



Script generated by TTT

Title: Grundlagen_Betriebssysteme (14.10.2015)

Date: Wed Oct 14 13:15:34 CEST 2015

Duration: 44:38 min

Pages: 18

Grundlagen: Betriebssysteme und Systemsoftware (GBS)

Johann Schlichter

Lehrstuhl 11

Angewandte Informatik / Kooperative Systeme

Oktober 2015



Eigene Vorstellung



- 2003 – 2006 Dekan der Fakultät für Informatik
- Stellvertretender Vorsitzender im Prüfungsausschuss (Bachelor)
- **Email:** schlichter@in.tum.de
- Sekretariat: Frau Lissner, Tel 289-18654
- Sprechstunde:
- Web-Site: <http://www11.in.tum.de/lehre>
 - PDF-Skript: Online Nutzung und für Druck
 - Geplant: Aufzeichnung der Vorlesung mit Hilfe von TTT, nur Audio und Annotationen
- Vorlesungszeiten:
 - Montags: 13:45 – 15:15 MW0001
 - Mittwochs: 13:15 – 14:00 FMI HS 1
- Übungsleitung: Florian Schulze



Eigene Vorstellung



- Studium und Promotion an der TU München
- 1982 – 1985 Siemens Research, Princeton, NJ, USA
- 1985 – 1989 Xerox Research, Rochester, NY, USA
- 1990 – 1991 Softlab, München
- Seit 1991 an der TUM, Lehrstuhl für Angewandte Informatik / Kooperative Systeme

Lehre aus dem Bereich Technische/Angewandte Informatik



Forschungsinteressen am Lehrstuhl



- mobile verteilte kooperative Anwendungen, Social Software, kontextsensitive Unterstützungssysteme für Teams, Groupware, Communityware, Recommendersysteme



Ziele der Veranstaltung



- Grundlegendes Verständnis der Betriebssysteme und der Systemsoftware
- Verstehen der Problembereiche
- Konzepte mit langer Wissenshalbwertszeit
- Aktuelle Beispiele zur Anschauung und Übung
- Jeder wird in die Lage versetzt, neue Betriebssysteme/Systemsoftware schnell zu verstehen, einzuordnen und einsetzen zu können
- Vertiefung in den Übungen mit Aufgaben
- Erlangen von aktivem Wissen
- Basierend auf Grundkenntnissen des ersten Studienjahres (Info1, TGI, SE, GAD)

4



betrieb_Intro.pdf - Adobe Acrobat Professional

File Edit View Document Comments Forms Tools Advanced Window Help

Sticky Note Text Edits No current selection

2 / 3 65,2%

Blätter und Hausaufgaben

- Übung richtet sich inhaltlich und zeitlich an Vorlesung aus und vertieft die dortigen Konzepte; fokussiert auf Implementierungen in C
- Hausaufgaben sind beim Tutor abzugeben (**2er-Teams**), Abgabeform legt dieser fest. Musterlösungen werden zentral veröffentlicht.
- Bewertung des Blattes im Gesamten, nicht pro Aufgabe
- zwischen 0 Punkten (keine sinnvolle Bearbeitung) und 3 (sehr gute Bearbeitung)
- Hausaufgaben und Übungsteilnahme sind freiwillig (Notenbonus)

Start

betrieb_Intro.pdf - Adobe Acrobat Professional

File Edit View Document Comments Forms Tools Advanced Window Help

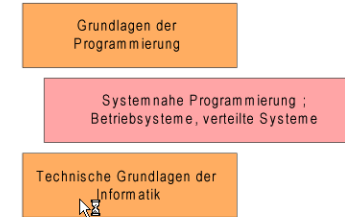
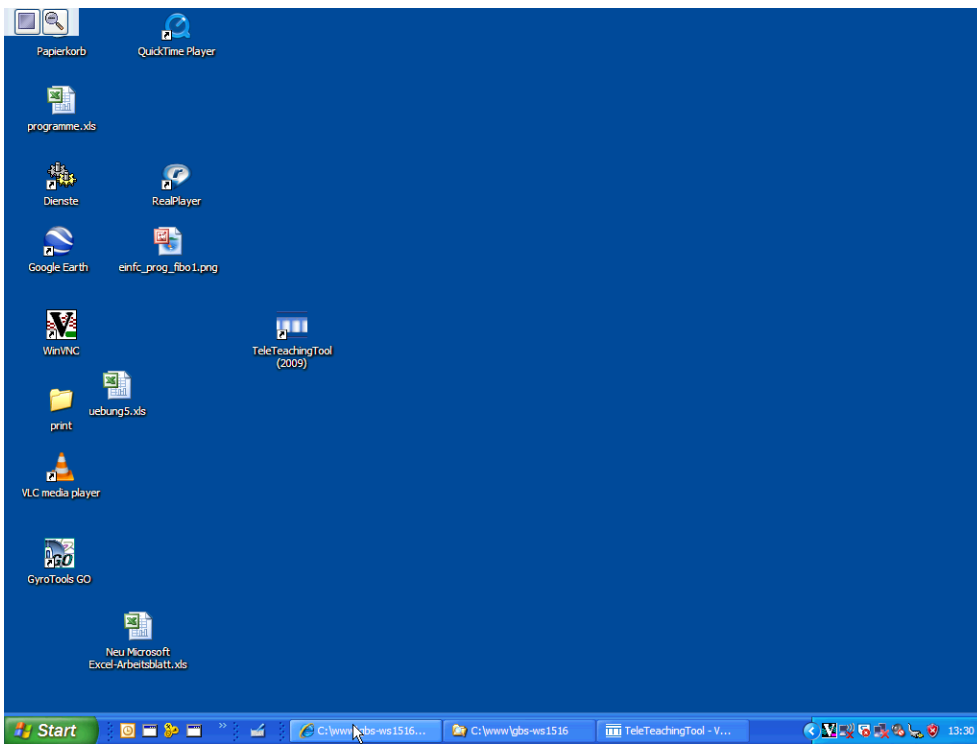
Sticky Note Text Edits No current selection

3 / 3 65,2%

Klausur und Notenbonus

- Nur Finalklausur, keine Midterm
- Termin: TBA
- Notenbonus von 0,3 auf bestandene Klausur
 - ab 22 der 33 möglichen HA-Punkte
 - höchstens ein Blatt mit 0 Punkten
 - **gilt nur für Hauptklausur (nicht für Wiederholungsklausur)**

Start



[Ziele der Vorlesung](#)

[Motivation](#)

[Themen der Vorlesung](#)

[Literaturübersicht](#)

Generated by Targeteam

Diese Vorlesung beschäftigt sich mit den technischen Aspekten von Rechner-Systemen und der Informationsverarbeitung, nämlich der **systemnahen Programmierung** im Zusammenhang mit Betriebssystemen. Dabei werden vor allem nicht verteilte Systeme betrachtet.

Einführung in Betriebssysteme und Systemsoftware.

Grundlagenprobleme paralleler Systeme: Modellierung des Systemverhaltens, Synchronisation, Verklemmungen.

Konzepte und Verfahren von Betriebssystemen: Arbeitsspeicherverwaltung, die Prozessverwaltung, die Prozessorzuteilung und Kontrolle der Nebenläufigkeit.

Prozesskommunikation - speicherbasiert und nachrichtenbasiert.

Ein-/Ausgabe - Anbindung von peripheren Geräten.

Sicherheit in Rechner-Systemen.

Generated by Targeteam

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

[Anforderungen an Rechner-Systeme](#)

[Struktur eines Rechner-Systems](#)

Generated by Targeteam

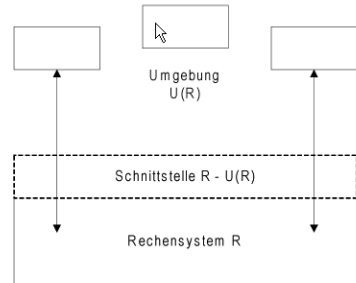


Ein Rechensystem **R** ist ein offenes System sagt zweierlei aus:

R ist als System eine durch Zusammenfassung gebildete, abgegrenzte Einheit, wo zwischen innen (**R**) und außen (**U(R)**) unterschieden wird.

R hat eine (offene) Schnittstelle, mit der Einwirkungen von **U(R)** auf **R** und Einwirkungen von **R** auf **U(R)** möglich sind.

Schnittstelle eines Rechensystems



Sichten eines Rechensystems

Generated by Targeteam



Es existieren verschiedene Sichten auf das Rechensystems **R**

Außensicht vs. Innensicht.

Black-box Sicht vs. White-box Sicht.

Sichten sind methodische Hilfsmittel für Systemanalysen:

Komponenten haben Eigenschaften, die denen von Systemen entsprechen.

Verbindungen zwischen Komponenten beschreiben Abhängigkeiten zwischen Komponenten.

Aufteilungen liefern verfeinerte White-box Sichten.

Zusammenfassungen und Abgrenzungen liefern Einheiten, für die Black-box Sichten möglich sind.

⇒ Rekursion bei der Entwicklung und Analyse von Rechensystemen.

Rekursive/iterative Vorgehensweise bei der Analyse und Entwicklung großer Rechensysteme.

Generated by Targeteam



Rechensysteme sind

offene, **dynamische**, **technische** Systeme

mit Fähigkeiten zur **Speicherung** und zur

Verarbeitung von Information, die für Anwendungen und Anwender nutzbar zur Verfügung gestellt werden sollen.

Offenes System

Dynamisches System

Eigenschaften des Rechensystems **R** ändern sich mit der Zeit

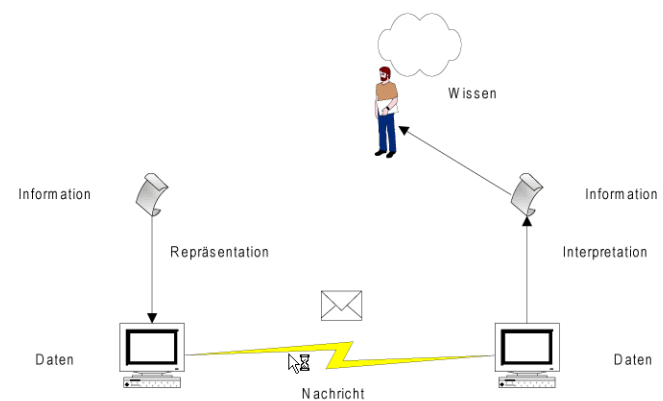
⇒ Beschreibung des Verhaltens von **R**.

Technisches System

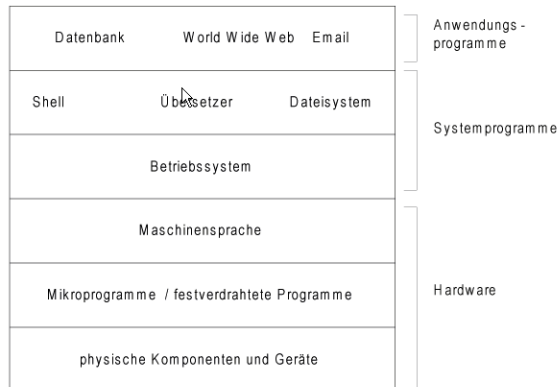
Das Rechensystem ist mit hardware- und softwaretechnischen Mitteln realisiert.

Informationsspeicherung und -verarbeitung

Generated by Targeteam



Generated by Targeteam



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 systemnahe Programmentwicklung; darunter maschinennahe Programmierung und nebenläufige ("concurrent") Ausführung von mehreren Teilabläufen ⇒ **Nichtdeterminismus**.

Generated by Targeteam

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

Anforderungen an Rechensysteme

Struktur eines Rechensystems

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