

Betriebssysteme (BS)

VL 1 – Einführung

Daniel Lohmann

Lehrstuhl für Informatik 4
Verteilte Systeme und Betriebssysteme

Friedrich-Alexander-Universität
Erlangen Nürnberg

WS 10 – 20. Oktober 2010

http://www4.informatik.uni-erlangen.de/Lehre/WS10/V_BS



Lernziele

- **Vertiefen** des Wissens über die interne Funktionsweise von Betriebssystemen
 - Ausgangspunkt: Systemprogrammierung
 - Schwerpunkt: Nebenläufigkeit und Synchronisation
- **Entwickeln** eines Betriebssystems *von der Pike auf*
 - OOSTuBS / MPSTuBS (neu!) Lehrbetriebssysteme
 - **Praktische** Erfahrungen im Betriebssystembau machen
- **Verstehen** der technologischen Hardware-Grundlagen
 - PC-Technologie verstehen und einschätzen können
 - Schwerpunkt: Intel x86 / IA-32



Vorwort

Die Lehrveranstaltung ist grundsätzlich für alle Studiengänge offen. Sie verlangt allerdings gewisse Vorkenntnisse. Diese müssen nicht durch Teilnahme an den Lehrveranstaltungen von 14 erworben worden sein.



Voraussetzungen

- Rechnerarchitektur, **Systemprogrammierung**
- C / **C++**, Assembler (x86)
- Ein gewisses Maß an **Durchhaltevermögen**
- Freude an systemnaher und **hardwarenaher Programmierung**

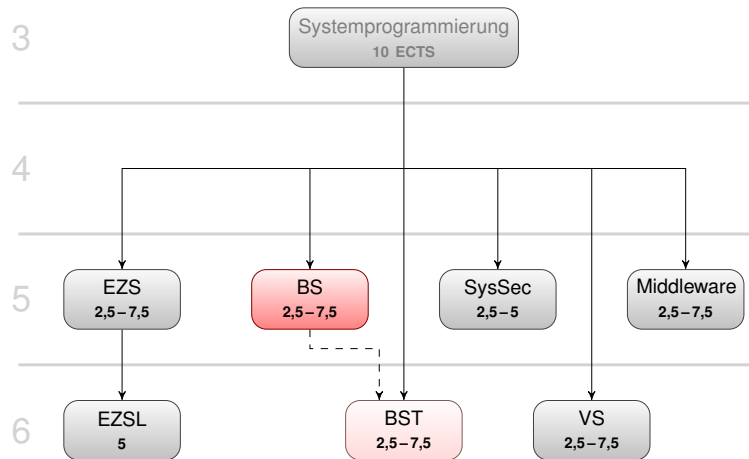
Wir arbeiten auf der “nackten Maschine” (*bare metal*)!

Die meisten sind überrascht, wie viel Spaß das macht :-)



Einpassung in den Musterstudienplan

(Bachelor)



dl Betriebssysteme (VL 1 | WS 10)

Lehrveranstaltungs-konzept – Einordnung

1-5

Verwendbarkeit, Scheinerwerb und Modulnote

- **Wahlpflichtmodul** (Bachelor/Master) der Vertiefungsrichtung **Verteilte Systeme und Betriebssysteme**
 - eigenständig (nur BS) VL + Ü oder VL + EÜ
 - mit weiteren Veranstaltungen VL oder VL + Ü oder VL + EÜ
- Studien- und Prüfungsleistungen
 - Bachelor benoteter Schein
 - Master Prüfungsleistung
 - erworben durch
 - erfolgreiche Teilnahme an den Übungen
 - erfolgreiche Bearbeitung aller Übungsaufgaben
 - 30 min. mündliche Prüfung
- Berechnung der Modulnote
 - Note der mündlichen Prüfung + "Übungsbonus" in Zweifelsfällen



dl Betriebssysteme (VL 1 | WS 10)

Lehrveranstaltungs-konzept – Organisation

1-7

Veranstaltungsbestandteile

(2,5 – 7,5 ECTS Punkte)

VL – Vorlesung 2,5

Vorstellung und detaillierte Behandlung des Lehrstoffs

+

Ü – Übung 2,5

- Übung **OOSTuBS**
- 6 – 7 Übungsaufgaben
- Abnahme alle 14 Tage

oder

EÜ – Erweiterte Übung 5

- Übung **MPStuBS**
- erweiterte Aufgaben
- Rechnerübung "Pflicht"

+

RÜ – Rechnerübung 0

- **Betreutes** Arbeiten am Rechner
- Hilfe zu OOSTuBS und **MPStuBS**



dl Betriebssysteme (VL 1 | WS 10)

Lehrveranstaltungs-konzept – Organisation

1-6

Übungsbetrieb

Übung (0.031, Abgaben in 01.155-N)

- Zwei Termine zur Auswahl
 - Montag, 12:15 – 13:45 oder Dienstag, 10:15 – 11:45
- Übungsaufgaben sind bevorzugt in Gruppen zu bearbeiten
 - 2er-Gruppe ~ Aufgabe 7 ist **freiwillig**
 - 3er-Gruppe ~ Aufgabe 7 ist **verpflichtend**
- Anmeldung über **WAFFEL** (URL siehe Webseite)
 - Freischaltung erfolgt nach der Vorlesung, heute im Tagesverlauf

Rechnerübung (01.155-N)

- Zwei Termine zur Auswahl
 - Dienstag, 12:15 – 13:45 oder Mittwoch, 12:15 – 13:45
- Betreuer können auch jederzeit direkt angesprochen werden



dl Betriebssysteme (VL 1 | WS 10)

Lehrveranstaltungs-konzept – Organisation

1-8

Terminübersicht Wintersemester 2010/2011

KW	Mo 12-14	Di 10-12	Di 12-14	Mi 8,5-10	Mi 12-14	Raum
18.10.				VL ₁		H4
25.10.	Ü ₁	Ü ₁	RÜ	VL ₂	RÜ	0.031
01.11.				VL ₃		01.155-N
08.11.	Ü ₂	Ü ₂	RÜ	VL ₄	RÜ	
15.11.	A ₁	A ₁	RÜ	VL ₅	RÜ	01.155-N
22.11.	Ü ₃	Ü ₃	RÜ	VL ₆	RÜ	
29.11.	A ₂	A ₂	RÜ	VL ₇	RÜ	
06.12.	Ü ₄	Ü ₄	RÜ	VL ₈	RÜ	
13.12.	A ₃	A ₃	RÜ	VL ₉	RÜ	
20.12.	Ü ₅	Ü ₅	RÜ			
10.01.	A ₄	A ₄	RÜ	VL ₁₀	RÜ	
17.01.	Ü ₆	Ü ₆	RÜ	VL ₁₁	RÜ	
24.01.	A ₅	A ₅	RÜ	VL ₁₂	RÜ	
31.01.	Ü ₇	Ü ₇	RÜ	VL ₁₃	RÜ	
07.02.	A ₆	A ₆		VL ₁₄		



Dozenten und Übungsleiter

Vorlesung



Daniel Lohmann



Wolfgang Schröder-Preikschat

Übung



Benjamin Oechslein



Isabella Thomm

Rechnerübung



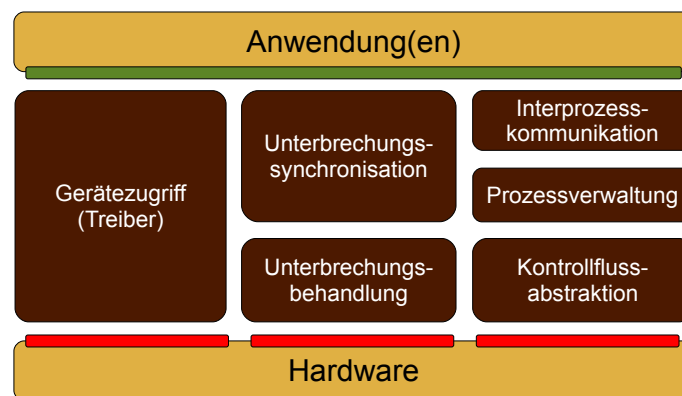
Isabella Thomm



Dirk Wischermann

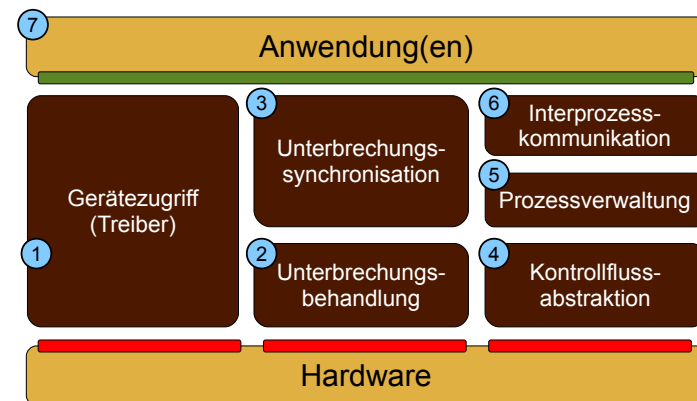


Aufbau eines Betriebssystems



Themenübersicht Übung

Am Beispiel von: OOSTuBS, MPStuBS

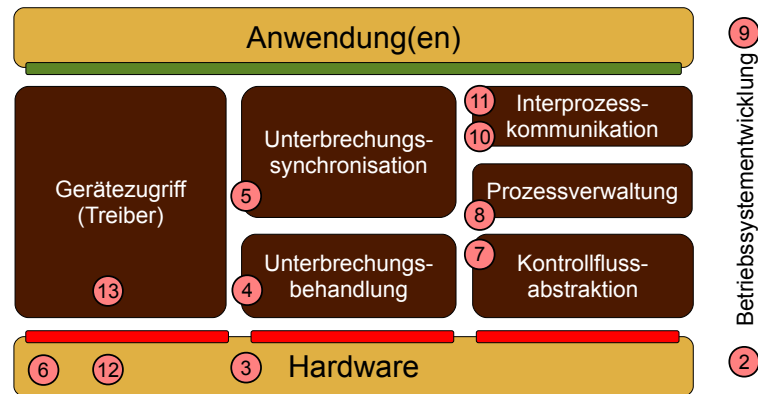


Betriebssystementwicklung



Themenübersicht Vorlesung

Am Beispiel von: x86, MC68k, TriCore; Windows, Linux



Verzahnung von Vorlesung und Übungsaufgaben

KW

18.10.	VL ₁ Einführung	
25.10.	VL ₂ BS-Entwicklung	→ Ü ₁ Ein-/Ausgabe
01.11.	VL ₃ IRQs (Hardware)	
08.11.	VL ₄ IRQs (Software)	→ Ü ₂ IRQ-Behandlung
15.11.	VL ₅ IRQs (Synchronisation)	
22.11.	VL ₆ Intel IA-32	→ Ü ₃ IRQ-Synchronisation
29.11.	VL ₇ Koroutinen und Fäden	
06.12.	VL ₈ Scheduling	→ Ü ₄ Fadenumschaltung
13.12.	VL ₉ BS-Architekturen	→ Ü ₅ Zeitscheiben-Scheduling
10.01.	VL ₁₀ Fadensynchronisation	
17.01.	VL ₁₁ PC Bussysteme	→ Ü ₆ Fadensynchronisation
24.01.	VL ₁₂ Gerätetreiber	
31.01.	VL ₁₃ IPC	→ Ü ₇ "Eine Anwendung" (opt.)
07.02.	VL ₁₄ Ausblick	



VL₂ Betriebssystementwicklung (oft ein harter Kampf!)

■ Erste Schritte

Wie bringt man sein System auf die Zielhardware?

- Übersetzen und Linken für "nakte Hardware"
- Bootvorgang

■ Testen und Fehlersuche

Was tun, wenn das System nicht reagiert?

- "printf"-Debugging
- Simulatoren
- Debugger
- Remote debugging
- Hardwareunterstützung



VL_{3–4} Programmunterbrechungen

■ im Prinzip

- Unterbrechungen, *Traps* und Ausnahmen
- Vektortabellen
- geschachtelte Unterbrechungen
- *spurious interrupts*

■ beim PC

- CPU und APIC
- Unterbrechungen in Multiprozessorsystemen

■ Behandlung im Betriebssystem

- Kopplungsfunktion
- Zustandssicherung



VL₅ Unterbrechungssynchronisation

- Zusammenspiel zwischen Unterbrechungsbehandlung und “normalem” Kontrollfluss
 - Ursache und Problem
 - Kontrollflussebenenmodell
- Hardware-Mechanismen zur “harten Synchronisation”
 - `cli` und `sti`
 - Unterbrechungsebenen
- Software-Mechanismen zur “weichen Synchronisation”
 - Pro-/Epilogmodell und Varianten
 - Unterbrechungstransparente Algorithmen



VL₇ Programmfäden

- Realisierung von Programmfäden
 - beim MC68k, Infineon TriCore, Intel x86
 - Fortsetzungen und Koroutinen als Basis
 - Implementierung des Kontextwechsels
- Fadenmodelle
 - leicht vs. schwer vs. federgewichtig vs. . . .
 - Umsetzung in einer Systemfamilie



VL₆ Das Intel IA-32 Programmiermodell

- Die Entwicklung der x86 CPU-Familie
 - vom 8086 bis zum Core i7
- Relikte und Eigenarten (*quirks*)
 - *Real Mode*
 - *A20 Gate*
- Neuerungen des *Protected Mode*
 - Ringe und Schutzmodell
 - *Task*-Modell
- Hardwarevirtualisierung



VL₈ Ablaufplanung

- Kurze Wiederholung und Vertiefung
 - Grundprinzipien
 - Klassifikation
 - neue Strategien
- Beispiele aus der Praxis
 - Windows
 - Linux
 - Scheduling auf Multiprozessor-Systemen
- Herausforderungen beim Betriebssystembau
 - Zusammenspiel Ablaufplanung $\Rightarrow$ Unterbrechungssynchronisation



VL₉ Architekturen

- Wie organisiert man ein Betriebssystem: Architekturmodelle
 - Bibliotheken
 - Monolithen
 - Mikrokerne
 - Exokerne
 - Hypervisor
- Geschichte: Revolutionen, Religionen ... und die Realität
 - Bewertungskriterien
 - Erfolgs- und Misserfolgsgeschichten
- Beispiele aus der Praxis
 - OS360, Unix, Linux, L4, Windows
 - exoKernel, xen, vmware
 - ...



VL₁₀ Fadensynchronisation

- Grundsätzliches
 - Voraussetzungen
 - aktives und passives Warten
- Synchronisationsprimitiven
 - *Mutex*, *Semaphore* und *Condition*
 - aus der Sicht des BS-Entwicklers
- spezielle Probleme
 - Wechselwirkung Synchronisation $\Rightarrow$ Ablaufplanung
 - Fortschrittsgarantie und Verklemmung
- Beispiele aus der Praxis
 - Synchronisationsprimitiven in Windows



VL₁₁ Interprozesskommunikation (IPC)

- Grundsätzliches
 - Wechselwirkung $\Rightarrow$ Synchronisation
 - implizite und explizite Synchronisation
- Abstraktionen jenseits von *Semaphore*
 - gemeinsamer und verteilter Speicher
 - Fern- und Nahaufrufe
- Dualität nachrichtenbasierter und prozeduraler Systeme
 - konkrete Beispiele
 - Mikrokern $\Leftrightarrow$ Monolith



VL₁₂ PC-Bussysteme

- Herkunft und Architektur
 - ISA und die Folgen
 - Programmiermodell
- Lokale Bussysteme
 - PCI und PCI Express
 - AGP
 - InfiniBand und HyperTransport
 - ...
- E/A-Bussysteme
 - USB, Firewire
 - SCSI, SATA
 - ...



VL₁₃ Geräteprogrammierung

- Treiber und ihre Bedeutung
 - Vielfalt von Geräten
 - Probleme
- Komponentenmodell für Treiber
 - Struktur eines E/A-Systems
 - Treiberklassen und -schnittstellen
- Beispiele aus der Praxis
 - Windows
 - Linux



VL₁₄ Zusammenfassung und Ausblick

- Zusammenfassung des Lernstoffes
- Diskussion der Evaluationsergebnisse
- Tipps und Hinweise für die Prüfung
- Ausblick



Das Endziel...



Viel Spaß!

