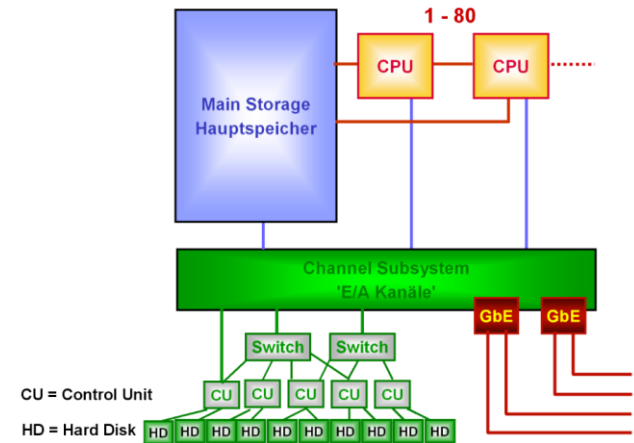
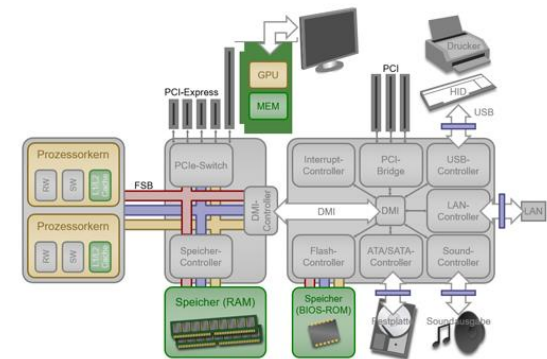


Abb. 1.3.9
Hardware Struktur eines Mainframe Systems



Betriebssystemfamilie

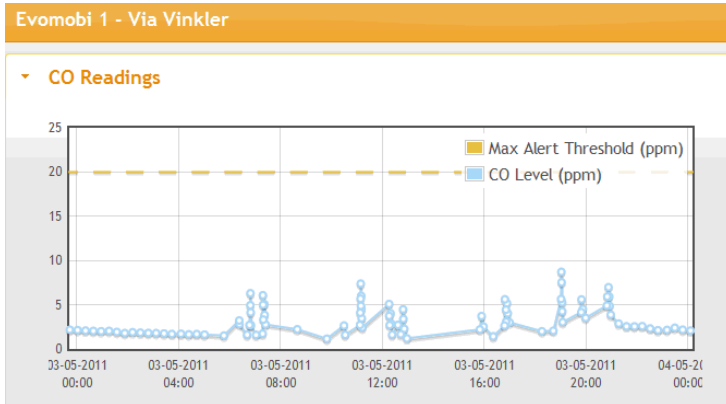
- **Eingebettete Systeme**
 - Auto, Fernseher, MP3-Player, ...
 - Nur vertrauenswürdige Software ausgeführt
 - Nachladen von Software durch Benutzer nicht möglich
- **Sensorknoten**
 - Kleine batteriebetriebene Computer mit Funkgeräten
 - Überwachungsaufgaben
- **Echtzeitbetriebssysteme**
 - Zeit ist essentiell bei Ressourcenvergabe
 - Steuerungsanlagen
 - Digitale Telefone, Audio- und Multimediasysteme
- **Smart Cards / Chipkarten**

Feldtest: Altersgerechtes Wohnen mit Sensorik



Betriebssystemfamilie

Feldtest: Altersgerechtes Wohnen mit Sensorik Wenig Sensorwerte, viele Rückschlüsse



- Messung der Luftqualität in Badezimmer und Küche
- Messung von Feuchtigkeit, Temperatur, CO₂, CO
- Definition von Schwellwerten zur Detektion von Notfällen
- Regelmäßige Übertragung der Messwerte ans Backend
- Initiierung von Maßnahmen durch den Notfall-Pflegedienst im Falle auffälliger Werte

Messwert	Verdacht auf	Aktion
Mehr Toilettenbesuche	Durchfall, Dehydrierung	Assistance / Pflegedienst aktivieren
Weniger Toilettenbesuche	Mangelnde Flüssigkeitsaufnahme, Dehydrierung	Assistance / Pflegedienst aktivieren
Regelmäßiger Küchenaufenthalt	Regelmäßige Nahrungsaufnahme	Keine Aktion erforderlich
Überdurchschnittlicher CO ₂ -Wert	Soziale Kontakte	Keine Aktion erforderlich

Auszug aus IBM Angebot Elderly Care

Überblick Computer-Hardware

- Vereinfachtes Modell (PC)

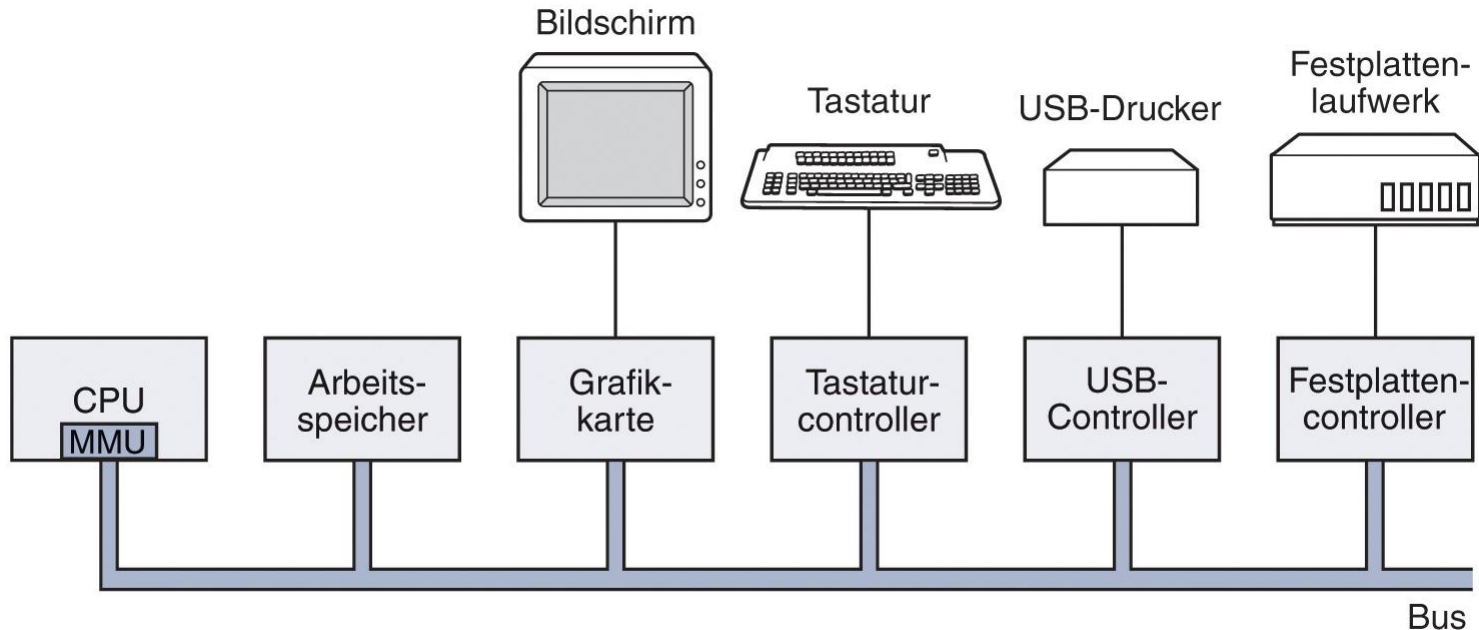


Abbildung 1.6: Einige Komponenten eines einfachen PCs

- Betriebssystem **muss** Details der Hardware kennen
 - Abstrahieren für Programmierer
 - Verwalten der Ressourcen

- **Prozessor**

- Gehirn des Computers
 - Hole Befehle aus dem Speicher und führe sie aus!
 - Abarbeitung von Programmen
- Unterschiedliche CPU-Typen haben unterschiedliche Menge von Befehlen
 - Pentium-Programm läuft nicht auf SPARC Maschine
- Laden von Befehlen dauert länger als Ausführung
 - Optimierung durch Register (Speicherbereiche) innerhalb der CPU
 - Ganzzahl-, Gleitkomma-Register
 - Befehle um ein Wort vom Speicher in Register zu schreiben
 - Befehle um ein Wort vom Register in Speicher zu schreiben
 - Befehle kombinieren zwei Operanden aus Registern

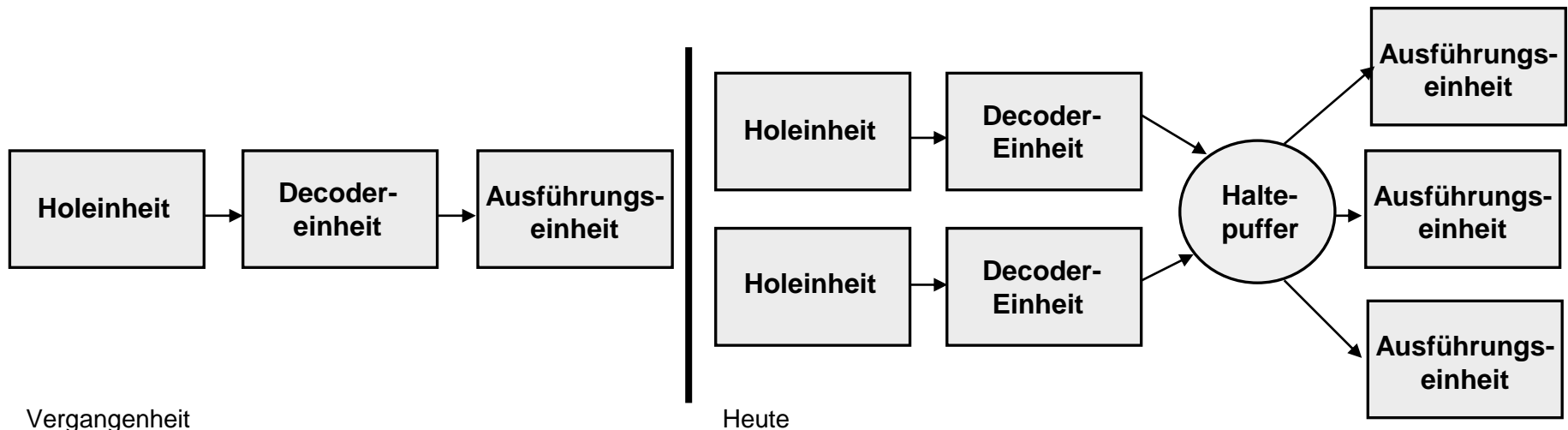
- **Prozessor ...**

- Spezialregister

- Befehlszähler (Program Counter PC)
 - Enthält die Speicheradresse des nächsten Befehls
 - Kellerregister (stack pointer)
 - Zeigt auf das Ende des aktuellen Kellers/stack
 - Hier werden „frames“ (Rahmen) für jede angesprungene, aber nicht beendete Prozedur abgelegt
 - Eingabeparameter, lokale Variable, ...
 - Programmstatuswort (Program Status Word, PSW)
 - Enthält Status-Bits, CPU-Priorität, Modus
 - Kernel Modus, Benutzer Modus
 - Begrenzter Zugriff für Benutzermodus
 - lesen ja, schreiben teilweise
 - **Wichtig bei Systemaufrufen und Ein-/Ausgabe**

- **Prozessor ...**

- Verwaltung durch Multiplexing
 - Zeitliche Aufteilung der CPU Ressource
 - Halte laufende Programm an und starte anderes!
 - Betriebssystem muss alle Register kennen
 - Speichern der Register und späteres Wiederherstellen
- Moderne Prozessoren
 - Ausführung mehrerer Befehle zur gleichen Zeit



- **Prozessor ...**

- Ausführungsmodi
 - Maßnahme, um den direkten Zugriff auf Systemressourcen durch Anwendungsprogramme zu unterbinden
 - Modus wird durch Bit im PSW (Programmstatuswort) gesetzt
- System-/Kern-Modus (kernel mode)
 - Jeder Befehl des Befehlssatzes kann ausgeführt werden
 - Jede Eigenschaft der Hardware kann ausgenutzt werden
- Benutzermodus (user mode)
 - Eingeschränkter Zugriff
 - Speicher nur über Speicherverwaltung
 - Keine privilegierten Bereiche
z.B. Ein-/Ausgabe

- **Prozessor ...**

- Systemaufruf (kontrollierter Moduswechsel)
 - Ein Benutzerprogramm nutzt Dienst des Betriebssystems
 - Spezieller Befehl (Systemaufruf, TRAP, system call)
 - bei Ausführung des Befehls:
 - Prozessor sichert PC im Keller (=Rückkehradresse)
 - Umschalten in Systemmodus
 - Verzweigung an vordefinierte Adresse im BS
 - BS analysiert Art des Systemaufrufs und führt den Aufruf aus
 - Rückkehrbefehl schaltet wieder in Benutzermodus
 - Andere Unterbrechungen erfordern das BS zu handeln
 - Interrupts (von Hardware erzeugt)
 - Exceptions (durch Programmfehler)

Aufgabe/Frage

Fragen:

- Bitte erklären Sie
 - TRAP
 - Interrupts
 - Exceptions

Bedingungen:

→ Zeit: ad hoc

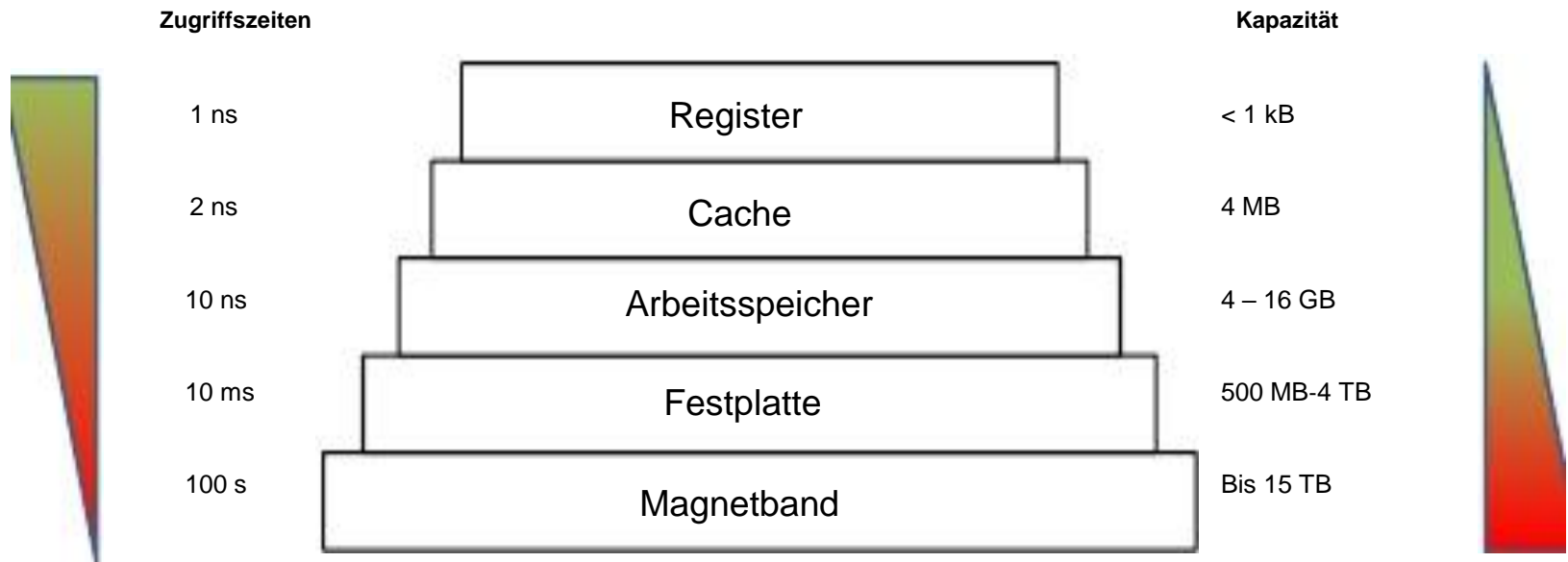


ad hoc

- **Prozessor ...**

- Entwicklung der Prozessoren geht weiter!
 - Hardware-Unterstützung
- Multi-Threading
 - Mehrere Threads in einem Prozessor mit schnellem Umschalten in nsec (10⁻⁹ sec)
 - Es gibt keine echte Parallelität
- Multi-Core
 - Eigene unabhängige Prozessoren

- **Speicher**



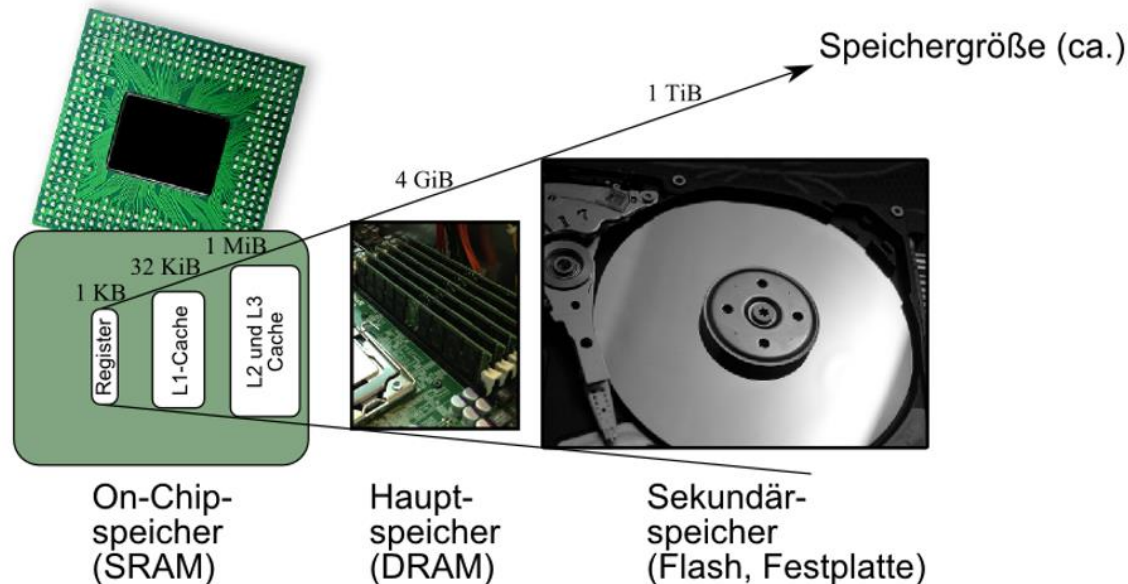
- Verschiedene Komponenten
 - Verwaltung durch Betriebssystem
 - Optimierung der Speicherverwendung
 - Cache-Verfahren



LTO 7:
Unkomprimiert
6 TB / 750 MB/s
Komprimiert
15 TB / 300 MB/s

„LTO2-cart-wo-top-shell“ von Austinmurphy in der Wikipedia auf Englisch.
Lizenziert unter CC BY-SA 3.0 über Wikimedia Commons
<https://commons.wikimedia.org/wiki/File:LTO2-cart-wo-top-shell.jpg#/media/File:LTO2-cart-wo-top-shell.jpg>

- **Speicherhierarchie**

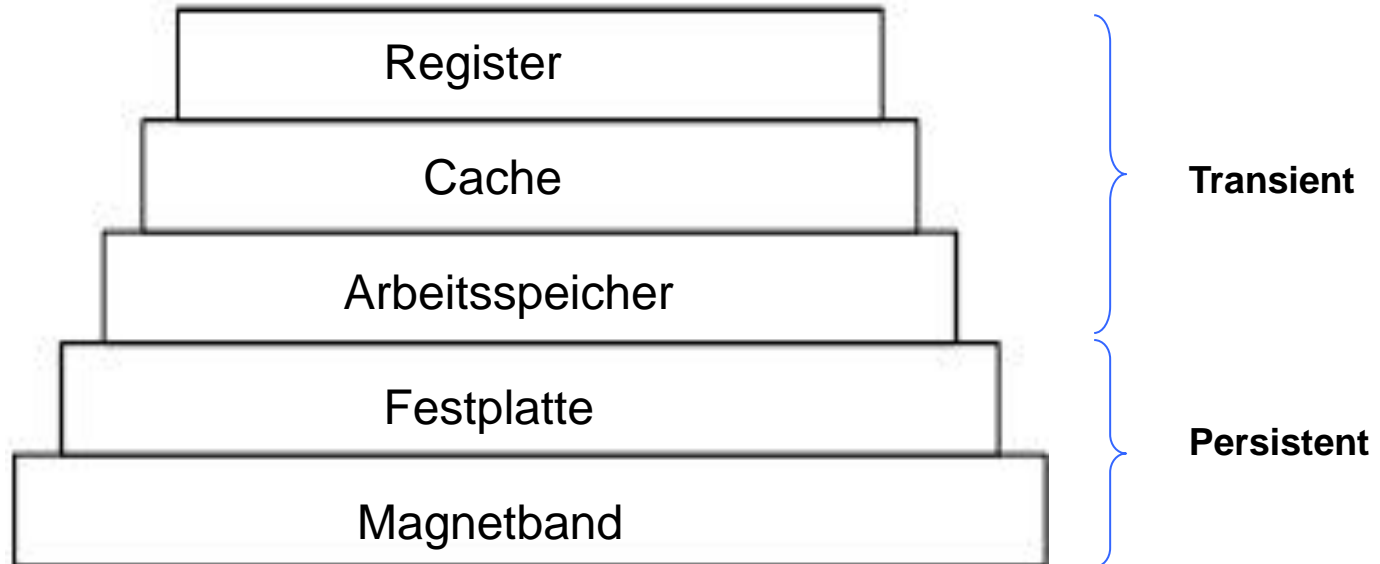


Erklärung KiB, MiB, GiB am Beispiel TiB:

Ein TiB (Tebibyte) ist eine Maßeinheit, die zur Beschreibung der Datenkapazität verwendet wird. Die Vorsilbe tebi stammt aus dem binären System zur Messung der Datenkapazität. Dieses System basiert auf Zweierpotenzen. Ein Tebibyte entspricht 2^{40} oder 1.099.511.627.776 Bytes.

Unterschied 1 MB zu 1 MiB: ein Megabyte kommt auf 1.000.000 Bytes; ein MiB steht für 1.048.576 Bytes

- **Speicher**



- Transienter (flüchtiger) Speicher
 - Daten gehen beim Ausschalten verloren
- Persistenter (dauerhafter) Speicher
 - Die Daten stehen langfristig zur Verfügung

- **Speicher...**
 - Register, Cache
 - Sehr nahe an der Prozessoreinheit
 - RAM (Random Access Memory)
 - Arbeitstier des Speichersystems
 - Was der Cache nicht kann, macht der RAM!
 - Andere Speicher
 - ROM, EEPROM, Flash, CMOS
 - Festplatte
 - Ermöglichen „Virtuellen Speicher“
 - Programme laufen, die größer als der physische Speicher sind
 - Verschiedene Zugriffszeiten
 - Hardwareunterstützung durch MMU auf CPU

Aufgabe/Frage

Fragen:

- Was bedeutet
 - MMU
 - CMOS
 - DMA
 - RAM
 - ROM?
- Was ist eine SSD? Was ist Flash?

Bedingungen:

→ Zeit: ad hoc



ad hoc

- **Speicher...**
 - Magnetbänder
 - Sicherungsmedium für Festplatten
 - Speicher sehr großer Datenmengen
 - Externes Ein-/Ausgabegerät
- **Ein-/Ausgabe-Geräte**
 - Integration in Computer durch Controller-Ansatz
 - Bietet vereinfachte (aber noch komplexe) Schnittstelle an
 - Spezielle Hardware, oft mit eigenem Mikroprozessor
 - Steuert das Gerät weitgehend autonom
 - Kann Interrupts senden
 - Geräte-Treiber
 - Software, die mit Controller kommuniziert
 - muss im Kernmodus laufen, also Teil des BS sein!

- **Ein-/Ausgabe-Geräte...**

- Anbindung an CPU
- Speicher-basierte E/A
 - Register des Controllers sind in Speicheradressraum eingeblendet
 - Normale Schreib- und Lesebefehle
 - Zugriffsschutz über MMU
- Separater E/A-Adressraum
 - Zugriff auf Controller-Register nur über spezielle (privilegierte) E/A-Befehle
- Beides im Einsatz

- **Ein-/Ausgabe-Geräte...**

- Arten der Ein- und Ausgabe

1. Aktives Warten

- Benutzerprogramm startet Systemaufruf
- System startet die E/A mit Treiber
- System wartet in Endlosschleife, bis die E/A Operation zu Ende ist
 - Falls beendet, Speichern der Daten
- Rücksprung in Benutzerprogramm
- Nachteil:
 - CPU wartet aktiv
 - CPU kann für keine anderen Aufgaben verwendet werden

- **Ein-/Ausgabe-Geräte...**

- Arten der Ein- und Ausgabe ...

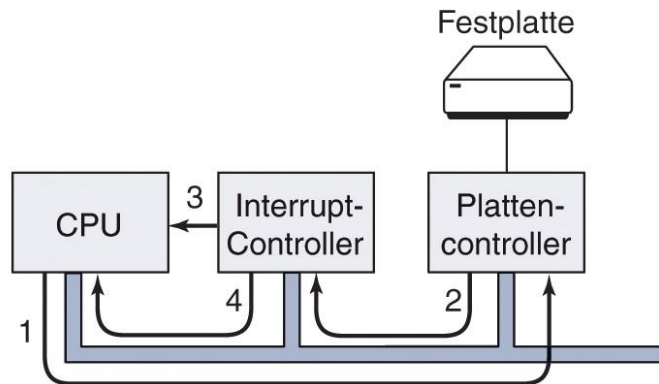
2. Interrupt

- Benutzerprogramm startet Systemaufruf
- System startet die E/A durch Controller mit Treiber
- Wenn Controller fertig ist, sendet er ein Signal an
den Interrupt-Controller über speziellen Bus
- Interrupt-Controller sendet Signal an CPU
- CPU behandelt Interrupt durch Wechsel in Kernmodus
 - Sprung an Unterbrechungsbehandlungsroutine (interrupt handler) und Ausführung
- Rückkehrbefehl schaltet wieder in Benutzermodus
- Hauptanwendung: Ein-/Ausgabe

- **Ein-/Ausgabe-Geräte...**

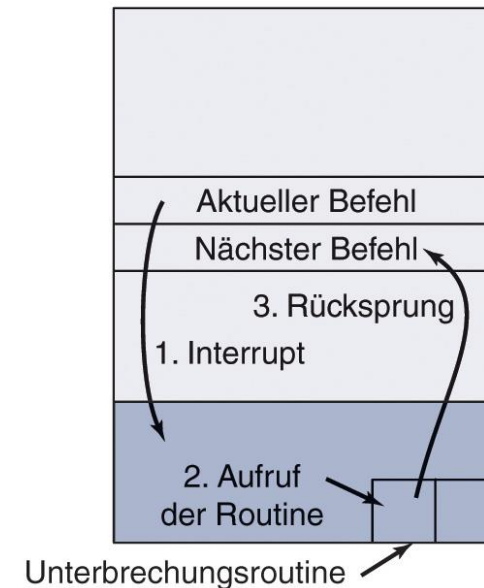
- Arten der Ein- und Ausgabe ...

- 2. Interrupt



a

0. Benutzerprogramm startet Systemaufruf
1. System startet die E/A durch Controller mit Treiber
2. Wenn Controller fertig ist, sendet er ein Signal an den Interrupt-Controller über speziellen Bus



b

1. Interrupt-Controller sendet Signal an CPU
2. CPU behandelt Interrupt durch Wechsel in Kernmodus
3. Sprung an Unterbrechungsbehandlungsroutine (interrupt handler) und Ausführung
4. Rückkehrbefehl schaltet wieder in Benutzermodus

- **Ein-/Ausgabe-Geräte...**

- Arten der Ein- und Ausgabe ...

3. DMA-Chip (Direct Memory Access)

- Regelt Datenfluss zwischen Controller und Speicher ohne CPU
 - Initialisierung durch CPU (Wieviele Bits wohin?)
 - Selbständige Aufführung
 - Interrupt nach der Beendigung der E/A
 - Behandlung wie zuvor

- **Bus-Systeme**

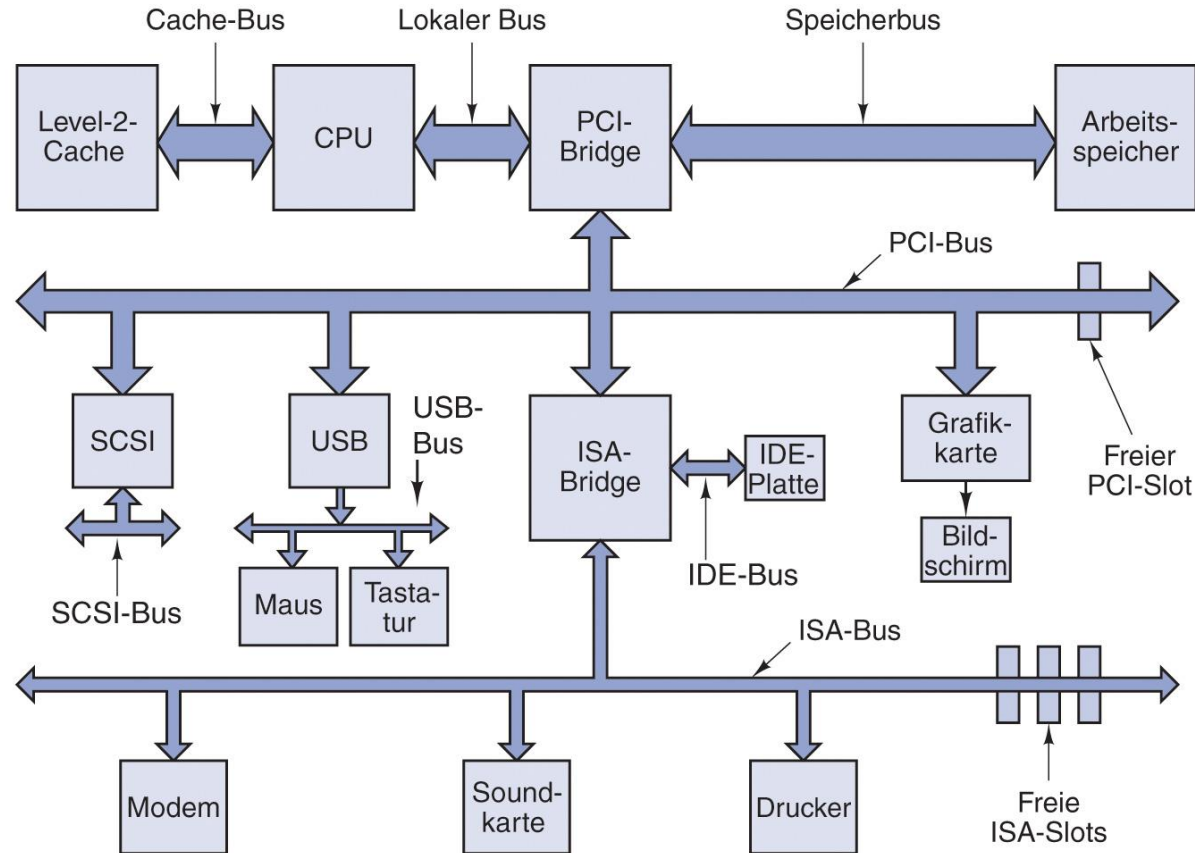


Abbildung 1.12: Der Aufbau eines ausgebauten Pentium-Systems

- **Betriebssystem muss unterschiedliche Geschwindigkeiten berücksichtigen**

Aufgabe/Frage

Fragen:

Sie kennen jetzt die Aufgaben von Betriebssystemen.

Sie kennen das einfache Computer-Modell.

Überlegen Sie sich die **wesentlichen Konzepte** eines Betriebssystems!

Bedingungen:

→ 5 min



5 min

Betriebssystemkonzepte

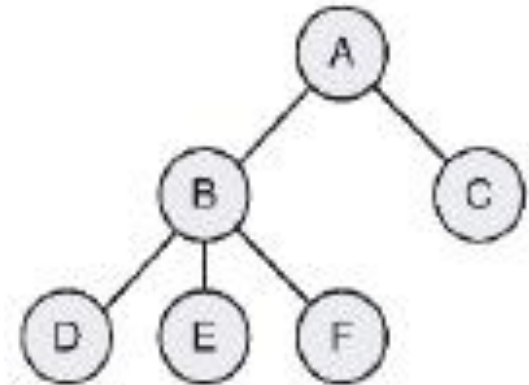
- **Prozess**

- Ein Programm in Ausführung
- Aktivitätseinheit mit Eigenschaften
 - Adressraum für Programm, Daten und Stack
 - Liste von Speicherstellen (0 bis Maximal-Wert)
 - Zustände und Ressourcen
 - Register (Befehlszähler und Kellerregister)
 - Liste der offenen Dateien
 - Liste der Fehlersignale
 - Verbundene Prozesse
 - Weitere Information für die Ausführung des Programmes
- Beispiel: Drei Programme in Ausführung
 - Multiprogrammierung notwendig
 - Betriebssystem muss die Verwaltung/Verteilung organisieren

Betriebssystemkonzepte

- **Prozess...**

- Verwaltung von mehreren Prozessen
 - Prozess-Tabelle
 - Speichert alle Information über Prozesse außer dem Inhalt des Adressraum
 - Teil des Betriebssystems
 - Speicherabbild
 - Adressraum eines angehaltenen Prozesses
- Systemaufrufe für Prozesse
 - In jedem System vorhanden
 - Erzeugung und Beendigung, etc...
 - Hierarchie von Prozessen
 - Evtl. mit Nachrichtenaustausch
 - Interprozesskommunikation



Betriebssystemkonzepte

- **Prozess..**
 - Signale
 - Nachrichten des Betriebssystems an Prozess
 - (Software-Variante) des Interrupts
 - Anhalten des Prozesses
 - Speichern im Stack
 - Spezielle Behandlungsmethode aufrufen

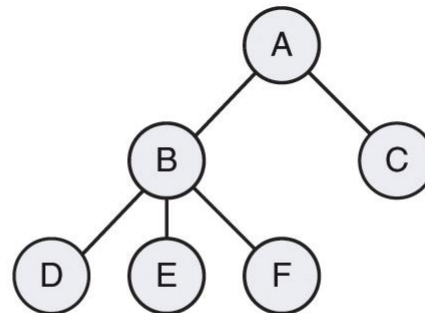


Abbildung 1.13: Ein Prozessbaum. A hat zwei Kindprozesse B und C erzeugt. Prozess B hat wiederum die Prozesse D, E und F erzeugt.

Betriebssystemkonzepte

- **Prozess..**

- Sicherheit von Prozessen

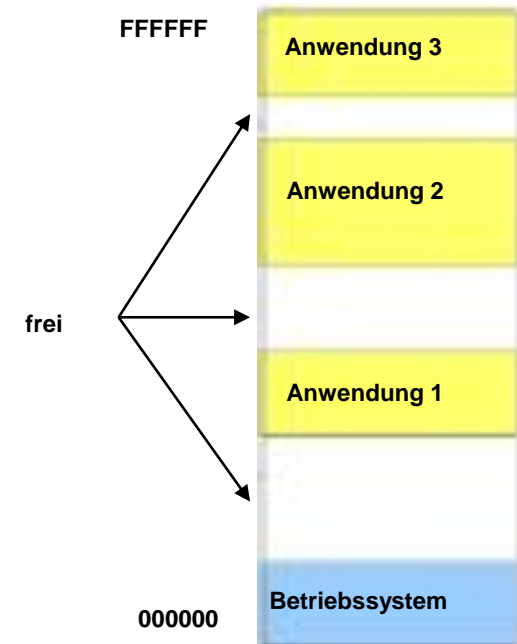
- Prozesse werden von Benutzern gestartet
 - merken sich die Benutzer ID, sind mit Benutzer verknüpft
 - BS verhindert, dass Prozesse von anderen Benutzern gestoppt werden

- Benutzer- und Gruppen-Konzept des Betriebssystem

- Benutzer-ID (user identification)
 - Unterschiedliche ID für jeden Benutzer des Systems
 - Gruppen-ID (group identification)
 - Menge von Benutzern
 - Administrator (root, super user)
 - Besitzt besondere Rechte

• Adressräume

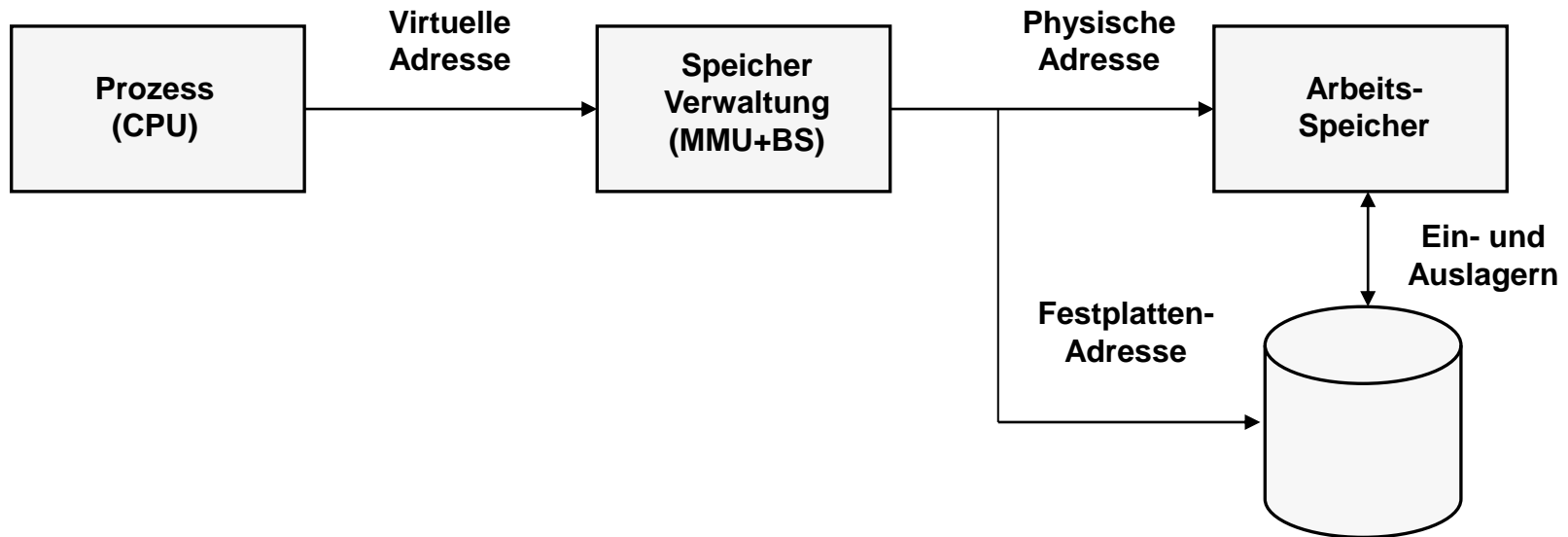
- Konzept zur Verwaltung des Arbeitsspeichers
 - Multiprogrammierung
 - Unterteilung des Arbeitsspeichers durch **Adressräume**
 - Keine gegenseitige Beeinflussung
 - Mechanismen oft in Hardware realisiert
 - Aber durch Betriebssystem verwaltet
 - Adressräume der Anwendungen
 - Von 0 bis Maximalem Wert!
 - entkoppelt vom physischen Speicher des Computers



Betriebssystemkonzepte

- **Adressräume**

- Problem: Begrenzter Speicher
 - Programm braucht mehr Speicher als physisch vorhanden ist
- Lösung: Virtueller Speicher
 - Teil des Adressraums im Arbeitsspeicher
 - Teil des Adressraums auf der Festplatte

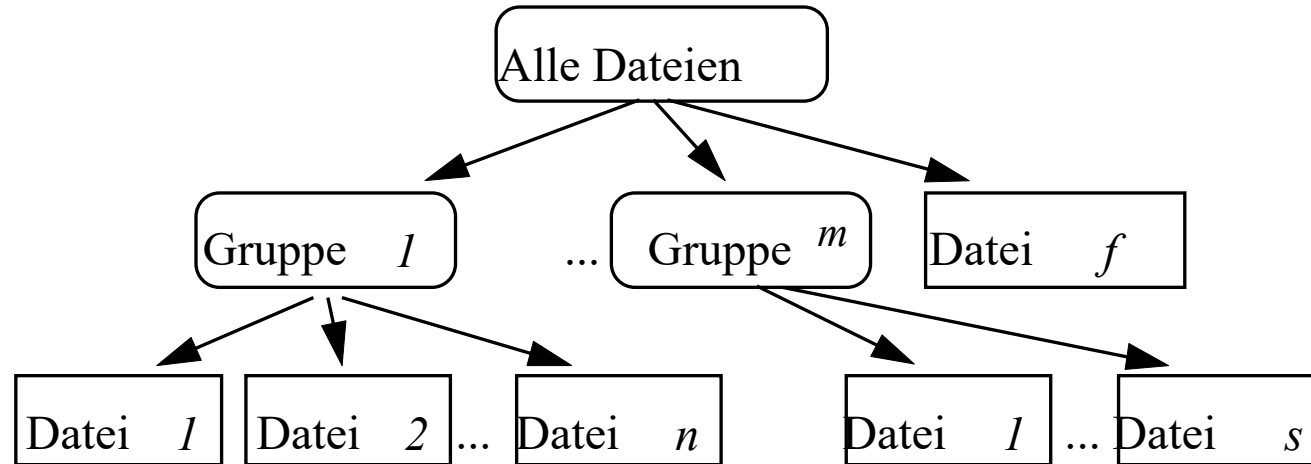


Betriebssystemkonzepte

- **Dateien und Dateisysteme**

- Datei = Einheit zum Speichern von Daten
 - Über das Ende des Programms hinaus (persistent)
- Verbergen der Details von Festplatten und anderen Speichergeräten
 - Abstraktes Modell von geräteunabhängigen Dateien
- Systemaufrufe für Dateien
 - Erzeugen, Verschieben, Öffnen, Lesen, Schreiben, Schließen, Löschen
 - Verzeichnis (Directory) für Verwalten von Dateien
 - Gruppierung von Dateien
 - Hierarchie von Dateien in einem Dateisystem

- Dateien und Dateisysteme



- Pfadnamen = Position im Baum

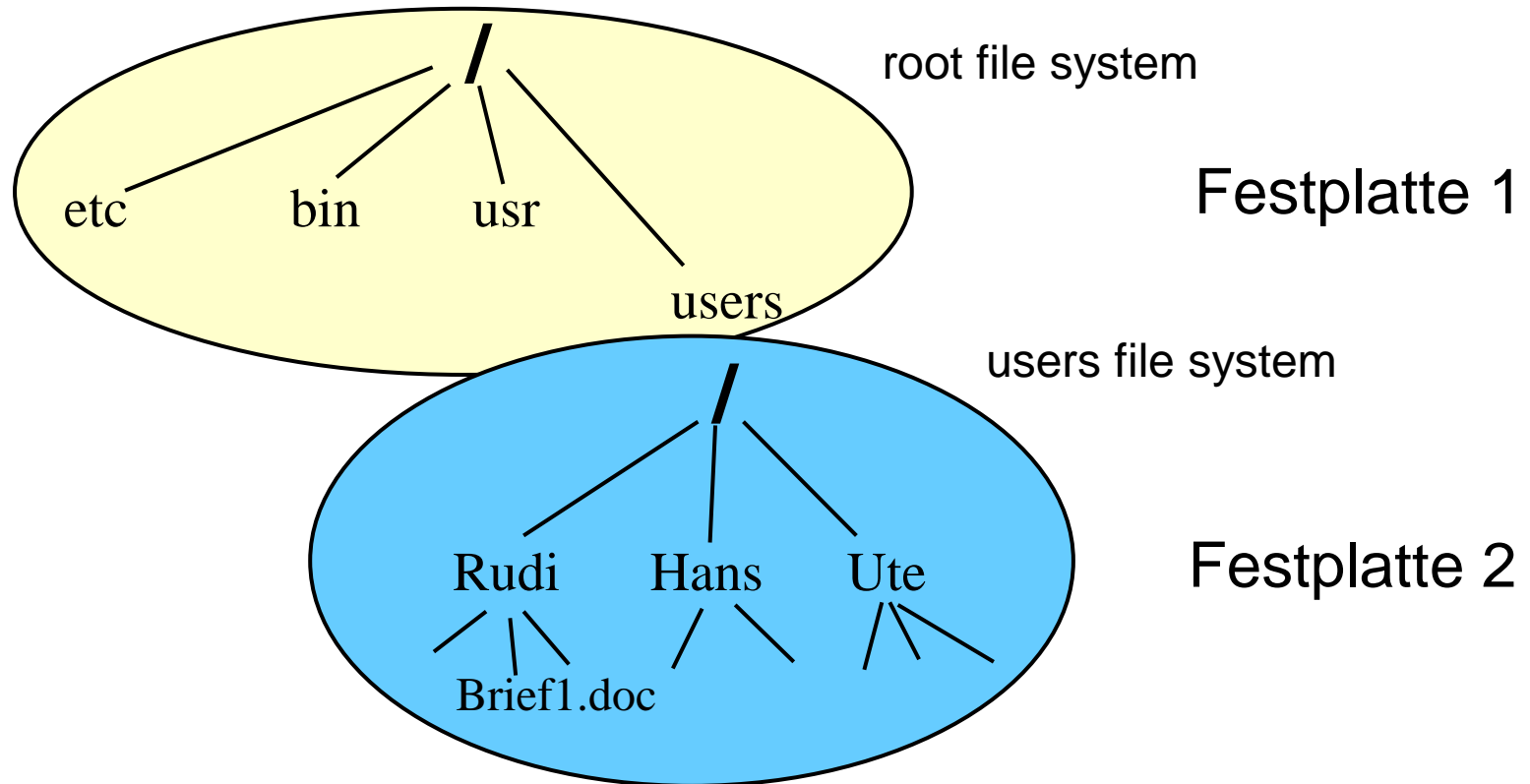
Betriebssystemkonzepte

- **Dateien und Dateisysteme**

- Wurzelverzeichnis (root directory)
 - Spitze der Hierarchie
 - Linux: nur eines im System
 - Weitere Geräte werden eingehängt (mounten)
- Arbeitsverzeichnis (working directory)
 - Aktuelles Verzeichnis = aktuelle Position im Baum
 - Relative Pfade möglich
 - Systemaufrufe notwendig zum Ändern
- Dateideskriptoren (file descriptor)
 - Beim Öffnen der Datei erzeugt
 - Falls Zugriff erlaubt ist
 - Sonst Fehlercode
 - wird dann von anderen Zugriffen verwendet

- **Dateien und Dateisysteme**

Einhängen eines Dateisystems (*mounten*)

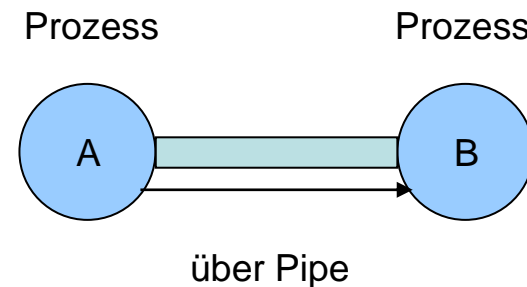
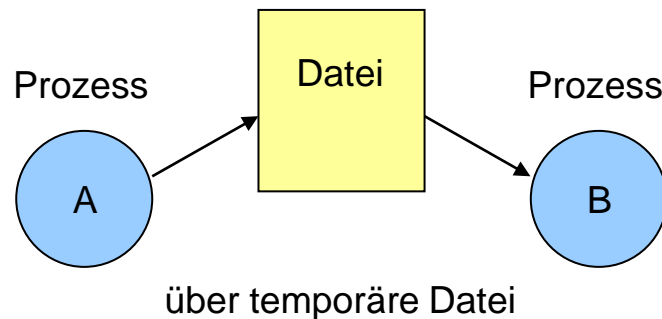


Zugriff mit `/users/Rudi/Brief1.doc`

Betriebssystemkonzepte

- **Dateien und Dateisysteme**

- Spezial-Datei (special file)
 - Einbinden von Ein-/Ausgabegeräten
 - **Blockdateien** (block special file)
 - Frei adressierbar (z.B. Festplatte, USB Stick)
 - **Zeichendateien** (character special file)
 - Unter UNIX unter /dev zu finden
- Pipe
 - Kommunikation zwischen Prozessen
 - ist dem Lesen und Schreiben sehr ähnlich
 - Unidirektionaler Kanal



Betriebssystemkonzepte

- **Weitere Konzepte**

- Kommen in den meisten Systemen vor

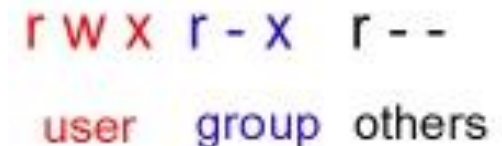
- **Datenschutz und Sicherheit**

- Authentifizierung:

- BenutzerID, GruppenID, Administrator
- Prozesse und Dateien haben Eigentümer
- Dateien haben auch Eigentümergruppen
- Zugriff auf Prozesse nur Eigentümer (und root)

- Autorisierung

- rwx-Bits beim Dateisystem
- Read, write, execute
- Besteht aus 3 x 3-bit Feldern
- Wenn das entsprechende Bit gesetzt ist, darf die Gruppe auf Datei entsprechend zugreifen
- Rechte werden bei jedem Zugriff geprüft



r	w	x	r	-	x	r	-	-
user			group			others		

Betriebssystemkonzepte

- **Benutzerschnittstelle (Shell)**

- Betriebssystem führt Systemaufrufe aus
- Kommandozeilen-Interpreter beliebtes Werkzeug in Betriebssystemen
 - DOS: DOS-Shell UNIX: Shell
 - Kein Teil des Betriebssystems, nutzt Systemaufrufe
 - Terminal als Standardeingabe/-ausgabe
 - Beim starten von Prozessen
 - date
 - Ausgabe in Dateien
 - date > date.txt
- Graphische-Oberflächen
 - ebenfalls kein Teil des Betriebssystems

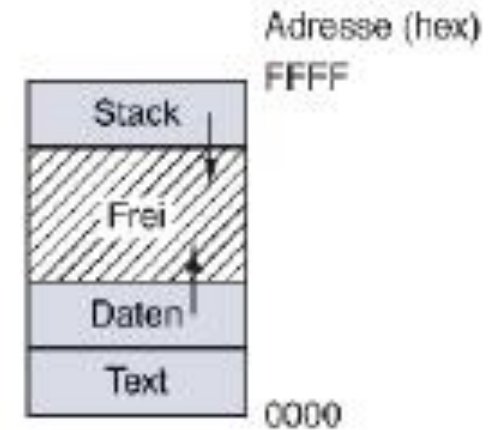
Systemaufrufe

- **Ablauf eines Systemaufrufs**

- BS sichert den vollständigen Prozessorstatus in der Prozesstabelle
- Aufrufende Prozess kann blockiert werden
- Rückkehr aus dem BS erfolgt über Scheduler
 - Rückkehr kann etwas verzögert sein
- Systemaufrufe sind bei PC die Hauptaufgabe des BS
 - Ressourcenverwaltung meist recht aufwändig
- Im folgenden POSIX Systembefehle
 - POSIX: Portable Operating System Interface (API)
 - IEEE Standard (DIN/EN/ISO/IEC 9945)

- **Systemaufrufe zur Prozessorverwaltung**

- Prozess-ID (PID) identifiziert Prozesse
- Speicheraufteilung von Prozessen
 - Textsegment für Programmcode
- Datensegment für Variablen
- Stacksegment



- Funktionen

- Fork - Erzeugen eines Kindprozesses
- Waitpid - Warten auf Beendigung eines Kindprozesses
- Execve - Speicherabbild eines Prozesses ersetzen /Ausführen eines anderen Programms
- exit - Beenden eines Programms

- **Systemaufrufe zur Dateiverwaltung**

- Funktionen

- open - Datei öffnen zum Lese, Schreiben oder Beides
 - close - Datei schliessen
 - read - Daten aus Datei in Puffer lesen
 - write - Daten aus Puffer in Datei schreiben
 - lseek - Dateipositionszeiger bewegen
 - stat - Status einer Datei ermitteln

- **Systemaufrufe zur Dateiverwaltung**

- Funktionen

- mkdir - Neues Verzeichnis erstellen
- rmdir - Löschen eines leeren Verzeichnisses
- link - Erzeugen eines neuen Eintrages, der auf einen anderen Eintrag im Verzeichnis zeigt
- unlink - Verzeichniseintrag löschen
- mount - Dateisystem einhängen
- umount - Eingehängtes Verzeichnis entfernen

- I-Nodes identifizieren Dateien

- Besitzer, Position auf der Platte, ...
- Verzeichnis ist Menge von I-Nodes und Namen

/usr/ast		/usr/jim	
16	mail	31	bin
81	games	70	memo
40	test	59	f.c.
70	note	38	prog1

- **Sonstige Systemaufrufe**

- Funktionen

- chdir - Verzeichnis wechseln
 - chmod - Dateirechte ändern
 - kill - Signal an einen Prozess senden
 - time - Zeit erfragen (seit dem 1.1.1970)

Aufgabe/Frage

Fragen:

Sie sollen ein Betriebssystem entwickeln.

- Überlegen Sie, welche Sprache Sie verwenden?
- Kommt für Sie Basic, Java oder C in Frage?
- Begründen Sie ihre Antwort!

Bedingungen:

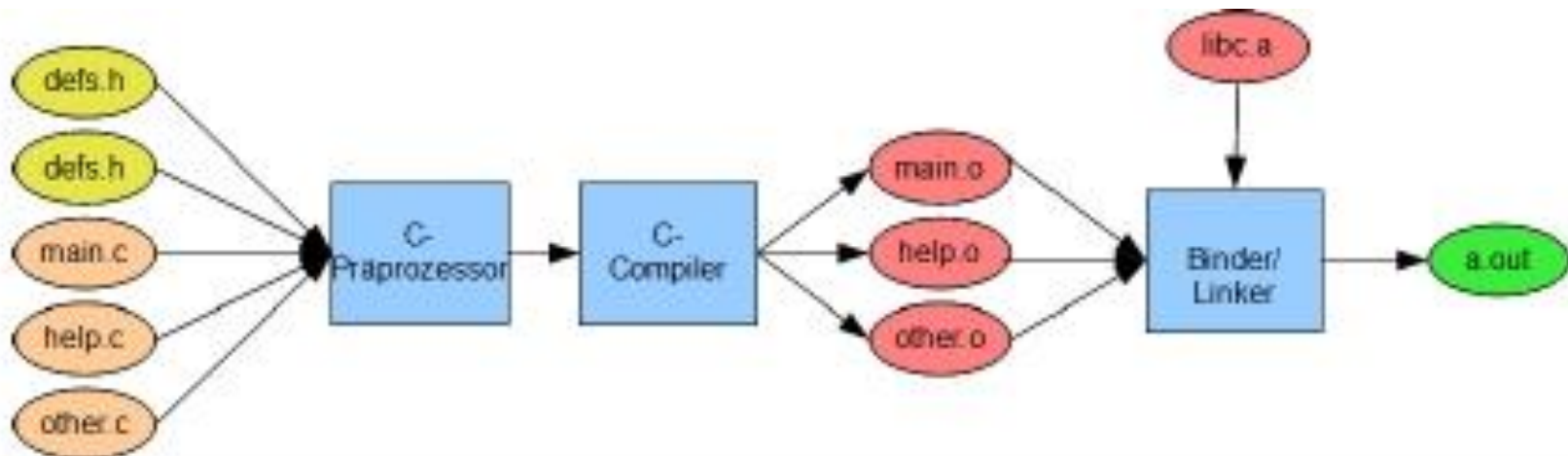
→ Zeit: ad hoc



ad hoc

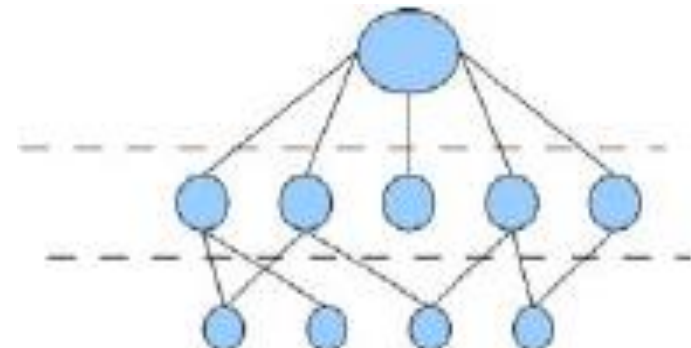
C und Betriebssysteme

- Programme bestehen aus
 - Header-Dateien (*.h)
 - Definitionen und Makros
 - Programm-Dateien (*.c)
 - Programmablauf-Code
- Übersetzungsprozess



Betriebssystemstrukturen

- **Innerer Aufbau von Betriebssystemen**
 - **Monolithische Systeme**
 - Häufigste Form
 - Eine große ausführbare Datei
 - Jede Funktion ist für andere Funktionen sichtbar
 - Monolithisch heißt nicht ohne Struktur
 - Hauptprogramm ruft Dienstprozedur auf
 - Dienstprozeduren führen Systemaufrufe aus
 - Hilfsfunktionen unterstützen Dienstprozeduren



- **Geschichtete Systeme**

5 - Operator
4 - Benutzerprogramme
3 - Ein/Ausgabeverwaltung
2- Operator-Prozess-Kommunikation
1 - Speicherverwaltung
0 - Prozessorzuteilung und Multiprogrammierung

Betriebssystemstrukturen

- **Mikrokerne**

- Sowenig wie möglich im Kernmodus laufen lassen
 - größere Stabilität, da nur wenige Programme das Gesamtsystem zu Fall bringen können

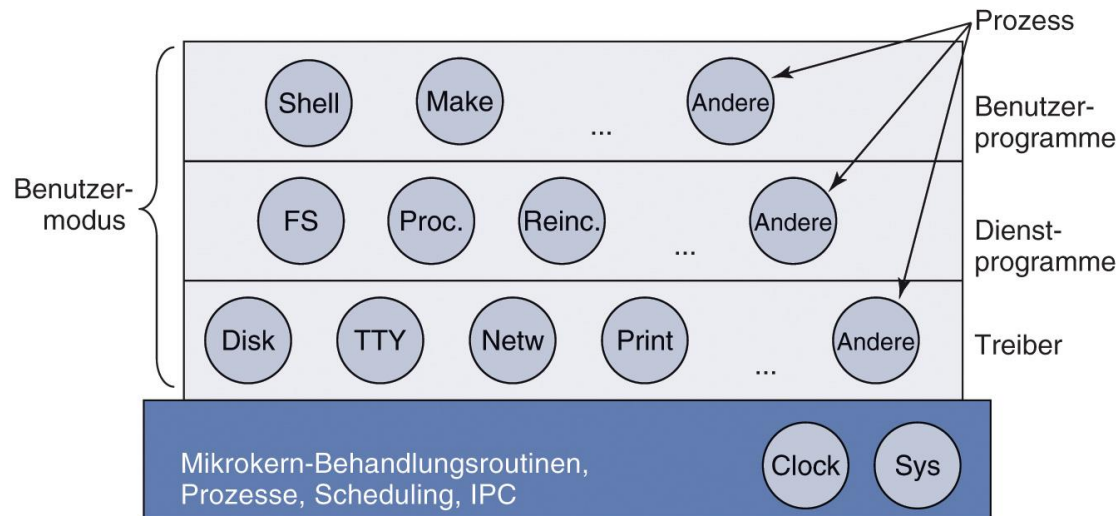


Abbildung 1.26: Struktur des MINIX-3-Systems

- Reincarnations-Server überprüft Dienstprogramme und startet bei Fehlern die Dienste wieder neu
- Einsatz in Echtzeit, industriellen, avionischen und militärischen Systemen

Betriebssystemstrukturen

• Client-Server-Modelle

- Extrapolation auf verteilte Systeme
 - Server stellen Dienste bereit
 - Clients nutzen diese Dienste
- Verteilung auf verschiedene Computer
 - Kommunikation über Netzwerk
 - Lokales Netzwerk
 - Fernnetzwerk

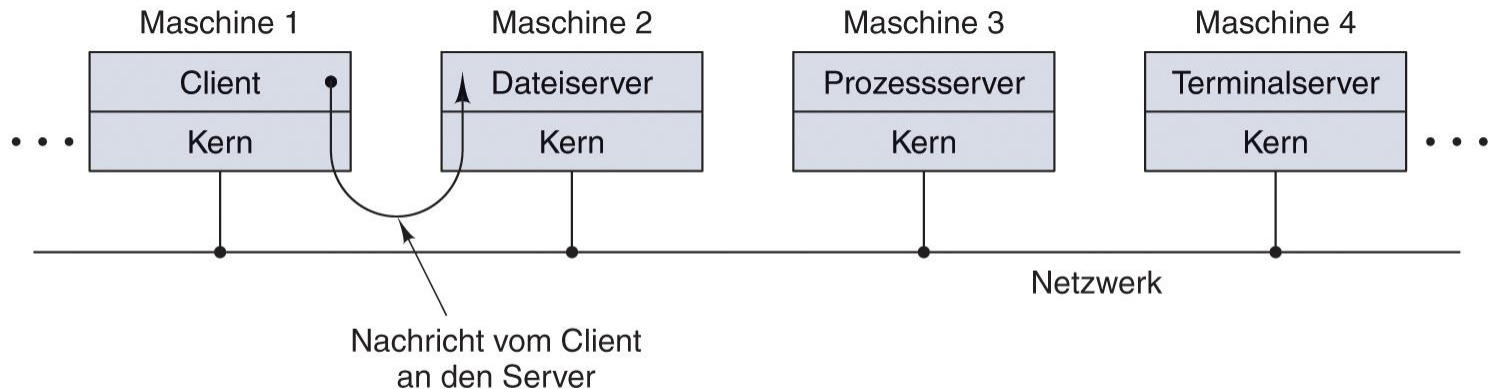


Abbildung 1.27: Das Client-Server-Modell über einem Netzwerk

- Funktionsweise des World Wide Web

Betriebssystemstrukturen

• Virtuelle Maschinen

- Aufgaben des Betriebssystems
 - Abstraktion der Ressourcen
 - Verwaltung der Ressourcen
 - Timesharing / Multiprogrammierung
 - Betriebssystem ist auch ein Programm
 - Können nicht mehrere Betriebssysteme auf einem System laufen?
 - Virtual Maschine Monitor
 - Conversational Monitor System
 - in den 1970er Jahren
 - Konsequente Abstraktion der Hardware
 - Seit 2000 auch auf PC's
 - Unterstützung durch Hardware
 - Multicore-CPU

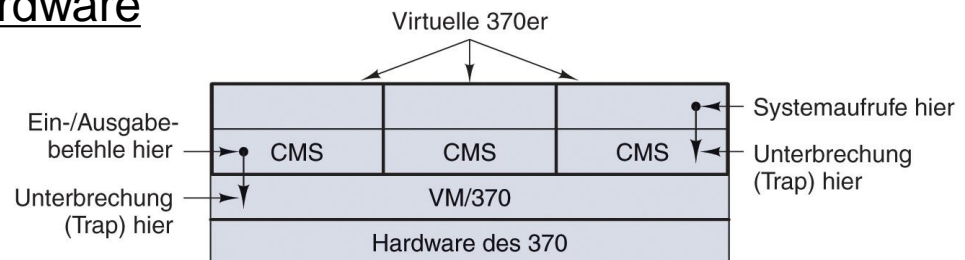


Abbildung 1.28: Die Struktur des VM/370-Systems mit CMS

- **Virtuelle Maschinen**

- Hypervisor Type 1
 - Entspricht VMM
- Hypervisor Type 2
 - Teil eines BS
 - Gastgebersystem kann noch Programme laufenlassen
- Sehr verbreitet
 - VBox, XEN, Vmware, KVM, MS Hyper-V
 - Einsatz in Rechenzentren
 - Grundlage für Cloud-Infrastrukturen



- **Zusammenfassung**
 - Aufgabe von Betriebssysteme
 - Abstraktion der Systemressourcen
 - Verwaltung der Ressourcen
 - Betriebssysteme liegen zwischen Anwendungen und Hardware
 - Zugriff auf Hardware nur über Betriebssystem
- **Historische Entwicklung**
- **Hardware eines Computer-Systems**
 - CPU
 - Ausführungsmodi
 - Systemmodus für Betriebssystem
 - Benutzermodus für Anwendungen
 - Unterbrechungen
 - Systemaufrufe, Interrupt

- **Zusammenfassung**
 - Grundkonzepte von Betriebssystemen
 - Prozesse
 - Speicherverwaltung
 - Dateiverwaltung
 - Sicherheit
 - Benutzerschnittstellen
 - Systemaufrufe
 - Programmierschnittstelle
 - Interne Strukturen von Betriebssystemen
 - Monolithisch, Schichtenarchitekturen, Mikrokerne, Client-Server, Virtuelle Maschinen