Betriebssysteme (BS)

VL 1 – Einführung

Daniel Lohmann

Lehrstuhl für Informatik 4
Verteilte Systeme und Betriebssysteme

Friedrich-Alexander-Universität
Erlangen Nürnberg

WS 11 – 19. Oktober 2011

http://www4.informatik.uni-erlangen.de/Lehre/WS11/V_BS

Vorwort

Die Lehrveranstaltung ist grundsätzlich für alle Studiengänge offen. Sie verlangt allerdings gewisse Vorkenntnisse. Diese müssen nicht durch Teilnahme an den Lehrveranstaltungen von I4 erworben worden sein.

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 Lehrbetriebssysteme
  - Praktische Erfahrungen im Betriebsystembau machen
- **Verstehen** der technologischen Hardware-Grundlagen
  - PC-Technologie verstehen und einschätzen können
  - Schwerpunkt: Intel x86 / IA-32

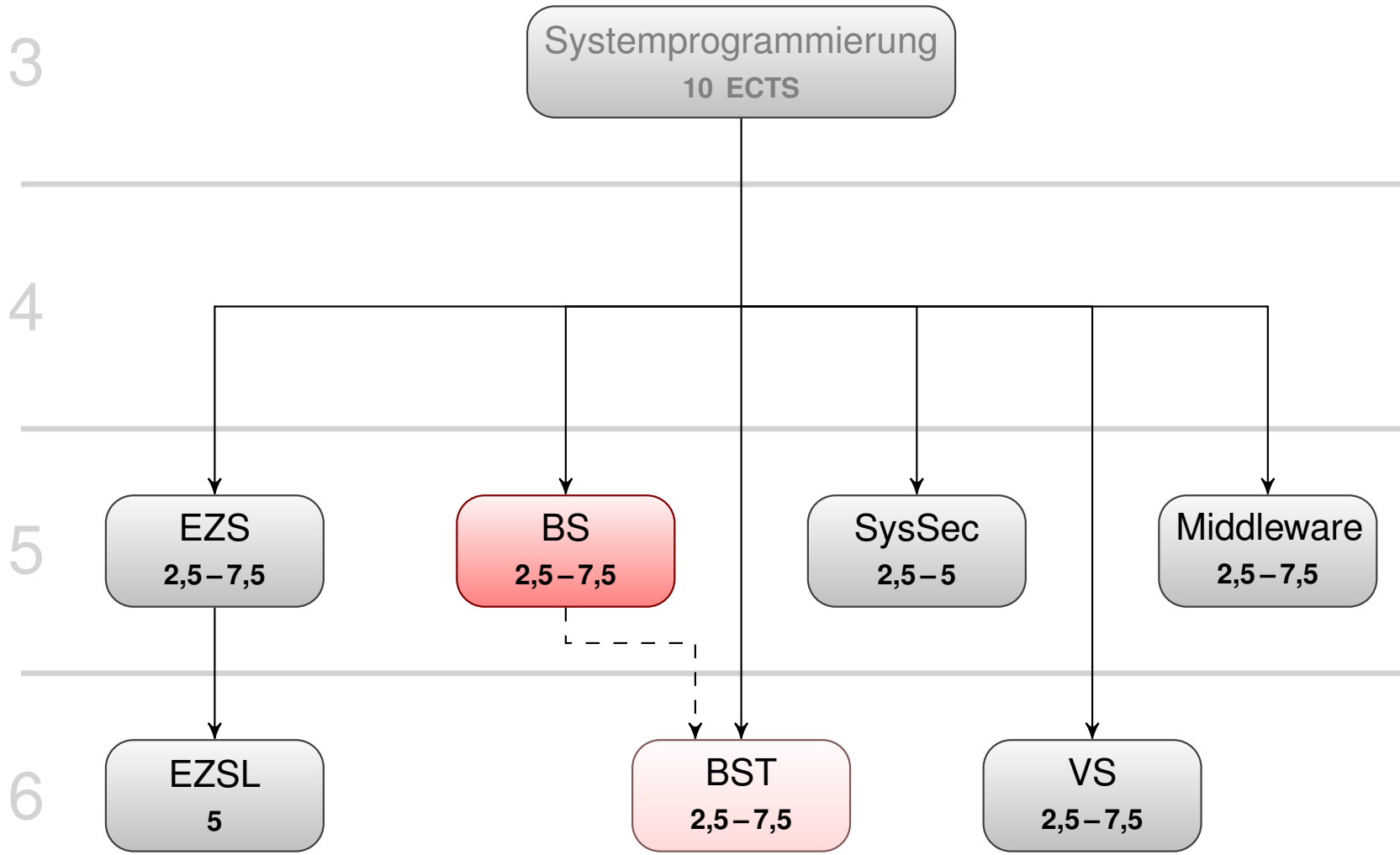
Voraussetzungen

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

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

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

Einpassung in den Musterstudienplan (Bachelor)



Veranstaltungsbestandteile

(2,5 – 7,5 ECTS Punkte)

VL – Vorlesung 2,5

Vorstellung und detailierte 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**

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

Übungsbetrieb

Übung (0.85 / K2, Abgaben in 0.01)

- Zwei Termine zur Auswahl
  - Mittwoch, 14:15 – 15:45 (0.85) *oder* 16:15 – 17:45 (K2)
- Übungsaufgaben sind in 2er-Gruppen zu bearbeiten
- Anmeldung über WAFFEL (URL siehe Webseite)
  - Freischaltung erfolgt nach der Vorlesung, heute im Tagesverlauf

Rechnerübung (0.01)

- Zwei Termine zur Auswahl
  - Montag, 14:15 – 15:45 *oder* Dienstag, 14:15 – 15:45
- Betreuer können auch jederzeit direkt angesprochen werden

Terminübersicht Wintersemester 2011/2012

KW	Mo 14-16	Di 14-16	Mi 12-14	Mi 14-16	Mi 16-18	Raum
17.10.			VL_1			0.031
24.10.			VL_2	$Ü_1$	$Ü_1$	0.85 / K2
31.11.			VL_3	RÜ	RÜ	0.01
07.11.	RÜ	RÜ	VL_4	$Ü_2$	$Ü_2$	
14.11.	RÜ	RÜ	VL_5	A_1	A_1	0.01
21.11.	RÜ	RÜ	VL_6	$Ü_3$	$Ü_3$	
28.11.	RÜ	RÜ	VL_7	A_2	A_2	
05.12.	RÜ	RÜ	VL_8	$Ü_4$	$Ü_4$	
12.12.	RÜ	RÜ	VL_9	$Ü_5$	$Ü_5$	
19.12.	A_3	A_3				
09.01.	RÜ	RÜ	VL_{10}	A_4	A_4	
16.01.	RÜ	RÜ	VL_{11}	$Ü_6$	$Ü_6$	
23.01.	RÜ	RÜ	VL_{12}	A_5	A_5	
30.01.	RÜ	RÜ	VL_{13}	$Ü_7$	$Ü_7$	
06.02.			VL_{14}	A_6	A_6	

Dozenten und Übungsleiter

Vorlesung

Daniel Lohmann

Wolfgang Schröder-Preikschat

Übung

Benjamin Oechslein

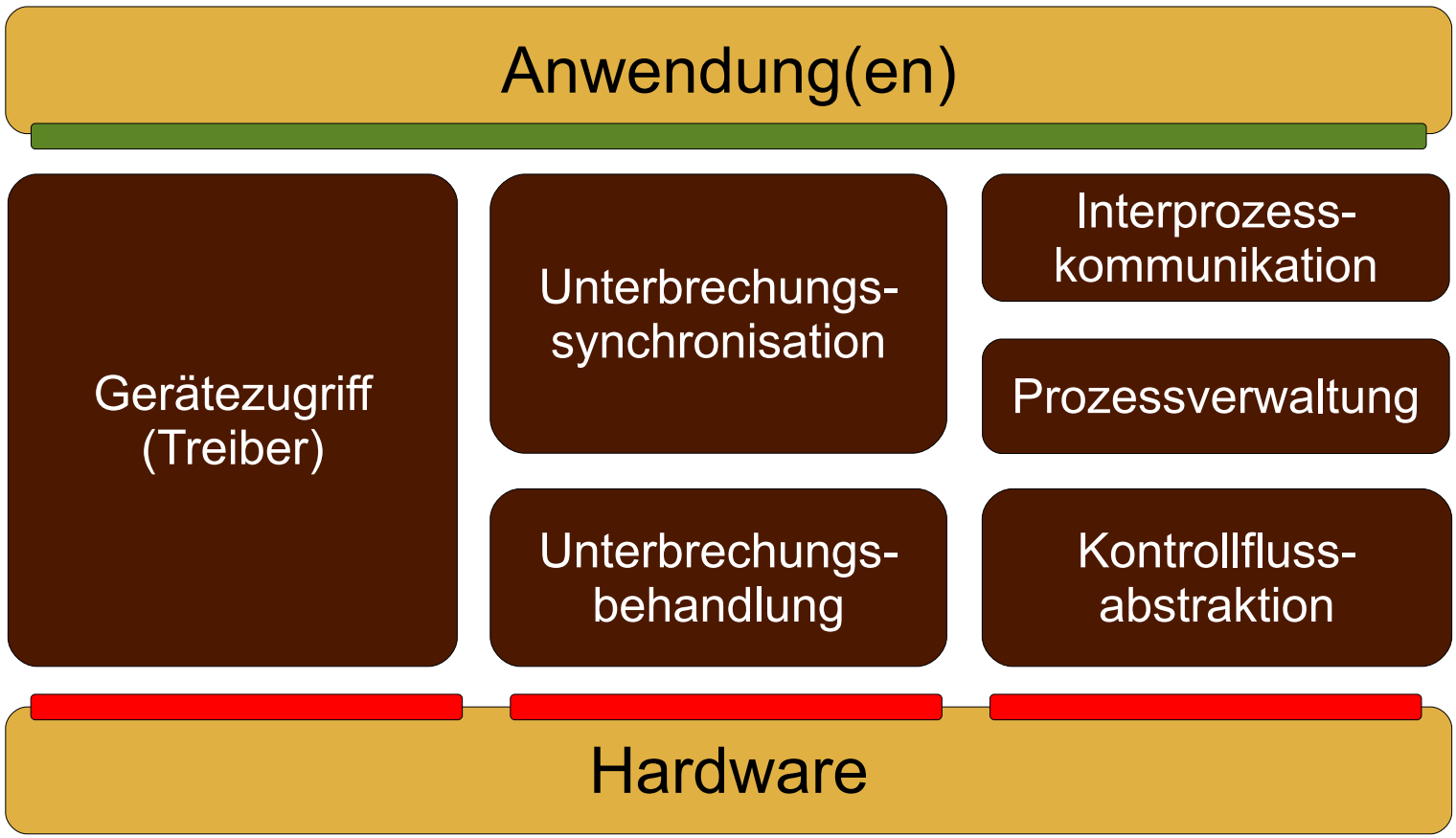
Isabella Thomm

Rechnerübung

Isabella Thomm

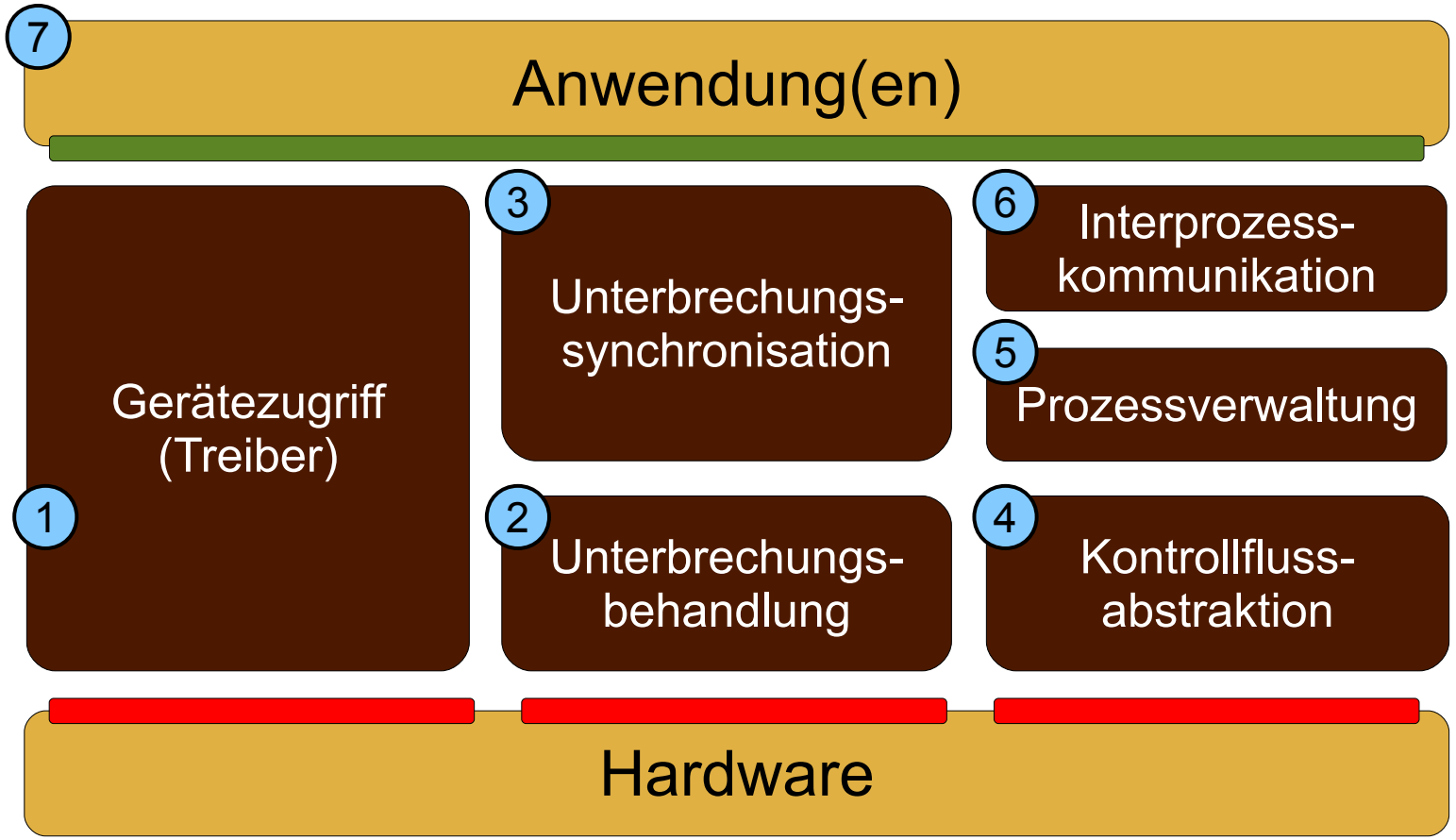
Dirk Wischermann

Aufbau eines Betriebssystem



Themenübersicht Übung

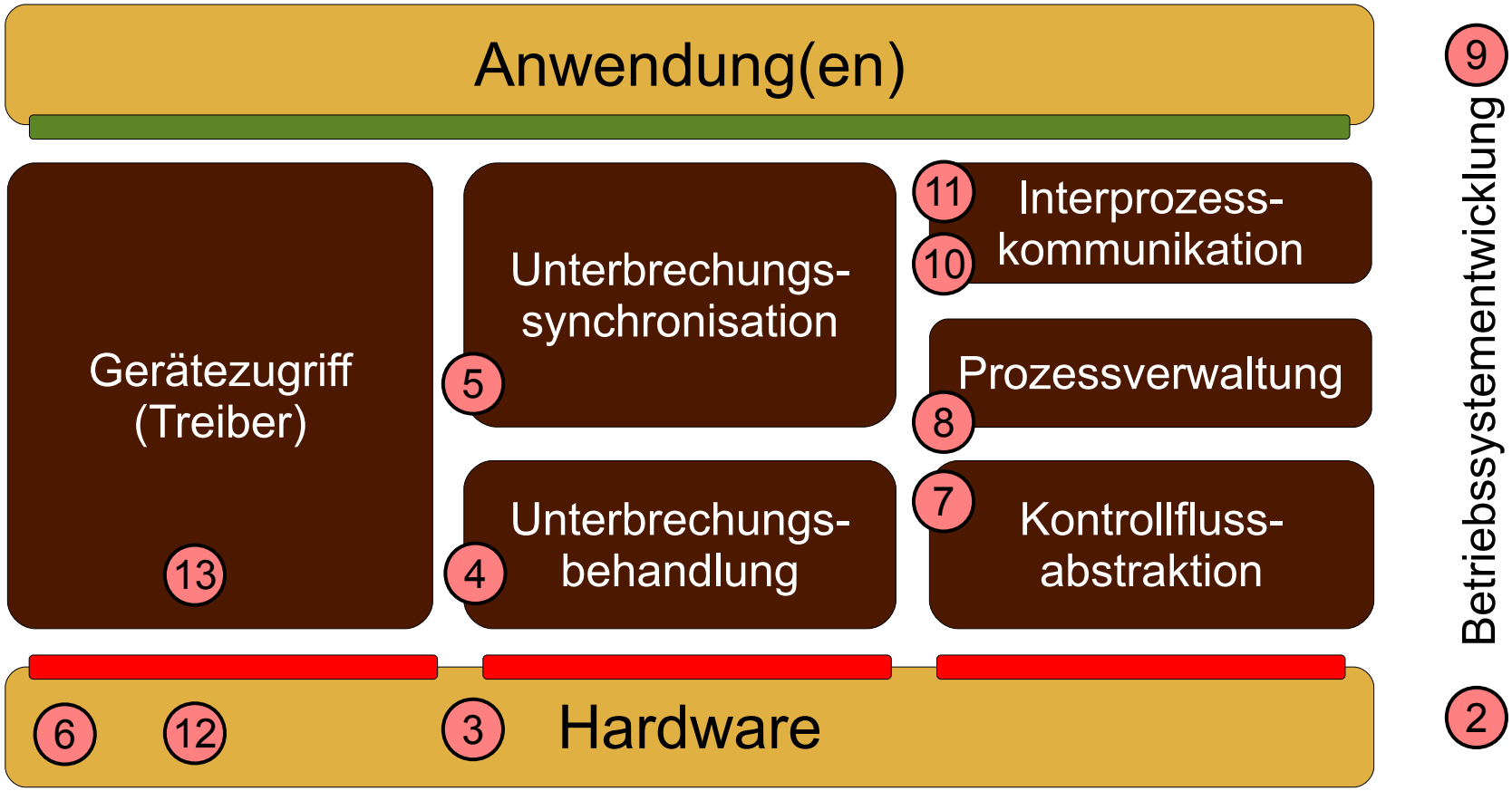
Am Beispiel von: OOStuBS, MPStuBS



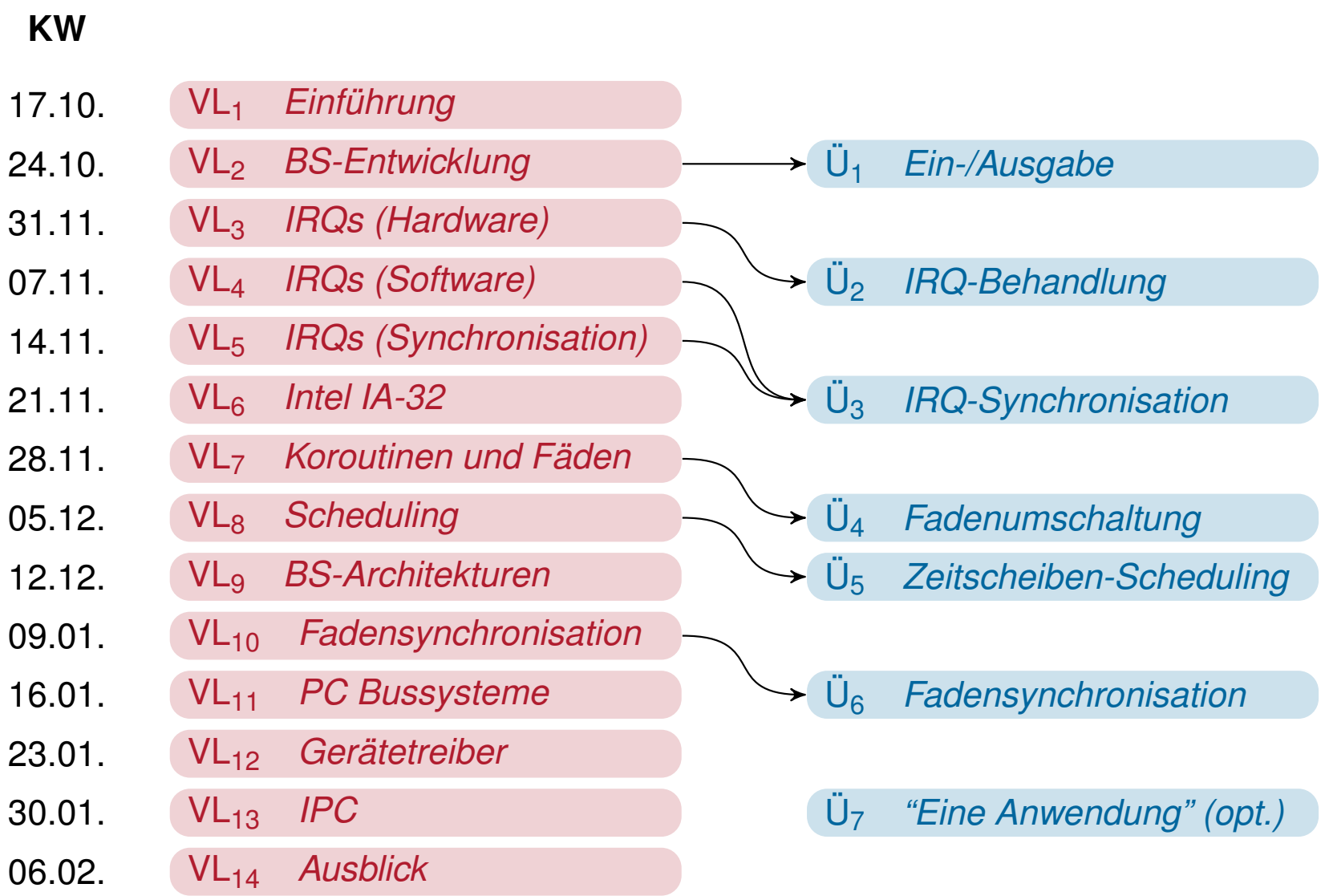
Betriebssystementwicklung

Themenübersicht Vorlesung

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



Verzahnung von Vorlesung und Übungsaufgaben



VL_2 Betriebssystementwicklung (oft ein harter Kampf!)

- Erste Schritte

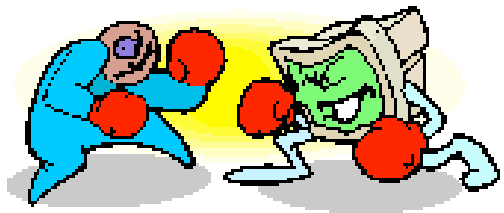
  Wie bringt man sein System auf die Zielhardware?

  - 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

VL_5 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

VL_6 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

VL_7 Programmfäden

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

VL_8 Ablaufplanung

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

VL_9 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
  - . . .

VL_{10} Fadensynchronisation

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

VL_{11} Interprozesskommunikation (IPC)

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

VL_{12} 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
  - ...

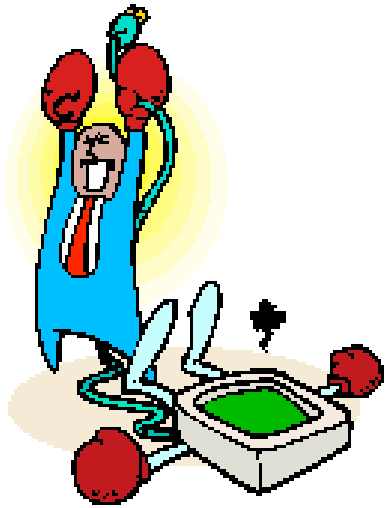
VL_{13} Geräteprogrammierung

- 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

VL_{14} Zusammenfassung und Ausblick

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

Das Endziel. . .



Viel Spaß!