

Betriebssysteme (BS)

http://www4.informatik.uni-erlangen.de/Lehre/WS07/V_BS/

Daniel Lohmann
Jürgen Kleinöder

Lehrstuhl für Informatik 4
Verteilte Systeme und Betriebssysteme
Friedrich-Alexander Universität
Erlangen-Nürnberg

lohmann@informatik.uni-erlangen.de
kleinoder@informatik.uni-erlangen.de



1

Lernziele

- das Wissen über die Funktionsweise von Betriebssystemen vertiefen
 - Ausgangspunkt: Softwaresysteme I
 - Schwerpunkte: Nebenläufigkeit und Synchronisation
- Entwicklung eines Betriebssystems *von der Pike auf*
 - OO-StuBS: Praktische Erfahrungen im Betriebssystembau sammeln
- PC-Technologie besser verstehen



BS © 2006, 2007 Wolfgang Schröder-Preikschat, Olaf Spinczyk, Jürgen Kleinöder, Daniel Lohmann

2

Einordnung (1)

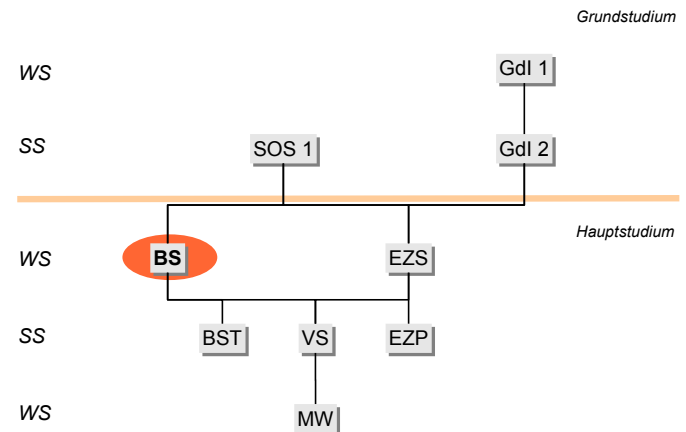
- Informatik, ab 5. Semester (s.u.)
- Wirtschaftsinformatik, ab 7. Semester (Wahl)
- Computational Engineering, ab 7. Semester (Wahl)
- Wahlpflicht für „Verteilte Systeme und Betriebssysteme“ ...
 - als Prüfungsfach (8 SWS) oder Schwerpunktfach (12 SWS)
 - als benoteter oder unbenoteter Schein
- sehr empfohlen bei vorgesehener Mitarbeit an F&E Arbeiten des Lehrstuhls



BS © 2006, 2007 Wolfgang Schröder-Preikschat, Olaf Spinczyk, Jürgen Kleinöder, Daniel Lohmann

3

Einordnung (2)



BS © 2006, 2007 Wolfgang Schröder-Preikschat, Olaf Spinczyk, Jürgen Kleinöder, Daniel Lohmann

4

Voraussetzungen

- Rechnerorganisation, **Softwaresysteme I** [1]
- C/C++, Assembler (x86)
- Spaß an hardwarenaher Programmierung
 - „Furchtlosigkeit“ vor nur schwer erkund- und fassbaren Sachverhalten
- ein gewisses Maß an **Durchhaltevermögen**



Organisation

- **integrierte Lehrveranstaltung** 4 SWS
(2 x 1,5 Std. wöchentlich)
 - **VL** Vorstellung und detaillierte Behandlung des Lehrstoffs
 - **Ü** Vertiefung sowie Besprechung und Abnahme der Übungsaufgaben (Abnahmen alle 14 Tage)

Anmeldung zu den Übungen über „waffel“ (siehe Web-Seite)
- **Rechnerübung** 0 SWS
(1,5 Std. wöchentlich)
 - betreutes Bearbeiten der Übungsaufgaben am Rechner
- **Vor-/Nacharbeit** 0 SWS
(N Std. wöchentlich, $0 < N < 163,5$)



Schein, Prüfung

- **Schein**
 - Ausstellung ... bei erfolgreicher Bearbeitung aller Aufgaben
 - Rücksprache ... bei nicht-erfolgreicher Bearbeitung einer Aufgabe
- **benoteter Schein**
 - Voraussetzung: Schein (s.o.)
 - abschließendes Gespräch („Scheinprüfung“) über Vorlesungs- und Übungsstoff
- **Prüfung**
 - für Fach „Verteilte Systeme und Betriebssysteme“
 - für Informatik oder Winf in Kombination mit einer anderen LV des Lehrstuhls (BST, VS, MW, ES, SysSec)
 - für IuK oder CE als Einzelprüfung
 - Inhalt ist der Vorlesungs- und Übungsstoff (Übungsaufgaben)



Dozenten, Übungsleiter

- **Jürgen Kleinöder**
 - <http://www4.informatik.uni-erlangen.de/~jklein>
- **Daniel Lohmann**
 - <http://www4.informatik.uni-erlangen.de/~lohmann>
- **Wanja Hofer**
 - <http://www4.informatik.uni-erlangen.de/~hofer>
- **Julio Sincero**
 - <http://www4.informatik.uni-erlangen.de/~sincero>



Empfohlene Literatur

- [1] W. Schröder-Preikschat. *Softwaresysteme I*. <http://www4.informatik.uni-erlangen.de/>, 2005. Vorlesungsskript.
- [2] R. G. Herrtwich and G. Hommel. *Kooperation und Konkurrenz - Nebenläufige, verteilte und echtzeitabhängige Programmsysteme*. Springer-Verlag, 1989. ISBN 3-540-51701-4.
- [3] W. Schröder-Preikschat. *The Logical Design of Parallel Operating Systems*. Prentice Hall International, 1994. ISBN 0-13-183369-3.
- [4] A. Silberschatz and P. B. Galvin. *Operating System Concepts*. Addison-Wesley, 1994. ISBN 0-201-59292-4.
- [5] A. S. Tanenbaum. *Structured Computer Organization*. Prentice Hall, 1990.
- [6] H.-P. Messmer, K. Dembowski. *PC-Hardwarebuch*. Addison-Wesley, 2003. ISBN 3-8273-2014-3.
- [7] Intel Corporation. *Intel Architecture Software Developer's Manual*. <http://www.intel.com/>

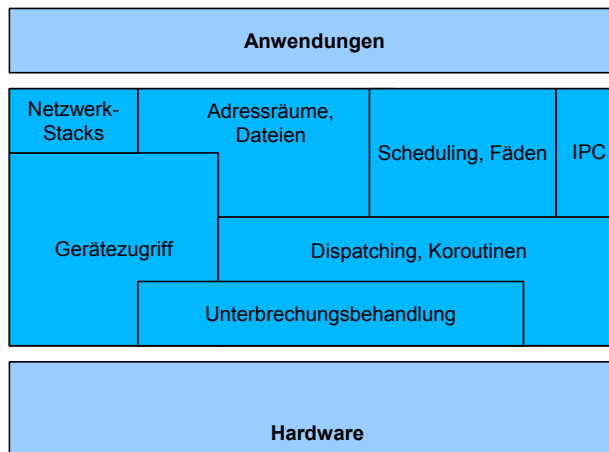


Verschiedene Sichten auf ein BS

- **Abstraktion**
 - Betriebssystem = erweiterte Maschine
 - Systemaufrufe (= Befehle der erweiterten Maschine) abstrahieren von der Komplexität von Gerätezugriffen, Dateiverwaltung, CPU-Verwaltung (Prozesse)
- **Virtualisierung**
 - ermöglicht transparente, gemeinsame Nutzung von Ressourcen
 - Zeitscheiben P virtuelle Prozessoren
 - virtueller Speicher P privater Speicher für Prozesse
 - Virtualisierung kompletter Hardwaresysteme (VM 370, XEN)
- **Ressourcenverwaltung**
 - Regelt die gemeinsame Nutzung von Ressourcen
 - Speicherverwaltung (Paging-Strategie), Scheduling, ...



Aufbau eines Betriebssystems



Zentrale Aufgabe: Verwaltung der CPU

- Alle Betriebssystemkomponenten benötigen letztlich die/eine CPU
- Konkurrenz unterschiedlicher Kontrollflüsse
 - ↳ inhärente Nebenläufigkeit
 - Interrupt-Bearbeitung
 - Interrupt-Handler / Top-Half / Prolog
 - Bottom-Halves / Tasklets / Epiloge
 - Threads / Prozesse
 - auf Anwendungsebene
 - im Betriebssystem



Betriebssysteme (BS)

alias *Betriebssystembau (BSB)*

Überblick



1

BS-Entwicklung (oft ein harter Kampf)

- erste Schritte
wie bringt man sein System auf die Zielhardware?
 - Übersetzung
 - Bootvorgang
- Testen und *Debugging*
was tun, wenn das System nicht reagiert?
 - „printf“ *debugging*
 - Simulatoren
 - *debugger*
 - *remote debugging*
 - Hardwareunterstützung



BSB © 2006 Wolfgang Schröder-Preikschat, Olaf Spinczyk

2

Programmunterbrechungen

- im Prinzip
 - Vektortabellen
 - *edge vs. level-triggered*
 - *spurious interrupts*
 - geschachtelte Unterbrechungen
- beim PC
 - PIC und APIC
 - Unterbrechungen bei Multiprozessorsystemen
 - IDT



BSB © 2006 Wolfgang Schröder-Preikschat, Olaf Spinczyk

3

Unterbrechungssynchronisation

- Zusammenspiel zwischen Unterbrechungsbehandlung und „normalem“ Kontrollfluss
- Hardwaremechanismen zur „harten Synchronisation“
- Softwaremechanismen zur „weichen Synchronisation“
 - Pro-/Epilogmodell und Varianten
 - Unterbrechungstransparenz



BSB © 2006 Wolfgang Schröder-Preikschat, Olaf Spinczyk

4

das Intel CPU Programmiermodell

- die Entwicklung der x86 CPU Familie
- Relikte
 - 8086 *Real Mode*
 - A20 *Gate*
- das ringbasierte Schutzkonzept im *Protected Mode*
- das *Task*-Modell



Programmfäden

- Realisierung von Programmfäden beim x86
 - Basis: Koroutinen
 - Implementierung des Kontextwechsels
- unterschiedliche Fadenmodelle
 - leicht vs. schwer vs. federgewichtig vs. ...
 - Umsetzung in einer Systemfamilie



Ablaufplanung

- Wiederholung und Vertiefung von Softwaresysteme I
 - Grundprinzipien und Klassifikation
- neue Strategien
- Tricks aus der Praxis
- spezielle Probleme
 - z.B. beim Zusammenspiel zwischen Ablaufplanung und Unterbrechungssynchronisation



Betriebssystemarchitektur

- verschiedene Klassen von Systemen entstehen durch unterschiedliche Komposition der BS-Mechanismen
- Mikrokern-Systeme, Monolithen, Exokernel, ...
 - L4, Solaris, Linux, Windows XP
- Hypervisor
 - XEN



Fadensynchronisation

- blockierend vs. nicht-blockierend
- Multiprozessor-Fadensynchronisation
- die ultimative Synchronisationsprimitive
 - Semaphore?
- spezielle Probleme
 - Wechselwirkung zwischen Synchronisation und Ablaufplanung
 - Wiederholung und Vertiefung von Verklemmungen



Interprozesskommunikation (IPC)

- Zusammenhang zwischen IPC und Synchronisation
 - konkrete Beispielsysteme
- Dualität nachrichten-basierter und prozeduraler Systeme
 - Lauer & Needham



PC Bussysteme

- Architektur und Programmierung
- Lokale Busse
 - PCI (Express)
 - AGP
 - InfiniBand
 - HyperTransport
- I/O Busse
 - USB, FireWire
 - SCSI, SATA
 - ...



Geräteprogrammierung

- typische PC Geräte und Probleme
 - Maus
 - Diskettenlaufwerk
 - hardware-beschleunigte Graphikkarten
- I/O System
- Treibermodelle





Viel Spaß!

