

Systemprogrammierung

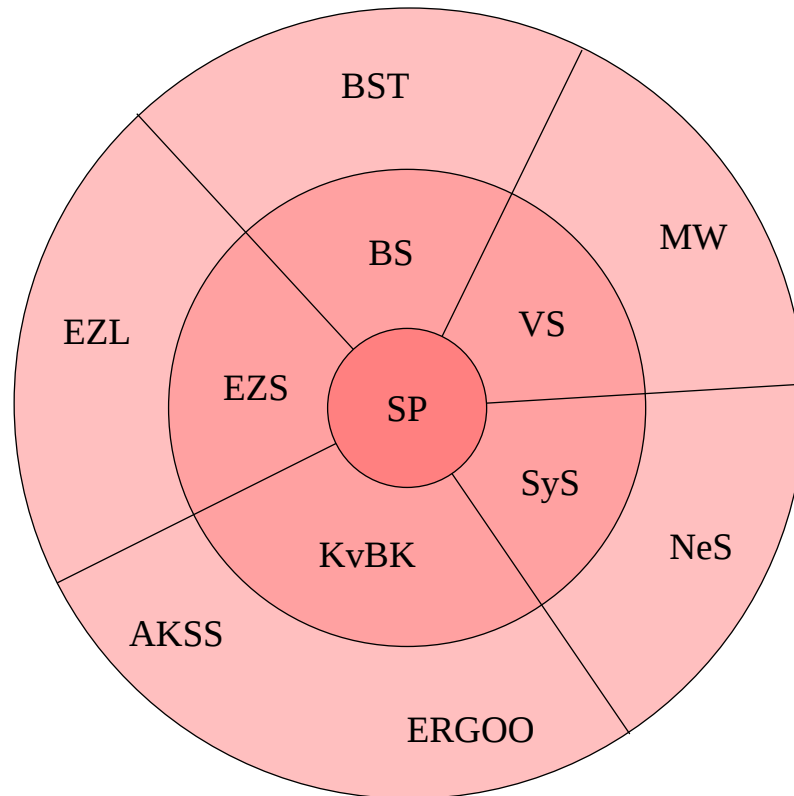
Lehrveranstaltungs-konzept

16. Oktober 2008

Überblick

Lehrveranstaltungs-konzept

- Einordnung
- Studiengänge
- Lernziele und Lehrinhalte
- Voraussetzungen
- Vorlesung und Übung
- Leistungsnachweise
- Kontakt



Lehre@I4: *post* SP — Aufbau und Spezialisierung

BS	Betriebssysteme	V/Ü
VS	Verteilte Systeme	V/Ü
EZS	Echtzeitsysteme	V/Ü
SyS	Systemicherheit	V/Ü
KvBK	Konzepte von Betriebssystemkomponenten	PS

BST	Betriebssystemtechnik	V/Ü
MW	Middleware	V/Ü
EZL	Echtzeitsystemlabor	IV
NeS	Netzwerksicherheit $\mapsto$ I7	V/Ü
AKSS	Ausgewählte Kapitel der Systemsoftware	HS
ERGOO	Erlangen Research Group on Distributed Objects and Operating Systems	KO

Integrierte Lehrveranstaltung: 8 SWS $\mapsto$ 10 ECTS

$$\text{Termine} \left\{ \begin{array}{ll} \text{Vorlesung} & 2 \\ \text{Übung} & 1 \\ \text{Rechner} & 1 \end{array} \right\} \equiv 4 \times 1,5 = 6 \text{ Zeitstunden wöchentlich}$$

Vor-/Nacharbeit

- ▶ N Stunden wöchentlich: $0 \leq N \leq (162 - X)$
- ▶ $X \ll 162$ ist das Zeitstundenäquivalent anderer „Pflichten“

$$\text{Arbeitsaufwand} \leadsto 30 \times V + 16 \times \ddot{U} + 16 \times P$$

Präsenzzeit 124 h, davon 93 h Kernzeit $\approx$ 6 Wochenstunden

Eigenstudium $1,5 \times \text{Kernzeit} = 139,5 \text{ h} \approx 9 \text{ Wochenstunden}$

Kernfach der Informatik

Diplom/Bachelor

- ▶ Informatik, IuK
- ▶ Mathematik (Wahlfach Informatik), Technomathematik
- ▶ Wirtschaftsinformatik

Bachelor

- ▶ Computational Engineering (CE)

Lehramt Informatik

- ▶ Gymnasium

Lernziele

Vorgänge innerhalb von Rechensystemen *ganzheitlich* verstehen:



Programmierung von **Systemsoftware** (in Grundzügen) selbst erleben

- ▶ im Kleinen praktizieren ➞ **Dienstprogramme**
- ▶ im Großen erfahren ➞ **Betriebssysteme**

Lehrinhalte

Vorlesung — Vorstellung und detaillierte Behandlung des Lehrstoffs

- ▶ Organisation (der Systemsoftware) von Rechensystemen
- ▶ Grundlagen von Betriebssystemen
- ▶ maschinennahe Programme

Übung — Vertiefung, Besprechung der Übungsaufgaben, Tafelübungen

- ▶ Systemprogrammierung in C
- ▶ Systemprogramme, -aufrufe, -funktionen von UNIX

Praktikum — Rechnerarbeit: Programmierung, Fehlersuche/-beseitigung

- ▶ UNIX (Linux), CLI (*shell*), GNU (gcc, gdb, make), vi. . .

Inhaltsüberblick

Teil A $\mapsto$ SPiC^a

1. Organisation
2. Einführung in C
3. Programm $\mapsto$ Prozess (UNIX)

Teil B $\mapsto$ Überblick

4. Einleitung
5. Rechnerorganisation
6. Betriebsarten
7. Abstraktionen (UNIX)
8. Zwischenbilanz

Teil C $\mapsto$ Vertiefung

9. Einplanung
10. Einlastung
11. Synchronisation
12. Verklemmungen
13. Adressräume
14. Arbeitsspeicher
15. Dateisysteme

^aSystemnahe Programmierung in C

Erforderliche Grundkenntnisse

Algorithmik $\mapsto$ Grundlagen strukturierter Programmierung

- ▶ Datentypen, Kontrollkonstrukte, Prozeduren
- ▶ statische und dynamische Datenstrukturen
- ▶ „Programmierung im Kleinen“

Technische Informatik $\mapsto$ Grundlagen der Rechnerorganisation

- ▶ „von Neumann Architektur“
 - ▶ Operationsbefehle, Befehlsoperanden, Adressierungsarten
 - ▶ Unterbrechungssteuerung (Pegel kontra Flanke)
 - ▶ Assemblerprogrammierung
- ▶ CPU, DMA, FPU, IRQ, MCU, MMU, NMI, PIC, TLB

Vorlesungsbetrieb und Lehrmaterialien

Vorlesungstermine ab KW 45/46

Montag	10:15 – 11:45	H8
Donnerstag	14:15 – 15:45	H9

C/UNIX Kompaktkurs

- ▶ 6–7 Vorlesungstermine
- ▶ direkt im Anschluss...

Handzettel (engl. *handout*) sind verfügbar wie folgt:

1. www4.informatik.uni-erlangen.de/Lehre/WS08/V_SP
2. Gutscheinverkauf zum Bezug von Folienkopien
 - ▶ die Kopien werden vor der Vorlesung ausgegeben
 - ▶ Schutzgebühr 1 EUR

Fachbegriffe der Informatik (Deutsch ↔ Englisch)

- ▶ www.babylonia.ork.uk
- ▶ www.inf.fu-berlin.de/inst/ag-ss/montagswort
- ▶ www.aktionlebendigesdeutsch.de

Ergänzende Literatur



Brian W. Kernighan and Dennis MacAlistair Ritchie.

The C Programming Language.

Prentice-Hall, Inc., second edition, 1988.



Jürgen Nehmer and Peter Sturm.

Systemsoftware: Grundlagen moderner Betriebssysteme.

dpunkt.Verlag GmbH, zweite edition, 2001.



Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne.

Operating System Concepts.

John Wiley & Sons, Inc., sixth edition, 2001.



Andrew Stuart Tanenbaum.

Structured Computer Organization.

Prentice-Hall, Inc., fourth edition, 1999.



Wolfgang Schröder-Preikschat.

Betriebssysteme — Grundlagen, Entwurf, Implementierung.

Springer, 2009.

Übungsbetrieb

Tafelübung: Beginn in KW 44 (Montag)

- ▶ Anmeldung über **WAFEL**¹ (URL siehe Webseite von SP)
 - ▶ Freischaltung erfolgt nach der Vorlesung, heute gegen 16:00 Uhr
- ▶ Übungsplätze werden FCFS² vergeben
 - ▶ unterbelegte Termine werden ggf. gestrichen
 - ▶ überbelegte Termine erhalten ggf. mehr Ressourcen
- ▶ Übungsaufgaben sind teilweise in Zweiergruppen zu bearbeiten
 - ▶ Übungspartner müssen für dieselbe Tafelübung angemeldet sein

Rechnerübung: Anmeldung ist nicht erforderlich

- ▶ die Teilnahme ist optional, je nach Bedarf
 - ▶ es können auch mehrere Termine pro Woche wahrgenommen werden
- ▶ Übungsleiter stehen bei Fragen zur Verfügung

¹Abk. für Webanmeldefrickelformular Enterprise Logic

²Abk. für engl. *first come, first served*, Einplanung nach Ankunftszeit

Bedeutung von Tafel- und Rechnerübungen

Tafelübungen $\leadsto$ „*learning by exploring*“

- ▶ Besprechung der Übungsaufgaben, Skizzierung von Lösungswegen
- ▶ Vertiefung des Vorlesungsstoffes, Klärung offener Fragen

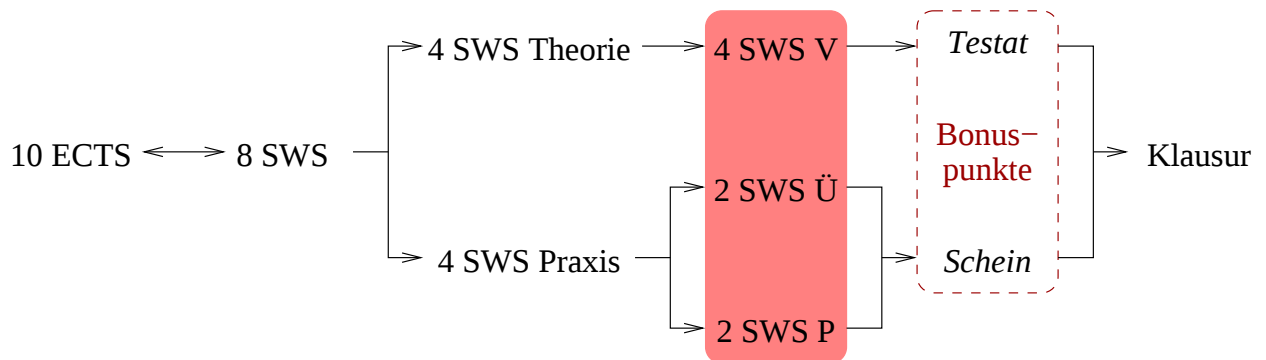
Rechnerübungen $\leadsto$ „*learning by doing*“

- ▶ selbständiges Bearbeiten der Übungsaufgaben am Rechner
- ▶ Hilfestellung beim Umgang mit den Entwicklungswerkzeugen
- ▶ der Rechner ist **kein Tafelersatz**, die Betreuung verläuft eher passiv

*Der, die, das.
Wer, wie, was?
Wieso, weshalb, warum?
Wer nicht fragt, bleibt dumm!*



Studien- und Prüfungsleistungen



1. Klausur: schriftliche Prüfung von 120 Minuten

- ▶ Einbringung von Bonuspunkten zur Notenverbesserung möglich

2. Bonuspunkte: **unbenoteter Übungsschein** als Nachweis

- ▶ ergeben sich anteilig aus Testat und Übungsaufgaben

Testat und Schein

Testat $\sim$ Ex $\mapsto$ **Miniklausur**: optional, „beliebiger“ Vorlesungstermin

- ▶ geprüft wird Stoff von Vorlesung und Übung
 - ▶ Fragen zu **Teil A** und **Teil B** der Vorlesung
 - ▶ Trockenübung in der Programmiersprache C
- ▶ Mehrfachauswahl (engl. *multiple choice*), 45 Minuten

Schein: obligatorisch, erfolgreiche Teilnahme an Übung und Praktikum

- ▶ 50 % der Punkte aus „Übungsaufgaben + Ex“ sind zu erreichen
- ▶ Punkte darüberhinaus gehen in die Bonuspunktberechnung ein

Bonuspunkte für die Klausur ansammeln

- ▶ max. 10 % der Klausurpunkte erreichbar $\approx$ 12 Punkte
- ▶ Erfahrungswert: $\approx$ 5.5 Klausurpunkte pro Notenstufe 0.3
- ▶ Verbesserung der Klausurnote um bis zu zwei Stufen möglich

Bonuspunkte und Klausur

Bonuspunkte: nur auf Basis des (unbenoteten) Scheins aus WS 2008/09

- ▶ beeinflussen die angelegte Notenskala nicht, werden jedoch bei bestandenen Klausuren auf die Klausurpunkte aufaddiert
- ▶ können die Note einer bestandenen Klausur verbessern, nicht jedoch den Ausschlag zum Bestehen der Klausur geben
- ▶ Erreichen der Bestehensgrenze muss also immer mit regulären Klausurpunkten erfolgen

Klausur: Termin steht noch nicht fest

- ▶ analog zum Testat, zusätzlich Fragen zu **Teil C** der Vorlesung

Präsenz und **aktive Mitarbeit** machen die Klausur „leicht“

- ▶ Programme im Team entwickeln, aber selbst zum Laufen bringen

www4.informatik.uni-erlangen.de/*

Dozenten

- ▶ Jürgen Kleinöder (~jklein)
- ▶ Wolfgang Schröder-Preikschat (~wosch)

Mitarbeiter

- ▶ Michael Stilkerich (~stilkerich)

Tutoren

- ▶ einige...



Fragen...

42