

Systemprogrammierung

Lehrveranstaltungs-konzept

Wolfgang Schröder-Preikschat

Lehrstuhl Informatik 4

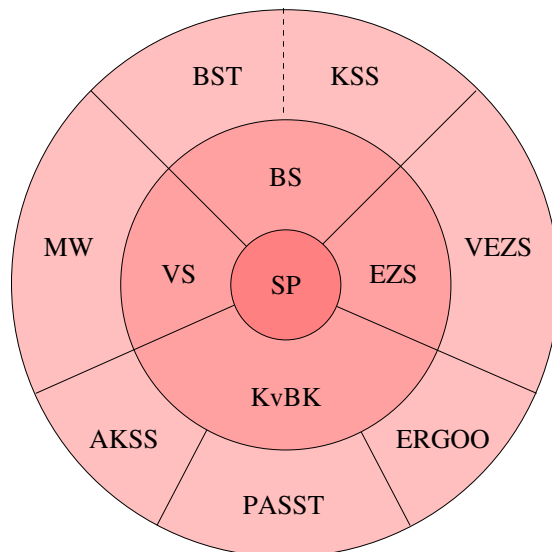
16. April 2013

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
EZS	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. 10)

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

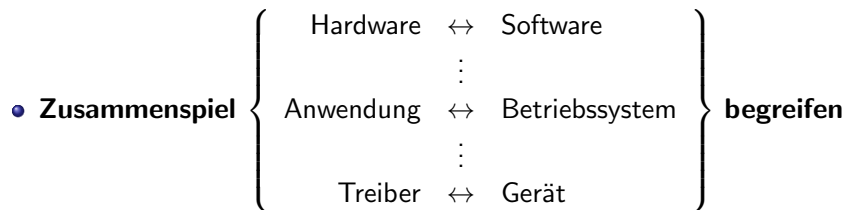
- beide Modulhälften werden dieses Semester zum letzten Mal **zugleich** angeboten
 - sie sind jedoch verschiedenen „Generationen“ zugeordnet
 - „Studienbeginn WS“ einerseits und „Studienbeginn SS“ andererseits
 - SP1 im „2. Fachsemester“ || SP2 im „3. Fachsemester“
 - der Stoff von SP2 ist „kausal abhängig“ vom Stoff von SP1 (vgl. S. 12)
- ⇒ von der Belegung beider Hälften im selben Semester wird abgeraten

Grundlagen der Systemprogrammierung (GSP)

- nur Systemprogrammierung 1 (SP1) 5 ECTS

Lernziele

Vorgänge innerhalb von Rechensystemen **ganzheitlich** verstehen



Grundzüge imperativer Systemprogrammierung (in C)

- im Kleinen praktizieren ~ Dienstprogramme
- im Großen erfahren ~ Betriebssysteme

Studiengänge und Zuordnung der Modulhälften

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

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

Bachelor	Mathematik, Nebenfach Informatik	×	
	Technomathematik	×	

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

Rechnerübungen — 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 16**
- SP1 Donnerstag 14:15 H7
- SP1-Zusatztermine**
- Dienstag, 16.04.2013 16:00 H8
- Dienstag, 23.04.2013 16:00 H8

- Ausfalltermine**
- SP1 Christi-Himmelfahrt (09.05.), Fronleichnam (30.05.)

- Handzettel** (engl. *handout*) sind verfügbar wie folgt:
 - 1 Austeilung von Folienkopien vor der Vorlesung
 - 2 www4.cs.fau.de/Lehre/SS13/V_SP1
- Fachbegriffe** der Informatik (Deutsch \leftrightarrow 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

Bedeutung von Tafel- und Rechnerübungen

Tafelübungen ~> „learning by exploring“

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

Rechnerübungen ~> „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!



Übungsbetrieb

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

- die Periode der Übungswoche startet mit zweiter Vorlesungswoche
 - www4.cs.fau.de/Lehre/SS13/V_SP1
- 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

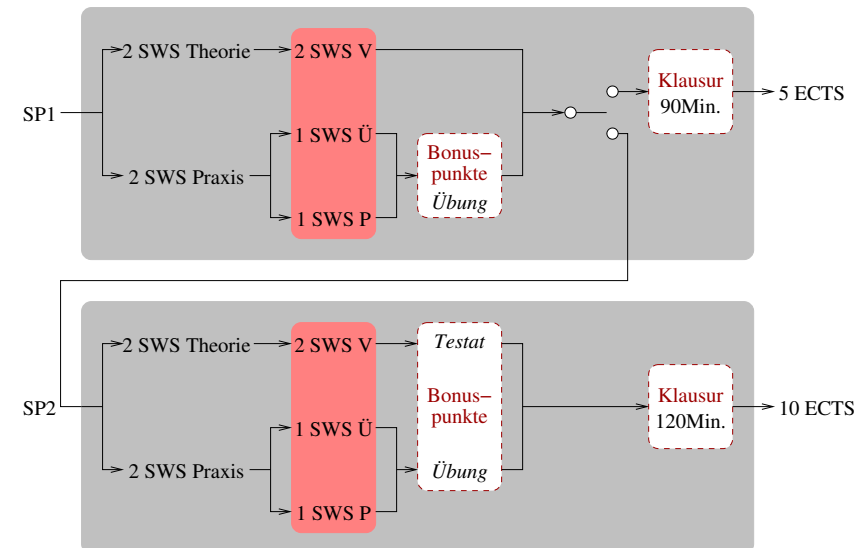
Rechnerübung: Anmeldung ist nicht erforderlich

- 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

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

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

Gliederung

- 1 Konzept
 - Lehrkanon
 - Lehrziele
 - Vorkenntnisse
 - Veranstaltungsbetrieb
 - Leistungsnachweise

2 Kontakt

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 (d. h. 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. 18), 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

www4.cs.fau.de/*



Dozenten

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

Mitarbeiter

- Christoph Erhardt (~erhardt)
- Jens Schedel (~schedel)

Tutoren

- Christopher Bross
- Christoph Egger
- Dominik Ernst
- Hans-Peter Deifel
- Maximilian Eschenbacher
- Stefan Konopik
- Daniel Laffling
- Florian Meier
- Dominik Paulus
- Sebastian Rachuj
- Stefan Reif
- Simon Ruderich
- Andreas Ruprecht
- Tobias Sammet
- Tobias Scharpff
- Christian Schlumberger
- Florian Schmaus

Fragen...

42