

# Systemprogrammierung

## Lehrveranstaltungskonzept

Wolfgang Schröder-Preikschat, Jürgen Kleinöder

Lehrstuhl Informatik 4

15. Oktober 2012

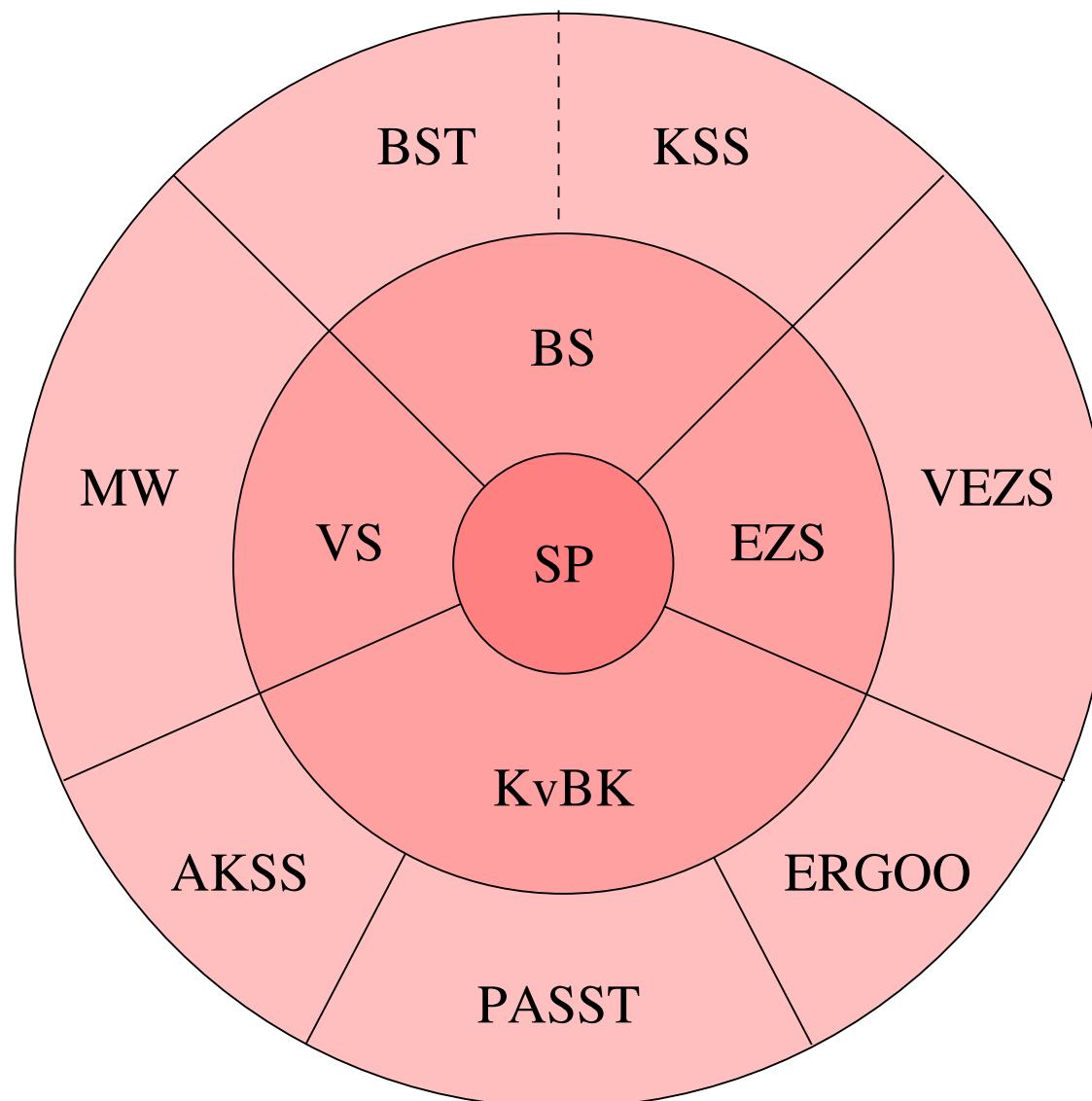
# Gliederung

## 1 Konzept

- Lehrkanon
- Lehrziele
- Vorkenntnisse
- Veranstaltungsbetrieb
- Leistungsnachweise

## 2 Kontakt

## Lehre@I4



# Lehre@I4: post SP — Aufbau und Spezialisierung

## Aufbau

<b>BS</b>	Betriebssysteme	V/Ü
<b>VS</b>	Verteilte Systeme	V/Ü
<b>EZS</b>	Echtzeitsysteme	V/Ü
<b>KvBK</b>	Konzepte von Betriebssystemkomponenten	Sem

## Spezialisierung

<b>BST</b>	Betriebssystemtechnik	V/Ü
<b>KSS</b>	Konfigurierbare Systemsoftware	V/Ü
<b>MW</b>	Middleware - Cloud Computing	V/Ü
<b>VEZS</b>	Verlässliche Echtzeitsysteme	V/Ü
<b>PASST</b>	Praktikum angewandte Systemsoftwaretechnik	PR
<b>AKSS</b>	Ausgewählte Kapitel der Systemsoftware	Sem
<b>ERGOO</b>	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) ↳ Vorlesungsteile A und B 5 ECTS
- ii Systemprogrammierung 2 (SP2) ↳ Vorlesungsteil C 5 ECTS

- beide Modulhälften werden **semesterweise zugleich** angeboten
  - sie sind jedoch verschiedenen „Generationen“ zugeordnet
    - „Studienbeginn WS“ einerseits und „Studienbeginn SS“ anderseits
    - SP1 im „2. Fachsemester“ || SP2 im „3. Fachsemester“
  - der Stoff von SP2 ist „kausal abhängig“ vom Stoff von SP1 (vgl. S. 11)

⇒ von der Belegung beider Hälften im selben Semester wird abgeraten

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	×	×

Als 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  $\rightsquigarrow$  Dienstprogramme
  - im Großen erfahren  $\rightsquigarrow$  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

**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** → Grundlagen der Programmierung

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

## Wünschenswert, jedoch nicht zwingend erforderlich...

### **Technische Informatik** → 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 42

SP1	Montag	12:15 – 13:45	H4
SP2	Donnerstag	10:15 – 11:45	H8

## SP1-Zusatztermin

Dienstag, 16.10.2012  
10:15 – 11:45 H15

## Ausfalltermine

SP2     • Allerheiligen (01.11.)

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

- ① Austeilung von Folienkopien vor der Vorlesung
- ② [www4.cs.fau.de/Lehre/WS12/V\\_SP](http://www4.cs.fau.de/Lehre/WS12/V_SP)

## Fachbegriffe der Informatik (Deutsch ↔ Englisch)

- [www.babylonia.org.uk](http://www.babylonia.org.uk)
- [www.inf.fu-berlin.de/inst/ag-ss/montagswort](http://www.inf.fu-berlin.de/inst/ag-ss/montagswort)
- [www.aktionlebendigesdeutsch.de](http://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**<sup>1</sup> über die Startseite von SP

- die Periode der Übungswoche startet mit zweiter Vorlesungswoche
  - [www4.cs.fau.de/Lehre/WS12/V\\_SP1](http://www4.cs.fau.de/Lehre/WS12/V_SP1)
- Übungsplätze werden FCFS<sup>2</sup> 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

---

<sup>1</sup>Abk. für Webanmeldefrickelformular Enterprise Logic

<sup>2</sup>Abk. 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.*

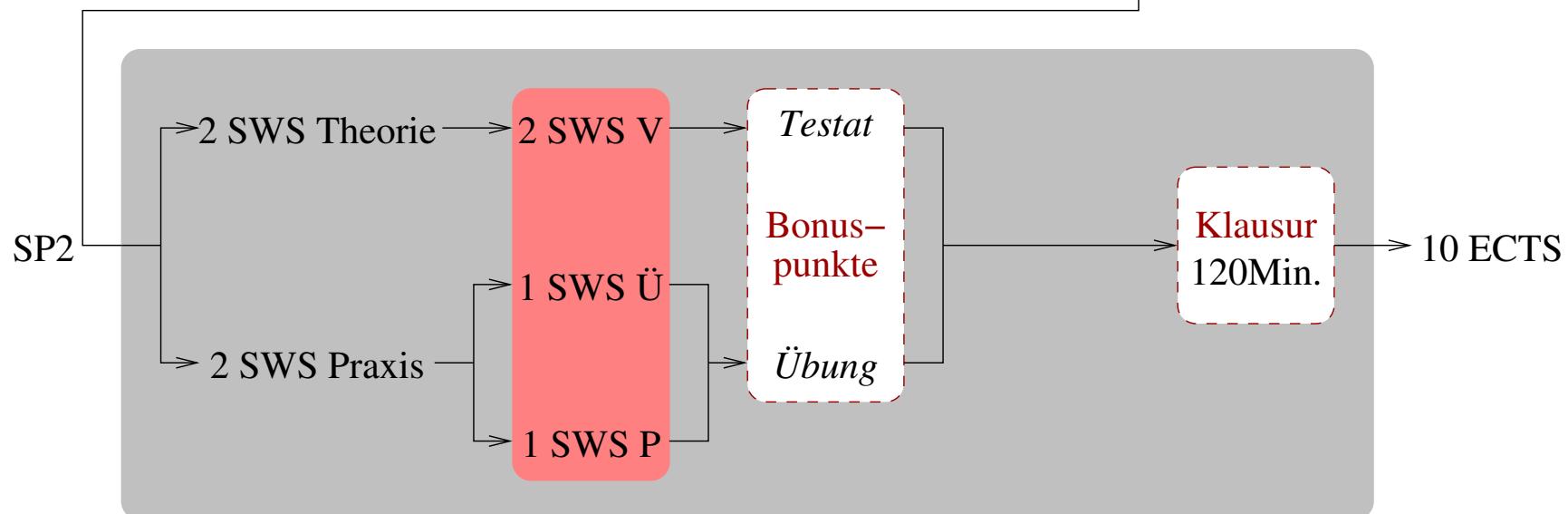
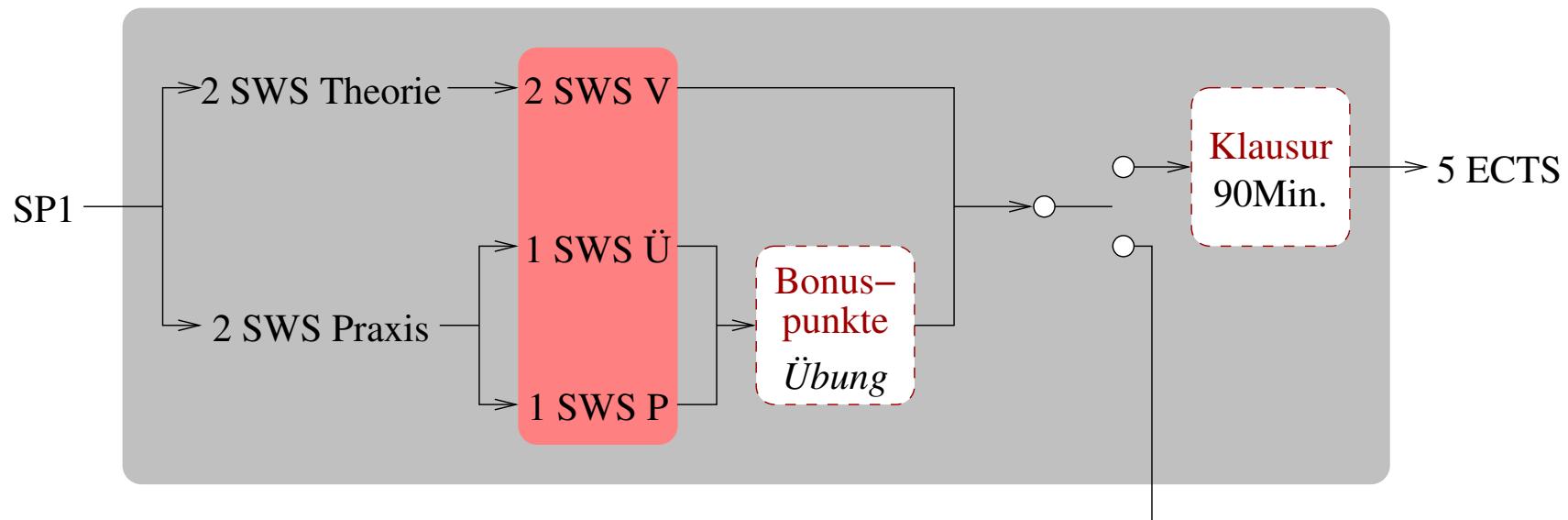
*Wer, wie, was?*

*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

- 50 % der Punkte aus „Übungsaufgaben + Miniklausur“ müssen erreicht sein
- 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 (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. 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

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

## Tutoren

- |                     |                         |
|---------------------|-------------------------|
| • Hans-Peter Deifel | • Stefan Reif           |
| • Stefan Hengelein  | • Simon Ruderich        |
| • Demian Kellermann | • Tobias Sammet         |
| • Johannes Knoedtel | • Tobias Scharpff       |
| • Philip Kranz      | • Florian Schmaus       |
| • Daniel Laffling   | • Markus Siko           |
| • Florian Meier     | • Matthias Utzschneider |
| • Dominik Paulus    |                         |

Fragen...

42