

## Script generated by TTT

Title: Grundlagen\_Betriebssysteme (19.10.2012)

Date: Fri Oct 19 08:29:30 CEST 2012

Duration: 89:03 min

Pages: 21

**Motivation**

*Aufgabe der Informatik* ist es, Rechnerysteme zu entwickeln und diese Anwendern als leistungsfähige Hilfsmittel für Lösungen ihrer Informationsverarbeitungsprobleme zur Verfügung zu stellen.

[Anforderungen an Rechnerysteme](#)

[Struktur eines Rechnerystems](#)

Generated by Targeteam

**Motivation**

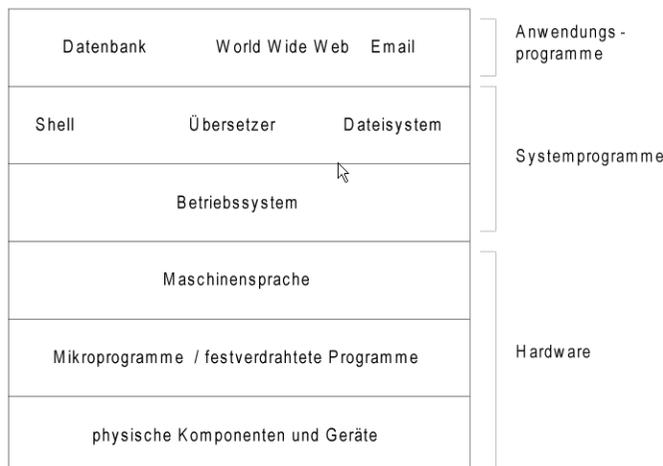
*Aufgabe der Informatik* ist es, Rechnerysteme zu entwickeln und diese Anwendern als leistungsfähige Hilfsmittel für Lösungen ihrer Informationsverarbeitungsprobleme zur Verfügung zu stellen.

[Anforderungen an Rechnerysteme](#)

[Struktur eines Rechnerystems](#)

Generated by Targeteam

**Motivation**



Darstellung von Programmen in maschinennaher Form für bestimmte Anwendungen auch heute noch unerlässlich, beispielsweise für den Übersetzerbau, eingebettete Systeme oder für systemnahe Programmierung in Teilen des Betriebssystems.

Thema der Vorlesung ist maschinennahe Programmierung, systemnahe Programmentwicklung; nebenläufige ("concurrent") Ausführung von mehreren Teilabläufen ⇒ **Nichtdeterminismus**.

Generated by Targeteam



Bereitstellung eines indirekten Zugangs zur Rechnerhardware über eine Dienstschicht. Ziel dieser Schicht ist die Realisierung einer virtuellen Maschine.

Die in den Diensten bereitgestellten Abstraktionen stellen einen Rahmen für die Organisation von Anwendungen zur Laufzeit dar, d.h. ein **Laufzeitmodell**.

Dienste des Laufzeitmodells: Systembedienung, Prozessmanagement, Prozessinteraktion, Datenhaltung, Gerätemanagement.

elementare Abstraktionen eines Laufzeitmodells.

#### Elemente des Laufzeitmodells

**Adressräume** = Abstraktion eines physischen Speichers.

**Threads** = Abstraktion eines physischen Prozessors.

**Prozessinteraktion**.

Generated by Targeteam



## Inhaltsübersicht



Im einzelnen werden in der Vorlesung die folgenden Themen behandelt:

1. Übergang von sequentiellen Systemen zu parallelen Systemen: Verhaltensbeschreibungen mit Hilfe von Spuren und Petrinetzen, Synchronisationskonzepte.
2. grundlegende Betriebssystemkonzepte: Prozessverwaltung, Prozessorzuteilung, Arbeitsspeicherverwaltung, Dateisysteme.
3. Prozesskommunikation - speicherbasiert und nachrichtenbasiert.
4. Sicherheit in Rechensystemen: Schutzmechanismen, Zugriffskontrolllisten, Capability-Listen, Kryptographie, Authentifizierung.

#### [Programmiersprachen](#)

Generated by Targeteam



## Programmiersprachen



Nutzung vor allem von Java. Daneben, wenn notwendig, Verwendung von

C Programmiersprache: zur Spezifikation von systemnahen Programmstücken.

entwickelt zwischen 1969 und 1973 von Dennis Ritchie bei Bell Labs

C und Unix sind eng miteinander verbunden

C ist eine einfache, kleine Sprache

C war und ist noch eine wichtige Sprache für systemnahe Programmierung

C bedingt jedoch einige Fehleranfälligkeiten, z.B.

dynamische Speicherbelegung und -freigabe.

Pointerarithmetik.

Generated by Targeteam



## Literaturübersicht



Literatur, die als Basis für die Vorlesung verwendet wird.

#### Begleitend zur Vorlesung

Andrew S. Tanenbaum, "Modern Operating Systems", Prentice Hall, 2008; dazu gibt es eine deutsche Übersetzung

Andrew S. Tanenbaum, "Moderne Betriebssysteme", Pearson Studium, 2009

Abraham Silberschatz, Peter Galvin, Greg Gagne, "Operating System Concepts. Operating System Concepts", Wiley & Sons, 2009

#### Weiterführende Literatur

Albrecht Achilles, "Betriebssysteme - Eine kompakte Einführung mit Linux", Springer, 2006

Uwe Baumgarten, Hans-Jürgen Siegert, "Betriebssysteme", Oldenbourg, 2006

Rüdiger Brause, "Betriebssysteme - Grundlagen und Konzepte", Springer, 2004

Eduard Glatz, "Betriebssysteme - Grundlagen, Konzepte, Systemprogrammierung", dpunkt.verlag, 2010

Jürgen Nehmer, Peter Sturm, "Systemsoftware - Grundlagen moderner Betriebssysteme", dpunkt.verlag, 2001

Randal E. Bryant, David R. O'Hallaron, "Computer Systems - A Programmer's Perspective", Pearson Education, 2010

William Stallings, "Operating Systems - Internals and Design Principles", Pearson International Edition, 2011

George Coulouris, Jean Dollimore, Tim Kindberg, "Distributed Systems - Concepts and Design", Addison-Wesley, 2012

Andrew S. Tanenbaum, Maarten van Steen, "Verteilte Systeme - Grundlagen und Paradigmen"



Abraham Silberschatz, Peter Galvin, Greg Gagne, "Operating System Concepts. Operating System Concepts", Wiley & Sons, 2009

*(Java Variante)*

## Weiterführende Literatur

*BS*

- Albrecht Achilles, "Betriebssysteme - Eine kompakte Einführung mit Linux", Springer, 2006
- Uwe Baumgarten, Hans-Jürgen Siebert, "Betriebssysteme", Oldenbourg, 2006
- Rüdiger Brause, "Betriebssysteme - Grundlagen und Konzepte", Springer, 2004
- Eduard Glatz, "Betriebssysteme - Grundlagen, Konzepte, Systemprogrammierung", dpunkt.verlag, 2010
- Jürgen Nehmer, Peter Sturm, "Systemsoftware - Grundlagen moderner Betriebssysteme", dpunkt.verlag, 2001
- Randal E. Bryant, David R. O'Hallaron, "Computer Systems - A Programmer's Perspective", Pearson Education, 2010
- William Stallings, "Operating Systems - Internals and Design Principles", Pearson International Edition, 2011
- George Coulouris, Jean Dollimore, Tim Kindberg, "Distributed Systems - Concepts and Design", Addison-Wesley, 2012
- Andrew S. Tanenbaum, Marten van Steen, "Verteilte Systeme - Grundlagen und Paradigmen", Pearson Studium, 2007
- Elliotte Rusty Harold, "Java Network Programming", O'Reilly, 2004
- Ralph Morelli, Ralph Walde, "Java, Java, Java - Object-Oriented Problem Solving", Prentice Hall, 2005
- Claudia Eckert, "IT-Sicherheit: Konzepte - Verfahren - Protokolle", Oldenbourg Verlag, 2012

Generated by Targateam



- Prof. J. Schlichter
  - Lehrstuhl für Angewandte Informatik / Kooperative Systeme, Fakultät für Informatik, TU München
  - Boltzmannstr. 3, 85748 Garching
  - Email: [schlichter@in.tum.de](mailto:schlichter@in.tum.de)
  - Tel.: 089-289 18654
  - URL: <http://www11.informatik.tu-muenchen.de/>

## Übersicht

### [Einführung](#)

### [Parallele Systeme - Modellierung, Strukturen](#)

### [Prozess- und Prozessorverwaltung](#)

### [Speicherverwaltung](#)

### [Prozesskommunikation](#)

### [Dateisysteme](#)

### [Ein-/Ausgabe](#)

### [Sicherheit in Rechenanlagen](#)

### [Entwurf von Betriebssystemen](#)

### [Zusammenfassung](#)

Generated by Targateam



Definition eines Betriebssystems nach DIN 44300:

Das Betriebssystem wird gebildet durch die Programme eines digitalen Rechenanlagens, die zusammen mit den Eigenschaften der Rechenanlage die Grundlage der möglichen Betriebsarten des digitalen Rechenanlagens bilden und insbesondere die Ausführung von Programmen steuern und überwachen.

### [Betriebssystem - Überblick](#)

### [Betriebssystem-Architektur](#)

### [Hardwarenahe Programme](#)

Generated by Targateam



Ein Betriebssystem realisiert die Schnittstelle zwischen dem Benutzer und der physischen Rechenanlage. Aus der Sicht des Benutzers entsteht durch ein Betriebssystem eine virtuelle Maschine.

### [BS-Hauptaufgaben](#)

### [Systemprogrammierung](#)

### [Hardwarekomponenten](#)

### [Betriebsarten](#)

### [Historie](#)

Generated by Targateam





Ein **Betriebssystem** (engl. operating system) erfüllt folgende Hauptaufgaben:

Veredeln der Hardware (Virtualisierung).

Steuerung und Kontrolle der Programmausführung.

Interprozesskommunikation.

Verwaltung der Ressourcen (Speicher, CPU, Platten, Netz etc.) ⇒ Betriebssystem kann als Ressourcenverwalter gesehen werden.

#### Ressourcenklassen

Anbieten von Diensten in Form von Schnittstellen, so dass die Ressourcen genutzt werden können ⇒ Hardwareunabhängige Programmierschnittstelle, z.B. geräteunabhängige Ein-/Ausgabefunktionen.

Sicherheitsmechanismen.

#### Arbeitsmodi des Betriebssystems

##### Struktureller Aufbau

##### Ein anderer Blickwinkel

Generated by Targeteam

10



Ein Rechensystem kann als strukturierte Sammlung von Ressourcenklassen betrachtet werden, wobei jede Klasse durch Dienste des Betriebssystems kontrolliert wird.

	Zentral Ressourcen	Periphere Ressourcen
Aktive Ressourcen	Prozessoren (CPUs)	Kommunikationseinheiten wie Endgeräte ( Tastatur, Drucker, Monitor, Maus) und Netzwerk (lokal, entfernt)
Passive Ressourcen	Arbeitsspeicher	Speichereinheiten wie Festplatten, CD, DVD

Generated by Targeteam

10



## BS-Hauptaufgaben



Ein **Betriebssystem** (engl. operating system) erfüllt folgende Hauptaufgaben:

Veredeln der Hardware (Virtualisierung).

Steuerung und Kontrolle der Programmausführung.

Interprozesskommunikation.

Verwaltung der Ressourcen (Speicher, CPU, Platten, Netz etc.) ⇒ Betriebssystem kann als Ressourcenverwalter gesehen werden.

#### Ressourcenklassen

Anbieten von Diensten in Form von Schnittstellen, so dass die Ressourcen genutzt werden können ⇒ Hardwareunabhängige Programmierschnittstelle, z.B. geräteunabhängige Ein-/Ausgabefunktionen.

Sicherheitsmechanismen.

#### Arbeitsmodi des Betriebssystems

##### Struktureller Aufbau

##### Ein anderer Blickwinkel

Generated by Targeteam

10



## Arbeitsmodi des Betriebssystems

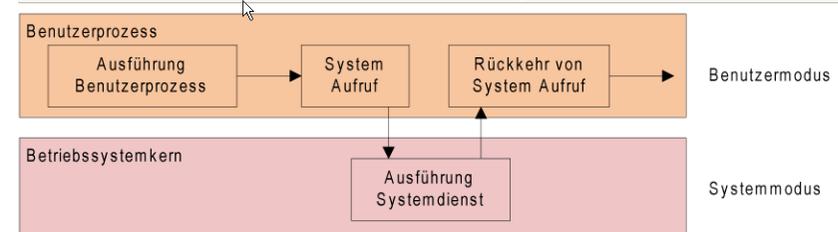


Operationen des Betriebssystems und der Hardware müssen vor Programmierfehlern in Anwendungsprogrammen geschützt werden ⇒ Einführung eines Privilegiensystems.

**Benutzermodus (user mode):** Ausführung von Benutzerprogrammen, kein direkter Hardware-Zugriff, keine privilegierten Befehl, nur virtuelle Adressen.

**Systemmodus (kernel mode):** Ausführungsmodus der Dienste des BS-Kerns, privilegierte Befehle erlaubt.

Benutzermodus	Systemmodus
begrenzte Auswahl von Maschinenbefehlen	alle ausführbaren Maschinenbefehle
Hardwarezugriff nur über BS	Vollzugriff auf Hardware
kein bzw. nur lesender Zugriff auf Systemcode oder Daten	exklusiver Zugriff auf Systemcode und Daten



Generated by Targeteam

10

Ein **Betriebssystem** (engl. operating system) erfüllt folgende Hauptaufgaben:

Veredeln der Hardware (Virtualisierung).

Steuerung und Kontrolle der Programmausführung.

Interprozesskommunikation.

Verwaltung der Ressourcen (Speicher, CPU, Platten, Netz etc.) ⇒ Betriebssystem kann als Ressourcenverwalter gesehen werden.

### Ressourcenklassen

Anbieten von Diensten in Form von Schnittstellen, so dass die Ressourcen genutzt werden können ⇒ Hardwareunabhängige Programmierschnittstelle, z.B. geräteunabhängige Ein-/Ausgabefunktionen.

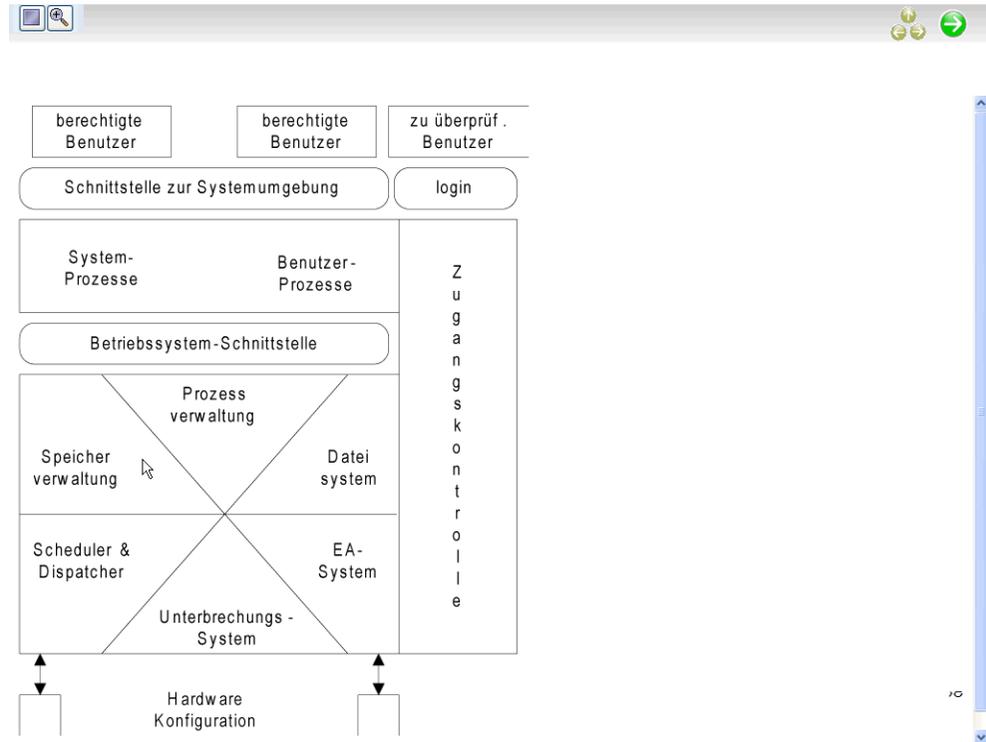
Sicherheitsmechanismen.

### Arbeitsmodi des Betriebssystems

#### Struktureller Aufbau

#### Ein anderer Blickwinkel

Generated by Targteam



NY Times, Sept, 3, 1997: A decade ago, an "operating system" was just the basic piece of software that ruled the machine and allowed it to manipulate files, converse with any peripherals, and start up programs. That was when a computer was just a nerd toy, not the foundation for the most vital part of our economy. But today, an "operating system" is much more a vote over who gets to be the richest men in the world. Windows means Microsoft, Java means Sun, while MacOs means That Steve Jobs won't go broke saving Apple. Linux means no one gets rich because the OS is free, thanks to the help of many volunteers.

Generated by Targteam

Die Programmierung eines Betriebssystems gehört zu dem Bereich der Systemprogrammierung.

### Definition

Die **Systemprogrammierung** befasst sich mit der Darstellung, der Realisierung, den Eigenschaften und der Konstruktion derjenigen Algorithmen für ein Rechensystem, die die Bearbeitung und Ausführung von Benutzerprogrammen unter vorgegebenen Qualitäts Gesichtspunkten organisieren, d.h. **steuern** und **kontrollieren**, und zum Teil selbst durchführen.

direkte Nutzung der generischen Systemprogrammierschnittstelle des BS.

meist in Programmiersprache C.

Qualitätskriterien können z.B. sein:

**Zuverlässigkeit** der durchgeführten Berechnung (Behandlung von Systemcrashes, Netzausfällen, fehlerhafter Nachrichtenübermittlung etc.).

**Effizienz und Performanz** einerseits systemglobal, d.h. es wird versucht, das System optimal auszulasten, andererseits Auftrags-lokal, z.B. es wird versucht, zu garantieren, dass eine Auftragsbearbeitung eine festgelegte Zeitdauer nicht überschreitet.

Einhaltung von **Realzeitanforderungen**: zeitkritische Aufträge besitzen z.B. eine Deadline bis zu der sie ausgeführt sein müssen.

Durchsetzung von **Sicherheitsanforderungen**: Schutz der Daten und Informationen vor unberechtigten Zugriffen und Einsichtnahme.

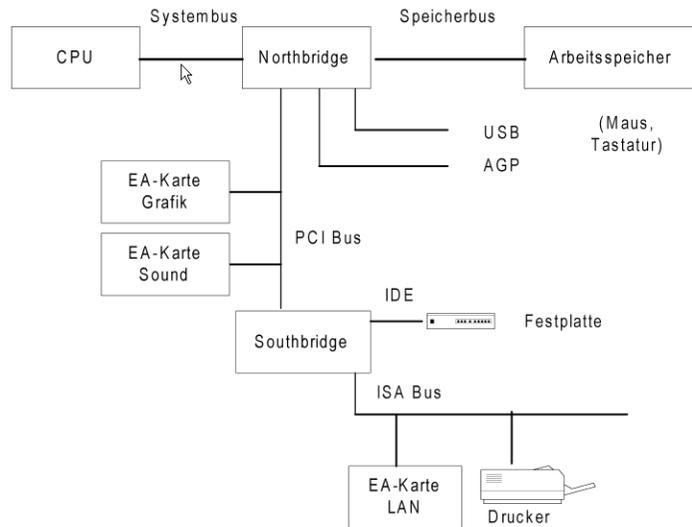
**Benutzerfreundlichkeit**: bequeme Formulierungsmöglichkeit von Benutzeraufträgen.

Generated by Targteam



Betriebssystem ist eng mit Hardware des Rechnerystems verknüpft.

Deshalb an dieser Stelle einen kurzen Überblick über den Aufbau eines Rechnerystems.



PCI = Peripheral Component Interconnect

Ein Betriebssystem realisiert die Schnittstelle zwischen dem Benutzer und der physischen Rechenanlage. Aus der Sicht des Benutzers entsteht durch ein Betriebssystem eine virtuelle Maschine.

[BS-Hauptaufgaben](#)

[Systemprogrammierung](#)

[Hardwarekomponenten](#)

[Betriebsarten](#)

[Historie](#)