

# Technische Informatik 2

## **Software**

(inklusive Betriebssysteme)

Prof. Dr. Mirosław Malek

Sommersemester 2009

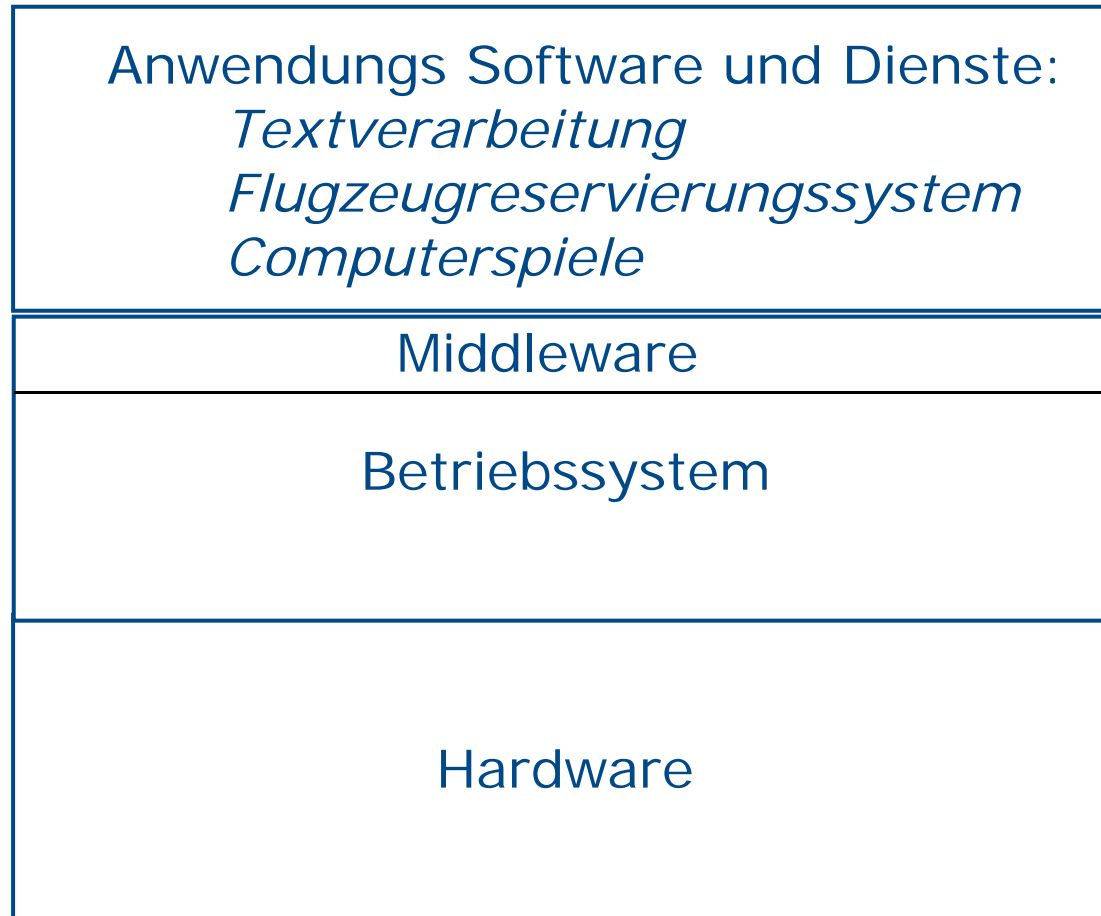
[www.informatik.hu-berlin.de/rok/ca](http://www.informatik.hu-berlin.de/rok/ca)



- Evolution der Software Schichten
- Übersetzer
  - Assembler                      AL → ML
  - Compiler                         HLL → ML
- Lader (Manuell, Bootstrap, Programm im ROM)
- Linker
- Programmabarbeitung (Stapelverarbeitung und Multiprogrammierung)
- Spooling
- Betriebssystem



# Allgemeiner Überblick

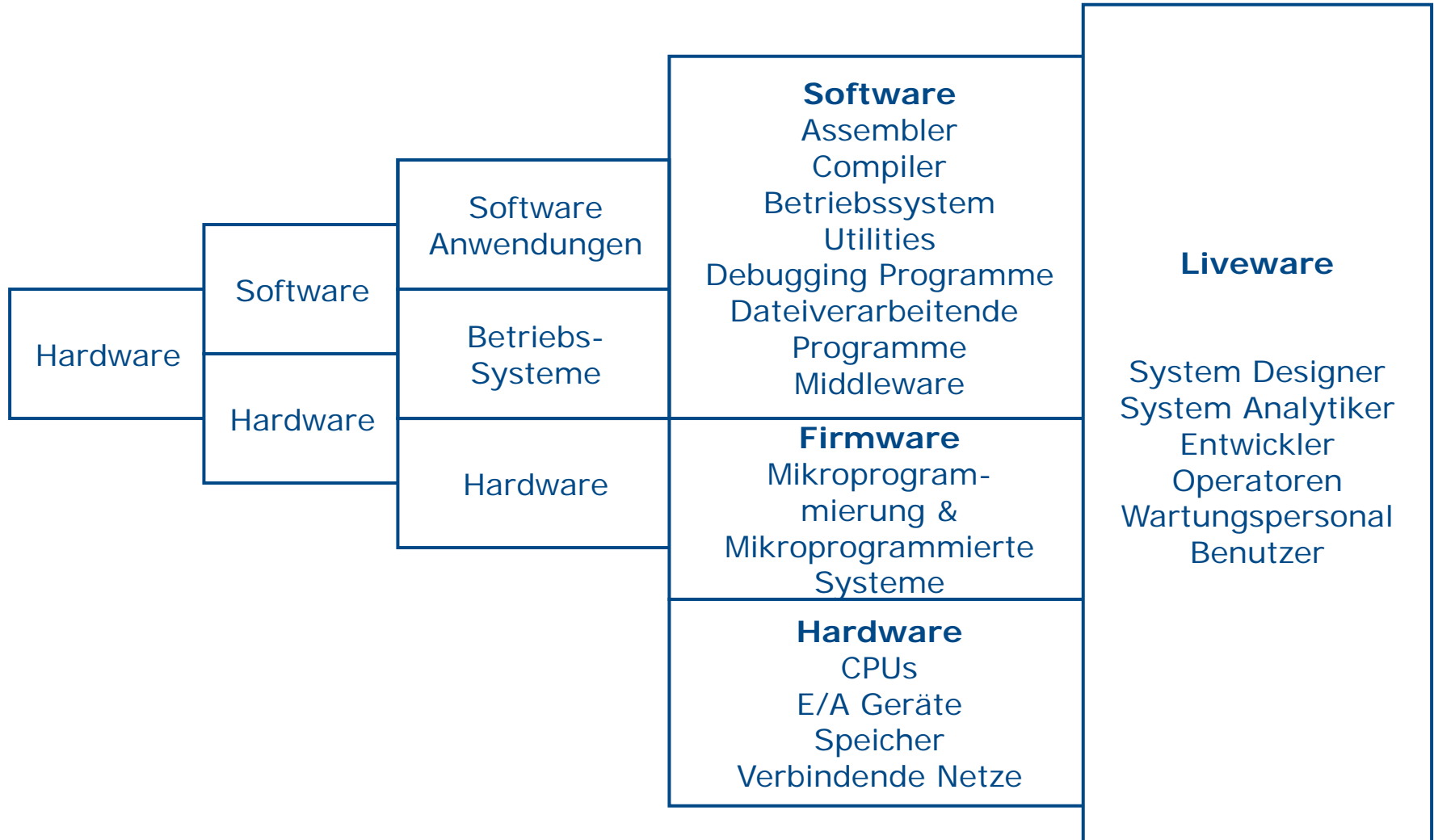




- Systemsoftware
- Compiler (Sprachprozessoren)
- Betriebssysteme
- Bibliotheken
- Benutzerprogramme
- Tools (Linker, Lader (Manuell, Bootstrap, Programm im ROM), Systemverwaltung, Monitoring)
- Dienste
- Anwendungssysteme (Sky is the limit)
- Behandlung von Fehlern und Programmausnahmen



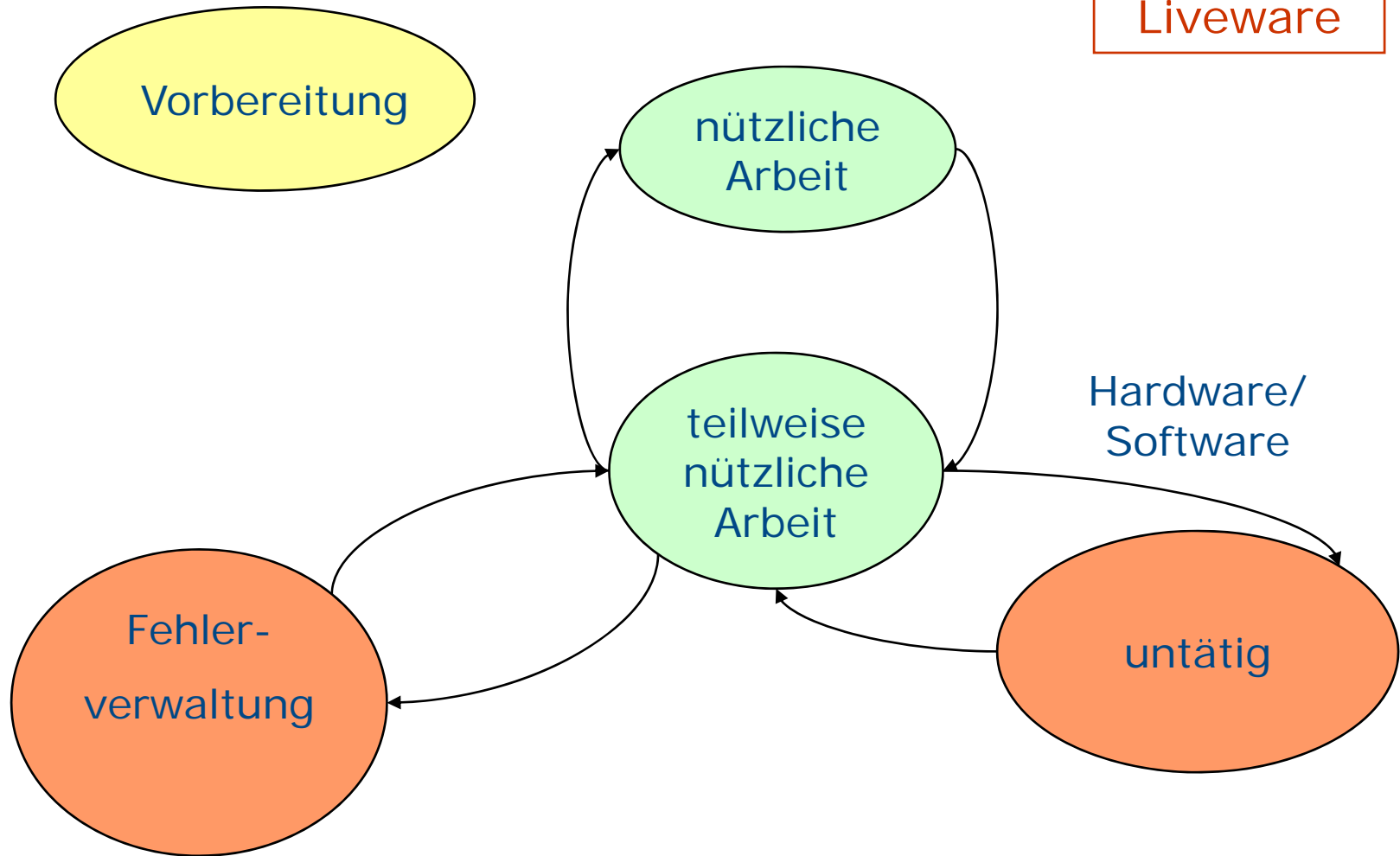
# Evolution von Software





# Lebenszyklus von Hard- und Software

Liveware



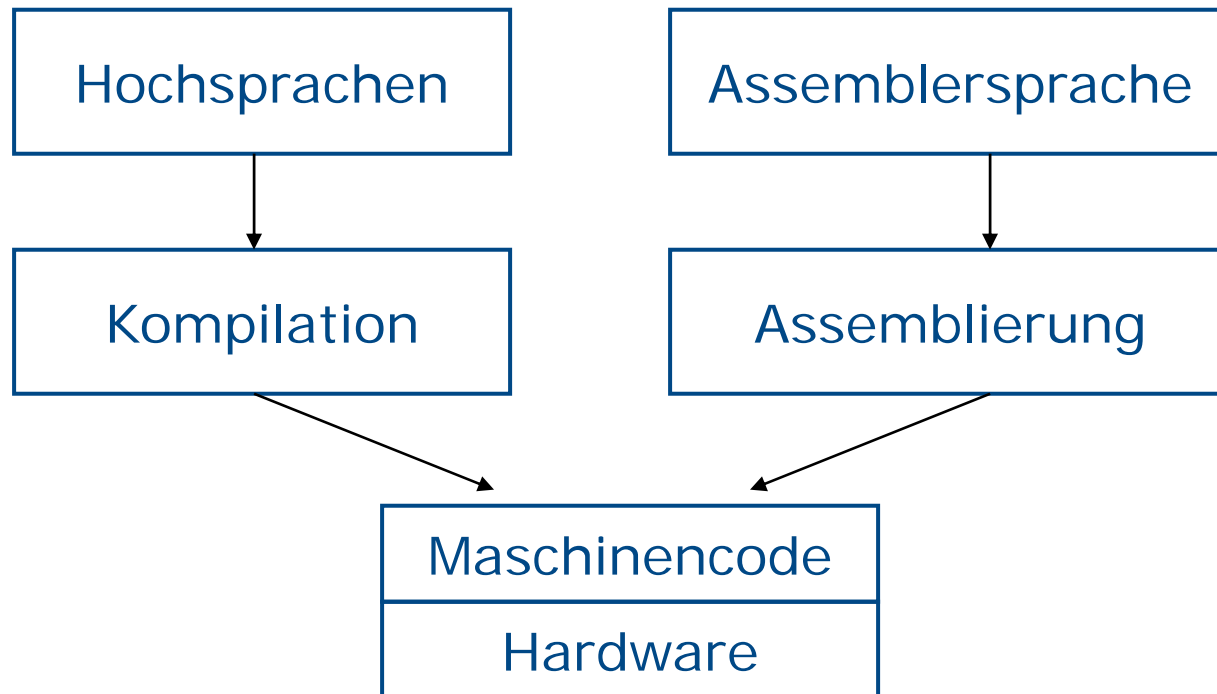


- Maschinensprache (binär, oktäl, hexadezimal)
- Assembler (Directiven, Assembler)
- Hochsprache (HLL – High-Level Language)
  - Betriebssteuerangaben
  - Compiler übersetzen Quellprogramm (Syntax der Sprache) und erzeugen das Objektprogramm
    - o Zwischensprache (Interpreter)
    - o Maschinenprogramm

Eine Hochsprachenanweisung entspricht typisch 1-20 Maschinencodemanweisungen. Eine Assembleranweisung entspricht typisch einer Maschinencodemanweisung, selten deutlich mehr (z.B. Floating Point)



## *Sprachen und Übersetzer (2)*





Wie laden wir ein Maschinenspracheprogramm in den Speicher? Wie veranlassen wir einen Computer, ein Programm auszuführen?

## Rechner einschalten:

- 1) Manuelles Laden eines Programms durch eine Konsole und Schalter.
  - a. Bootstrap (einfacher Lader)
  - b. Komplexere Lader (Paritätsprüfung, usw.)
- 2) Nach Reset arbeitet der Computer ein ROM-Programm ab, das weitere Programme lädt  
(als Basic Input Output System (BIOS) bezeichnet man die Firmware des Laders)



- Objektmodule sind assemblierte Versionen von verschiedenen Routinen
- Der Linker ist ein Programm, das mehrere Objektmodule zu einem Lademodul verbinden kann.
- Statischer Linker  
Einmaliges Linken nach Kompilation
- Dynamischer Linker  
Linken nach Bedarf zur Laufzeit

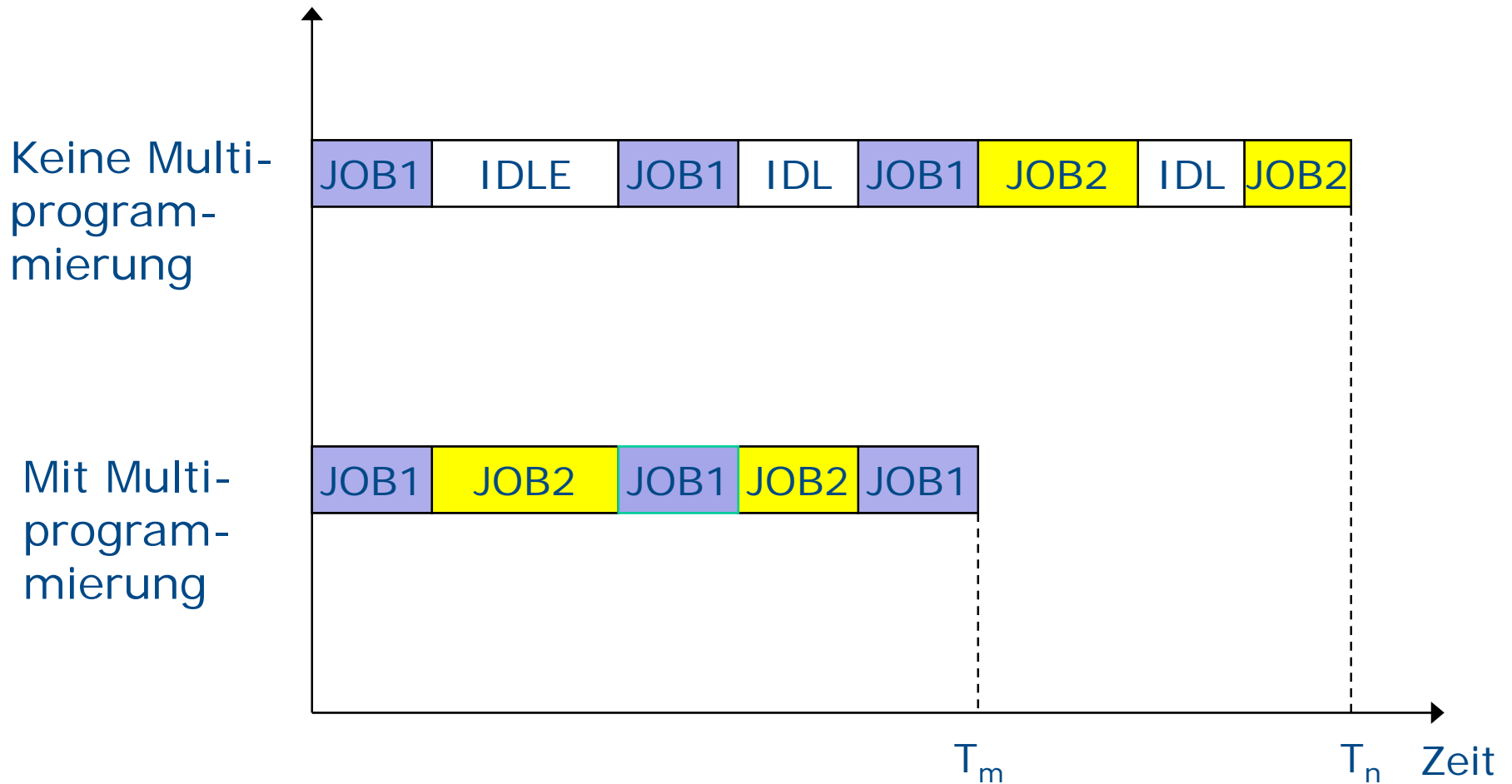


# *Programmabarbeitung: Typen*

- **Stapelverarbeitung**  
Stapelverarbeitung (Batch Processing):  
Jobs werden nacheinander in sequentieller Abarbeitung ausgeführt.
- **Multiprogrammierung**  
Das Konzept der Multiprogrammierung ist es, mehrere Programme im Hauptspeicher zu haben. Wenn die Ausführung eines Programms für E/A-Operationen ausgesetzt werden muß, wird die CPU veranlasst, eines der anderen Programme im Speicher auszuführen.
- **Generelles Problem: Ein/Ausgabe ist deutlich langsamer als Prozessor**



# Vergleich: Mit und ohne Multiprogrammierung



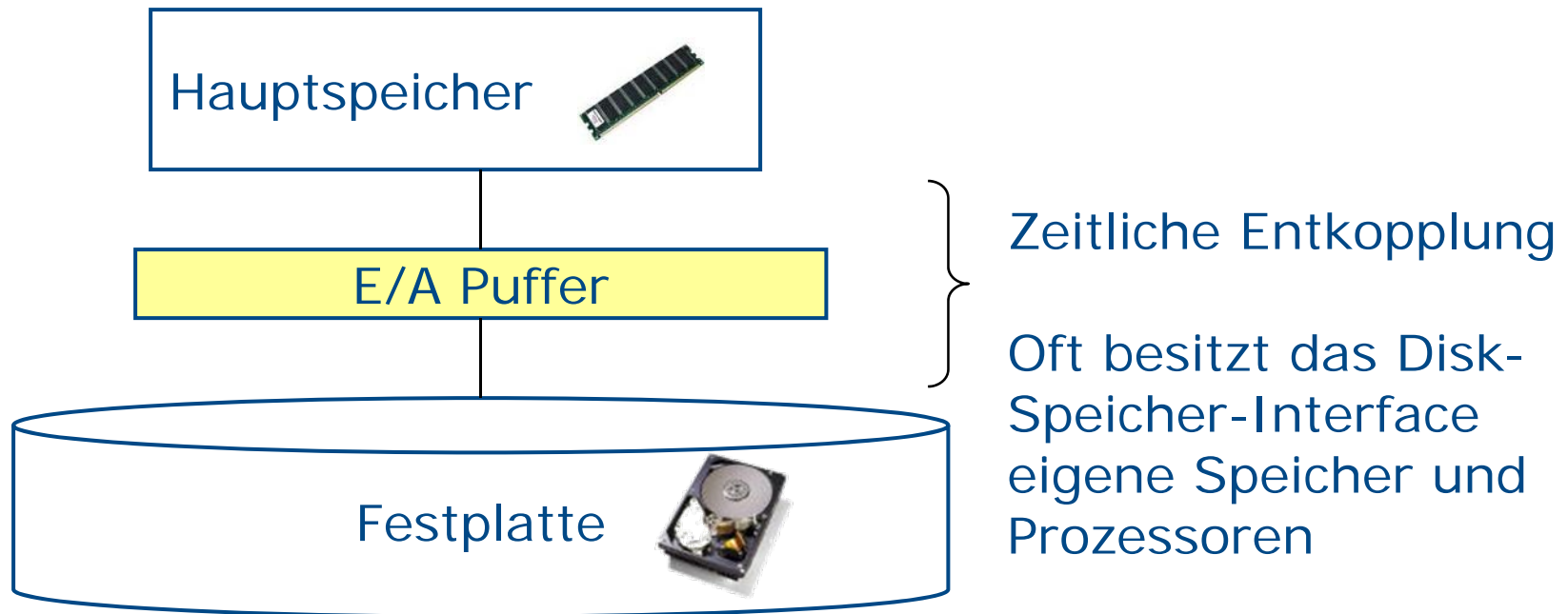


- Abarbeitung von parallelen Programmsträngen innerhalb *eines* Programms um die Wartezyklen der CPU effizient auszunutzen.
- Diese voneinander unabhängigen Programmteile, die simultan ausgeführt werden können, werden „Threads“ genannt.
- Diese Unabhängigkeit kann nur durch den Entwickler durch umsichtiges Programmieren sichergestellt werden.
- Multithreading erhöht die Geschwindigkeit um 20% bis 50% auf einer CPU



## Ziele:

- 1) Überwinden des Geschwindigkeitsunterschiedes von Hauptspeicher und Festplatte
- 2) Erhöhung der Geschwindigkeit und Glätten von E/A-Operationen

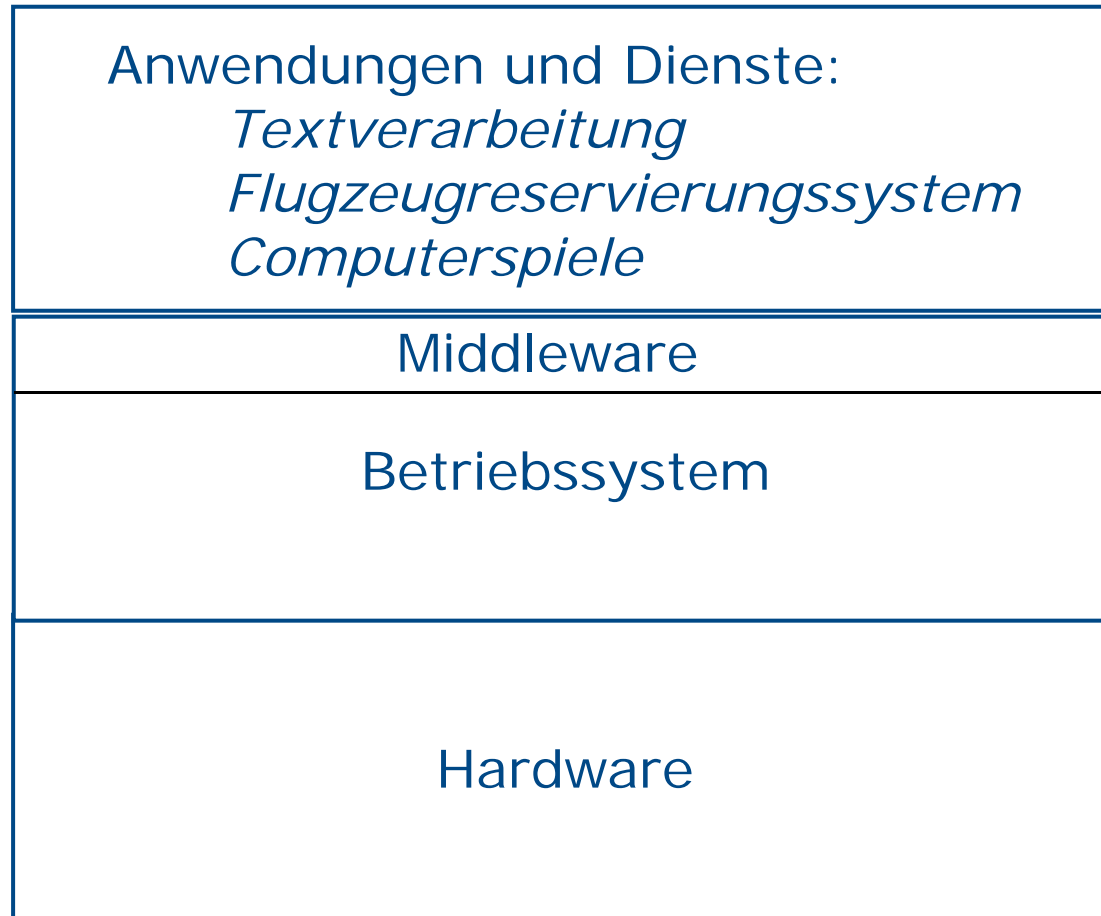




- Betriebssystem – Funktionalität
- Analogien
- Betriebssystementwicklung
- Scheduling von Prozessen
- Speichermanagement
- Systemsicht des Betriebssystems



# Allgemeiner Überblick





- Integraler Bestandteil des Computerdesigns
  - Laden
  - Linken
  - Speicherverwaltung
  - CPU-Verwaltung (Scheduling und Speicheranforderung)
  - Verwaltung der Ein/Ausgabe
- Betriebssysteme sind Programme, die die Ausführung von Programmen handhaben. Das Betriebssystem handhabt die CPU bezüglich der verfügbaren E/A Ressourcen. Ressourceteilung kann Kosten reduzieren, wenn dies zu einer Erhöhung des Systemdurchsatzes führt, z.B. mehr laufende Benutzerprogramme pro Zeiteinheit.



- BS = Regierung, es führt das Management der Ressourcen durch, aber es produziert nichts, da die Produktion den Anwendungsprogrammen obliegt
- BS = Fabrikmanager, ein Fabrikmanager erleichtert den Fluss zwischen den einzelnen Arbeitsschritten, produziert jedoch keine Güter
- BS ist ein Magier, es versteckt Software- und Hardwareressourcen und gibt dem Benutzer das „Gefühl“, einer problemfreien Welt, ohne typische Hardwaregrenzen



## *Einige weitere Sichten eines Betriebssystems*

- BS als Verkehrsampel, die Verkehrsampel reguliert den Fluss der Autos, das BS reguliert den Fluss der Prozessausführung (Eingabe/Ausgabe Operationen, Interrupts, Speicherschutz, usw.)
- BS als Wertpapierspekulant, Lernen aus Fehlern die in der Vergangenheit gemacht wurden, um den Profit (Performance) in der Zukunft zu maximieren
- BS als Stellglied (*controller*) - verbindet Benutzer, Programme und Prozesse, die Hardware benutzen
- BS als Mittler – dient als Vermittler zwischen Hardware und der Anwendungssoftware



1. Erschaffen und Beenden von Prozessen
2. Überblick über die und Kontrolle des Fortschritts der Ausführung der Prozesse
3. Laden und Linken der Programme
4. Speicherverwaltung
5. Scheduling, z.B.: definiert, in welcher Reihenfolge die Prozesse ausgeführt werden
6. Behandlung von Ausnahmen, Fehlern oder Schutzverletzungen
7. Zuordnung von Hardware- und Software-Ressourcen zu Prozessen



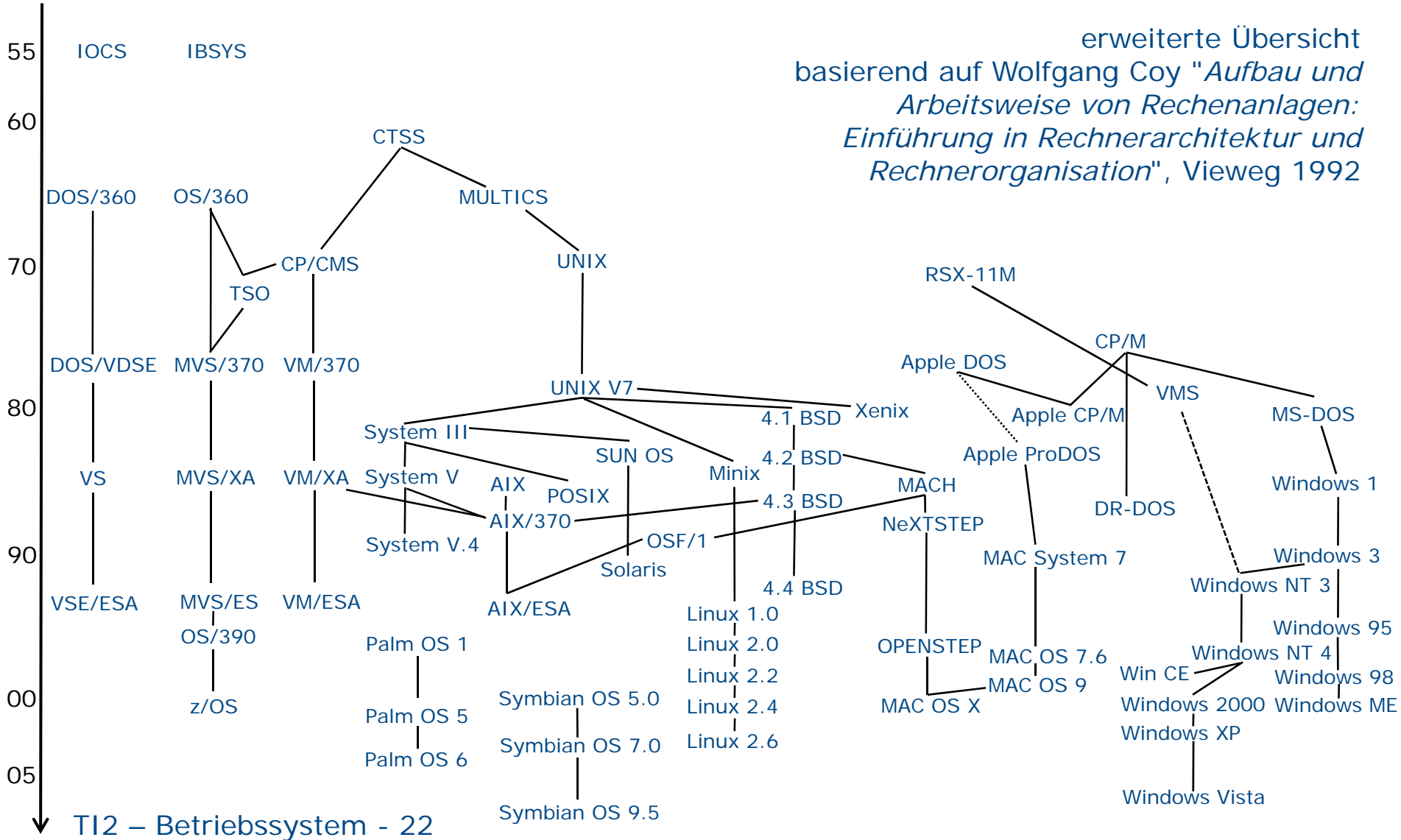
8. Bieten Zugriffe zur Software und Daten, z.B.:  
Compiler und Assembler, Bibliotheksprogramme,  
Text Dateien, Bilder, Musik
9. Sichert benötigte Qualitätseigenschaften (QoS)  
unter Beachtung von Sicherheits-, Echtzeit-,  
Zuverlässigkeit- und anderen Aspekten
10. Unterstützende Kommunikation und Dienste -  
auch ein Browser kann als Teil eines  
Betriebssystems angesehen werden

Ein Prozeß ist eine Routine, die als Teil der Abarbeitung eines geplanten Jobs ausgeführt wird. Beinhaltet Informationen, die die Unterbrechung und folgende Wiederaufnahme der Ausführung eines Programms erlaubt.



# Betriebssystementwicklung

erweiterte Übersicht  
basierend auf Wolfgang Coy "Aufbau und  
Arbeitsweise von Rechenanlagen:  
Einführung in Rechnerarchitektur und  
Rechnerorganisation", Vieweg 1992





# *Scheduling von Prozessen*

- Der Prozess wird in eine (oder mehrere) Warteschlange(n), in der auch die anderen, vom Betriebssystem auszuführenden Prozesse zu finden sind, eingereiht.
- Priorität als Mittel, Position in Warteschlange zu beeinflussen
- Information über den Zustand des Prozesses
  - Prozessnummer (id)
  - Prozesszustand
  - Schutz
  - Scheduling-Information
  - Statusinformation



## *Scheduling von Prozessen (2)*

- Kriterien für das Auswählen eines Prozesses:
  - CPU Ausnutzung
  - Durchsatz
  - Antwortzeit
  - Responsivität (Echtzeit System)
  - Priorität



# *Scheduling Strategien*

1. First-In-First-Out (FIFO)
2. Kürzester Job Zuerst [Shortest-Job-First (SJF)]
3. Prioritäten Scheduling  
(Priority Scheduling)
4. Round Robin Scheduling
5. Früheste Deadline zuerst  
[Earliest Deadline First (EDF)]
6. Monotones Scheduling [Rate Monotonic Scheduling (RMS)  
Algorithm ]
7. Minimalster Überhang (Least Laxity Scheduling)  
(Differenz zwischen Deadline und Ausführungszeit) Kleinerer  
Überhang = höhere Priorität



# *Zustandsdiagramm*

Ein Prozess durchläuft eine Anzahl von Zuständen:

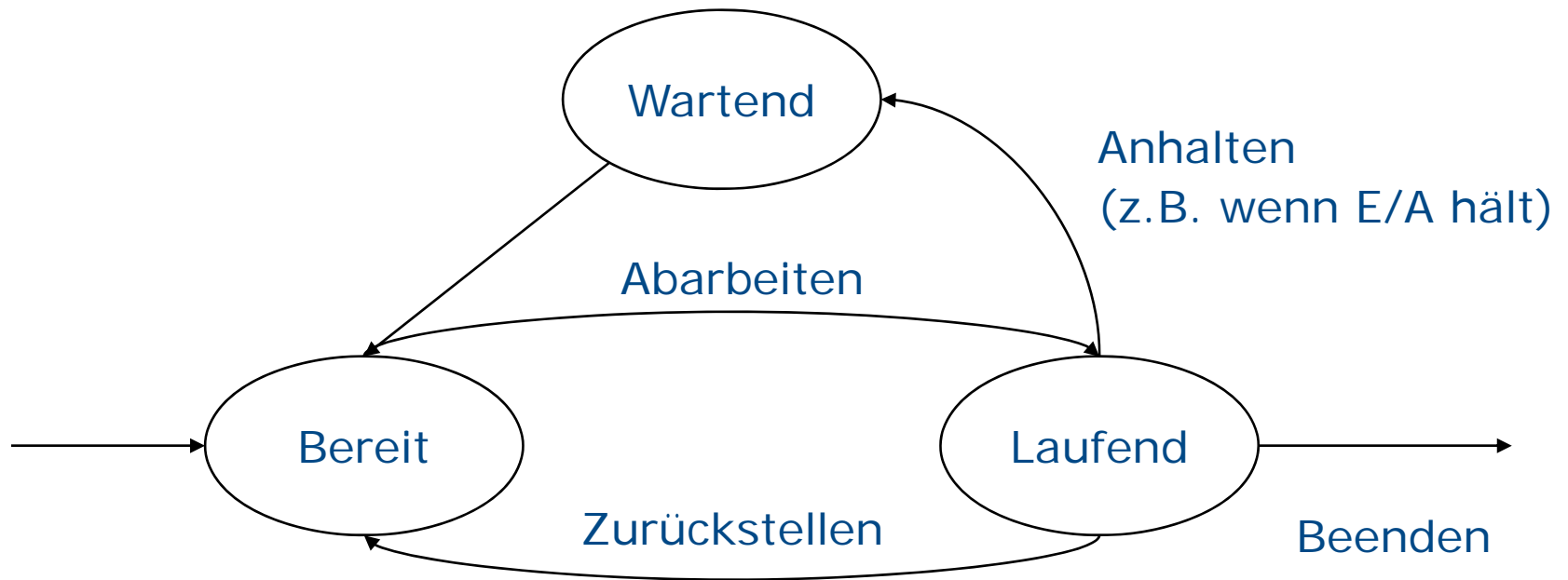
- ausführbar
- blockiert
- in Ausführung

Das Zustandsdiagramm illustriert die Steueraktivitäten, die für die Prozessverwaltung nötig sind. Die Beschriftungen entlang der Pfeile zeigen die Gründe der Änderung von einem Zustand in den anderen.

- Interrupts können Zustandsänderungen hervorrufen.
- Programme mit höherer Priorität können laufende Programme unterbrechen.



## Zustandsdiagramm (2)





## *Arten der Verarbeitung:*

Arten der Verarbeitung, die vom BS unterstützt werden:

1. Batch-Verarbeitung (Jobs werden gruppiert)
2. Transaktionenbasierte Verarbeitung (Operationen, wie Einkaufen mittels E-Commerce oder Banktransfers werden unterstützt)
3. Interaktive Verarbeitung (z.B.: „Browsen“ im Internet oder Spiele gehören dieser Kategorie an.)



## *Einige Aufgaben des OS aus Benutzersicht*

Betriebsmittel- verwaltung	Anwendungs- programme	Übersetzer
Prozessverwaltung	Editoren	Assembler
Speicherverwaltung	Tabellenkalkulation	Compiler
Datenverwaltung	Datenbank	Interpreter
Ein-/ Ausgabeverwaltung	Linker-/ Ladeprogramme	Makrointerpreter
Jobverwaltung	Dateibearbeitung	
Systembefehls- interpreter	Abrechnung	
Diagnose und Test	lokale Netzverwaltung	
Mehrprozessor- verwaltung	Telekommunikation	



# Systemsicht des Betriebssystems: Schichtmodell

Schichtname	Typische Objekte	Typische Operationen
Schaltkreise	Register, Gatter, Busse	NAND, NOR, XOR
Maschinensprache	Befehlszähler, ALU	Add, Move
Unterprogramm- verwaltung	Prozedurblock, Stack	Call, JSR, RTS
Interrupts	Fehlerrountinen	Busfehler, Reset
einfache Prozesse	Semaphor, Puffer	warten, bereitstellen, ausführen
lokaler Zusatzspeicher	Datenblock, E/A-Kanal	lesen, schreiben, öffnen, schließen
virtueller Speicher	Seiten	lesen, schreiben, austauschen
Prozesskommuni- kation	Übergabekanal (Pipe)	lesen, schreiben, öffnen



# Systemsicht des Betriebssystems: Schichtmodell (2)

Schichtname	Typische Objekte	Typische Operationen
...weiter...		
Dateiverwaltung	Dateien	lesen, schreiben, öffnen, kopieren
Geräteverwaltung	Zusatzspeicher, Terminals	lesen, schreiben
Ein-/Ausgabeströme	Datenströme	lesen, schreiben, öffnen, schließen
Benutzerprozesse	Benutzerprozesse	einloggen, ausloggen, Verzweigungen
Verzeichnisverwaltung	interne Tabellen	erstellen, löschen, ändern
Benutzeroberfläche	Fenster, Menüs, Icons	Betriebssystembefehle



## *Services und Serviceorientierte Architektur (SOA)*

- Services sind selbstbeschreibende, offene Komponenten, die eine kostengünstige Entwicklung und einen kostengünstigen Einsatz von verteilten Anwendungen erlauben
- Das Hauptziel serviceorientierter Architekturen ist die Unterstützung von transparenten, flexiblen und dynamischen Interaktionen zwischen Services und ihren Clients in vernetzten Domänen
- SOA - Leistungsmerkmale:
  - verbesserte Effizienz durch Auslagerung von Aufgaben/Geschäftsprozessen (outsourcing) und Wiederverwendung von Komponenten
  - erhöhte Flexibilität and Mobilität auf Geschäfts- und IT-Ebene, einfachere Integration
  - Ermöglichung von herstellerunabhängiger Standardsoftware



- Neue Anwendungsgebiete entwickeln sich ständig
- Middleware nimmt zu
- Betriebssysteme
  - werden immer komplexer
  - beeinflussen die Leistungsfähigkeit, Zuverlässigkeit und Einfachheit der Benutzung eines Systems
  - Komplexität und Funktionalität sind fast außer Kontrolle und häufig von der Verständlichkeit entfernt.
- Computersysteme gehören einer Klasse der komplexesten und kompliziertesten Geräte an, die jemals von der Menschheit erschaffen wurden.
- Verschiebung von Leistung zu Zuverlässigkeit und Sicherheit wird zunehmen.