

Systemprogrammierung

Lehrveranstaltungs-konzept

Wolfgang Schröder-Preikschat

Lehrstuhl Informatik 4

9. April 2014

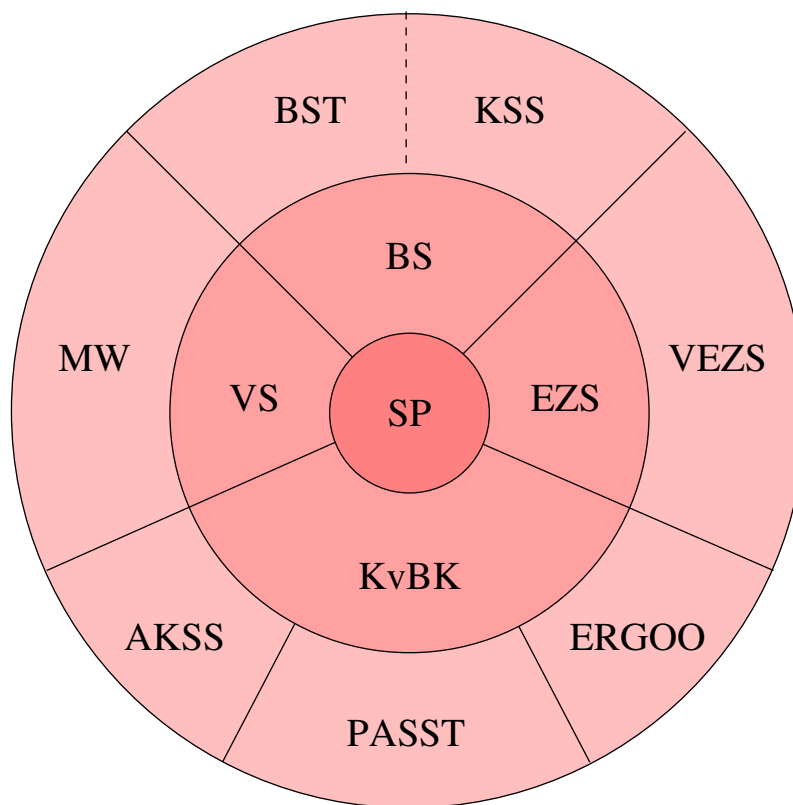
Gliederung

1 Konzept

- Lehrkanon
- Lehrziele
- Vorkenntnisse
- Veranstaltungsbetrieb
- Leistungsnachweise

2 Kontakt

Lehre@I4



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

Aufbau

| | | |
|-------------|--|-----|
| BS | Betriebssysteme | v/Ü |
| VS | Verteilte Systeme | v/Ü |
| EVS | Echtzeitsysteme | v/Ü |
| KvBK | Konzepte von Betriebssystemkomponenten | Sem |

Spezialisierung

| | | |
|--------------|--|-----|
| BST | Betriebssystemtechnik | v/Ü |
| KSS | Konfigurierbare Systemsoftware | v/Ü |
| MW | Middleware - Cloud Computing | v/Ü |
| VEZS | Verlässliche Echtzeitsysteme | v/Ü |
| PASST | Praktikum angewandte Systemsoftwaretechnik | PR |
| AKSS | Ausgewählte Kapitel der Systemsoftware | Sem |
| ERGOO | Erlangen Research Group on Distributed Objects and Operating Systems | KO |

Module SP (10 ECTS) und GSP (5 ECTS)

Systemprogrammierung (SP) ~ geteiltes Modul (vgl. S. 9)

- i Systemprogrammierung 1 (SP1) \mapsto Vorlesungsteile A und B 5 ECTS
- ii Systemprogrammierung 2 (SP2) \mapsto Vorlesungsteil C 5 ECTS

- der Stoff von SP2 ist „kausal abhängig“ vom Stoff von SP1
 - SP1 liefert Grundlagen für SP2 (vgl. S. 11)
- beide Modulhälften sind Grundlage insbesondere von BS und BST

Grundlagen der Systemprogrammierung (GSP)

- nur Systemprogrammierung 1 (SP1) 5 ECTS

Studiengänge und Zuordnung der Modulhälften

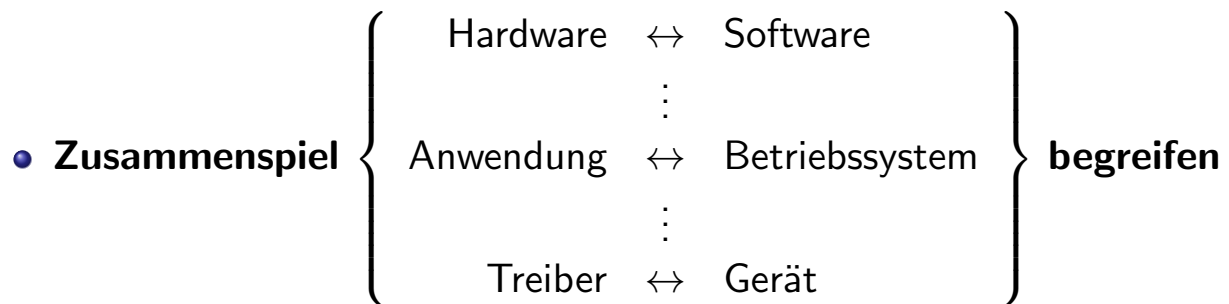
| Abschluss | Studiengang | SP1 | SP2 |
|-----------|---|-----|-----|
| Bachelor | Informatik | × | × |
| | Informations- und Kommunikationstechnik | × | × |
| | Computational Engineering | × | × |
| | Wirtschaftsinformatik | × | × |
| | Informatik, 2-Fach Bachelor | × | |
| Lehramt | Informatik, Gymnasium | × | × |

Alternative zu [Systemnahe Programmierung in C \(SPiC\)](#):

| | | |
|----------|----------------------------------|---|
| Bachelor | Mathematik, Nebenfach Informatik | × |
| | Technomathematik | × |

Lernziele

Vorgänge innerhalb von Rechensystemen **ganzheitlich** verstehen



Grundzüge imperativer Systemprogrammierung (in C)

- im Kleinen praktizieren \leadsto Dienstprogramme
- im Großen erfahren \leadsto 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

Rechnerarbeit — Programmierung, Fehlersuche/-beseitigung

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

Inhaltsüberblick: Kapitelzuordnung und -folge

I. Lehrveranstaltungsüberblick

Teil A ~ C-Programmierung

II. Einführung in C

III. Programm \mapsto Prozess (UNIX)

Teil B ~ Grundlagen

IV. Einleitung

V. Rechnerorganisation

VI. Abstraktionen (UNIX)

VII. Betriebsarten

VIII. Zwischenbilanz SP1

XIV. Fragestunde SP1 & SP2

Teil C ~ Vertiefung

IX. Prozessverwaltung

- Einplanung
- Einlastung

X. Koordinierung

- Synchronisation

XI. Betriebsmittelverwaltung

XII. Speicherverwaltung

- Adressräume
- Arbeitsspeicher

XIII. Dateisysteme

- Speicherung
- Fehlererholung

Voraussetzungen zum Verständnis des Lehrstoffs

Obligatorisch

Algorithmen & Datenstrukturen \mapsto Grundlagen der Programmierung

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

Wünschenswert, jedoch nicht zwingend erforderlich...

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

Abhängigkeiten zwischen den Vorlesungsteilen

Systemprogrammierung 1

- Teil A**
- setzt grundlegende Programmierkenntnisse voraus
 - vermittelt Grundlagen der **Programmierung in C**
- Teil B**
- setzt grundlegende Programmierkenntnisse in C voraus
 - vermittelt **Operationsprinzipien von Betriebssystemen**

Systemprogrammierung 2

- Teil C**
- setzt Kenntnisse erwähnter Operationsprinzipien voraus
 - vermittelt die **interne Funktionsweise** von Betriebssystemen

Erlangung der benötigten Vorkenntnisse

- (i) durch Vorlesungsteilnahme
- empfohlene sequentielle Belegung der Vorlesungsteile
- (ii) durch Lehrbuchlektüre, aus anderen Lehrveranstaltungen, ...

Vorlesungsbetrieb und Lehrmaterialien

Vorlesungstermine ab KW 15

SP1 Mittwoch 10:15 H11

SP1-Zusatztermine

10.04.2014 16:00 H7

15.04.2014 18:00 H11

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

- 1 Austeilung von Folienkopien vor der Vorlesung
- 2 www4.cs.fau.de/Lehre/SS14/V_SP1

Fachbegriffe der Informatik (Deutsch ↔ Englisch)

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

Ergänzende Literatur (nicht nur) zum Vorlesungsstoff

- [1] KERNIGHAN, B. W. ; RITCHIE, D. M.:
The C Programming Language.
Prentice-Hall, Inc., 1988. –
ISBN 0-131-10362-8
- [2] NEHMER, J. ; STURM, P. :
Systemsoftware: Grundlagen moderner Betriebssysteme.
dpunkt.Verlag GmbH, 2001. –
ISBN 3-898-64115-5
- [3] SCHRÖDER-PREIKSCHAT, W. :
Betriebssysteme — Grundlagen, Entwurf, Implementierung.
Springer, 2012 (geplant; Manuskriptauszüge erhältlich)
- [4] SILBERSCHATZ, A. ; GALVIN, P. B. ; GAGNE, G. :
Operating System Concepts.
John Wiley & Sons, Inc., 2001. –
ISBN 0-471-41743-2
- [5] TANENBAUM, A. S.:
Structured Computer Organization.
Prentice-Hall, Inc., 1979. –
443 S. –
ISBN 0-130-95990-1

Übungsbetrieb

Tafelübung: Anmeldung mit **WAFFEL**¹ über die Startseite von SP

- die Periode der Übungswoche startet mit zweiter Vorlesungswoche
 - www4.cs.fau.de/Lehre/SS14/V_SP1
- Übungsplätze werden FCFS² vergeben
 - unterbelegte Termine werden ggf. gestrichen
 - überbelegte Termine erhalten ggf. mehr Ressourcen
 - die Plätze werden nach Beginn der Anmeldung angepasst
- 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

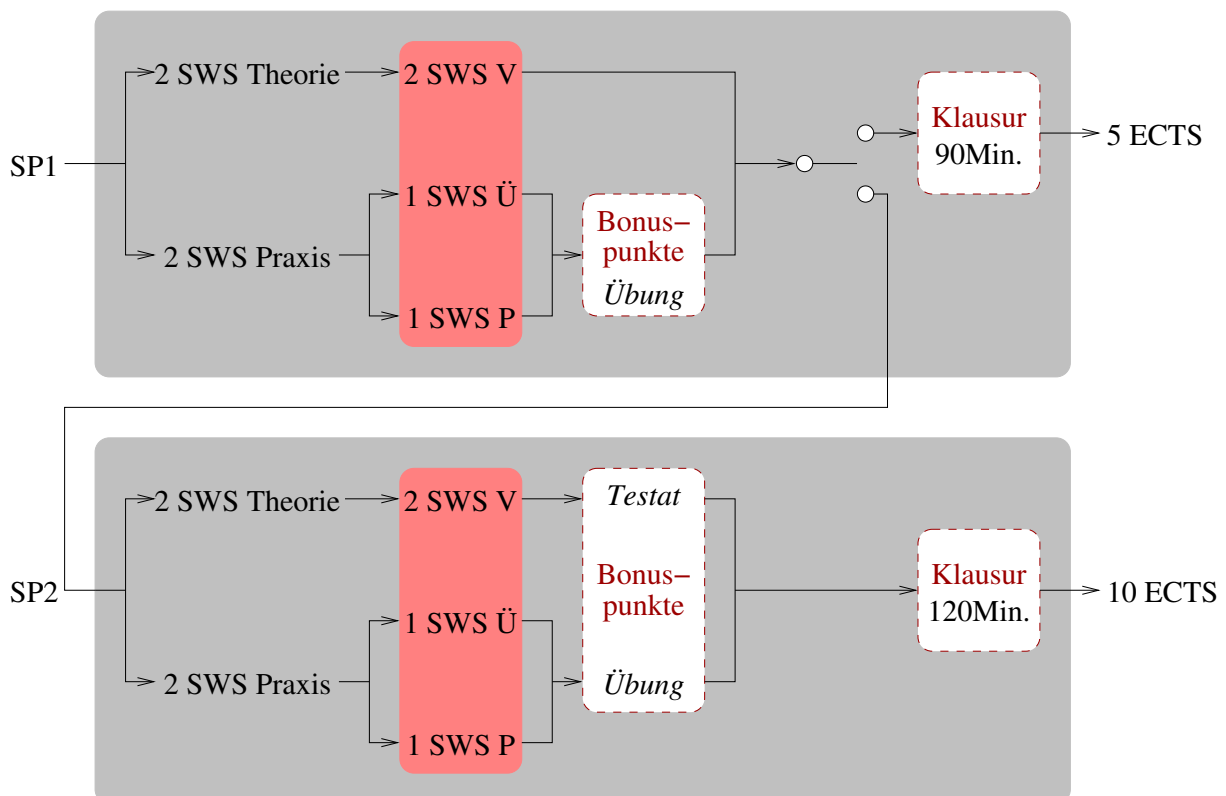
Rechnerübungen \leadsto „*learning by doing*“

- 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



Übungsaufgaben und Miniklausur

Übungsaufgaben: 6 (SP1) + 5 (SP2) Programmieraufgaben

- abgegebene Programme werden korrigiert und bewertet

Miniklausur: am **Anfang von Teil C** (SP2)

- geprüft wird Stoff von Vorlesung und Übung, 30 Minuten
 - Fragen zu **Teil A** und **Teil B** der Vorlesung
 - Trockenübung in der Programmiersprache C
- mit Anteilen von Mehrfachauswahl (engl. *multiple choice*)

Notenbonus für die Klausur

- verlangt sind 50 % der Punkte aus „Übungsaufgaben + Miniklausur“
- Punkte darüberhinaus gehen in die Bonusberechnung ein
- maximal ist ein Notenbonus von 0,7 erreichbar

Notenbonus und Klausur

Notenbonus: nur auf Basis der Übungen **des laufenden SP-Moduls**

- beeinflusst die Punkte-Notenskala der Klausur nicht, wird jedoch bei **bestandener** Klausur auf die Klausurnote angewendet (abgezogen)
- kann 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, Anfang vorlesungsfreie Zeit

GSP • Struktur analog Miniklausur (S. 17), jedoch 90 Minuten Dauer

SP • Struktur analog GSP, jedoch 120 Minuten Dauer

- zusätzlich Fragen zu **Teil C** der Vorlesung

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

- Programme im Team entwickeln, aber selbst zum Laufen bringen

Gliederung

1 Konzept

- Lehrkanon
- Lehrziele
- Vorkenntnisse
- Veranstaltungsbetrieb
- Leistungsnachweise

2 Kontakt

www4.cs.fau.de/*



Dozenten

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

Mitarbeiter

- Jens Schedel (~schedel)

Tutoren

- | | |
|---------------------------|--------------------------|
| • Christian Bay | • Nicolas Pfeiffer |
| • Felix Dreißig | • Sebastian Rachuj |
| • Hans-Peter Deifel | • Stefan Reif |
| • Maximilian Eschenbacher | • Andreas Ruprecht |
| • Stephan Gabert | • Tobias Sammet |
| • Daniel Laffling | • Christian Schlumberger |
| • Florian Meier | • Florian Schmauss |

Fragen...

42