

Systemprogrammierung

Grundlagen von Betriebssystemen

Teil A – I. Organisation

18. April 2024

Jürgen Kleinöder
Rüdiger Kapitza

(© Wolfgang Schröder-Preikschat, Rüdiger Kapitza)



Lehrstuhl für Informatik 4
Systemsoftware



Friedrich-Alexander-Universität
Technische Fakultät

Dozenten

Lehrstuhl für Systemsoftware



Jürgen Kleinöder



Rüdiger Kapitza

SP

A – 1 / 2

Agenda

Einleitung

Konzept

Lehrkanon

Lehrziele

Vorkenntnisse

Arrangement

Veranstaltungsbetrieb

Leistungsnachweise

Ausklang

SP

Einleitung

A – 1 / 3

Gliederung

Einleitung

Konzept

Lehrkanon

Lehrziele

Vorkenntnisse

Arrangement

Veranstaltungsbetrieb

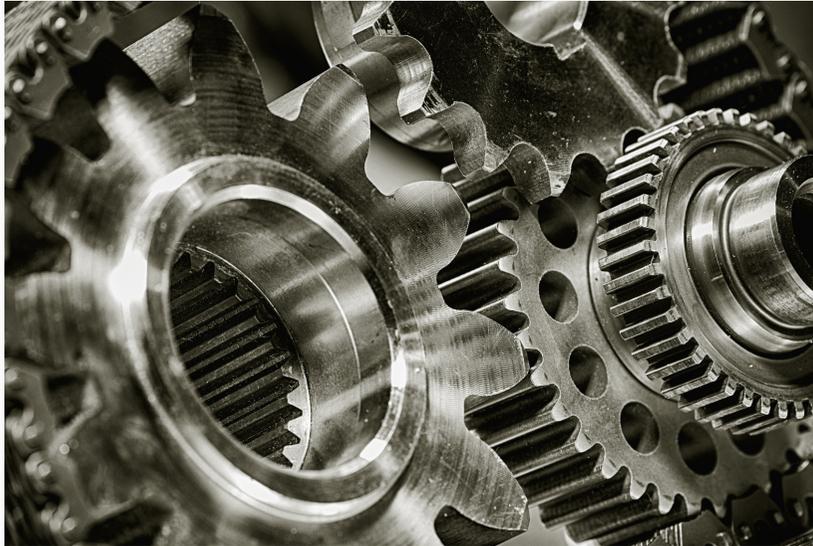
Leistungsnachweise

Ausklang

SP

Einleitung

A – 1 / 4



Quelle: fotalia.com

SP

Einleitung

A-1/5

Definition (Systemprogrammierung)

Erstellen von Softwareprogrammen, die Teile eines Betriebssystems sind beziehungsweise mit einem Betriebssystem direkt interagieren oder die Hardware (genauer: Zentraleinheit^a und Peripherie^b) eines Rechensystems betreiben müssen.

^acentral processing unit (CPU), ein-/mehrfach, ein-, mehr- oder vielkernig.

^bGeräte zur Ein-/Ausgabe oder Steuerung/Regelung „externer Prozesse“.

Auch schon mal zwischen zwei Stühlen sitzend:

- **Anwendungssoftware** („oben“) einerseits
 - ermöglichen, unterstützen, nicht entgegenwirken
- **Plattformsysteme** („unten“) andererseits
 - anwendungsspezifisch verfügbar
 - problemorientiert betreiben, bedingt verbergen



Quelle: arcadja.com, Franz Kott

SP

Einleitung

A-1/6



Quelle: <https://www.bell-labs.com/user/dmr/www>

SP

Einleitung

A-1/7

Einleitung

Konzept

Lehrkanon

Lehrziele

Vorkenntnisse

Arrangement

Veranstaltungsbetrieb

Leistungsnachweise

Ausklang

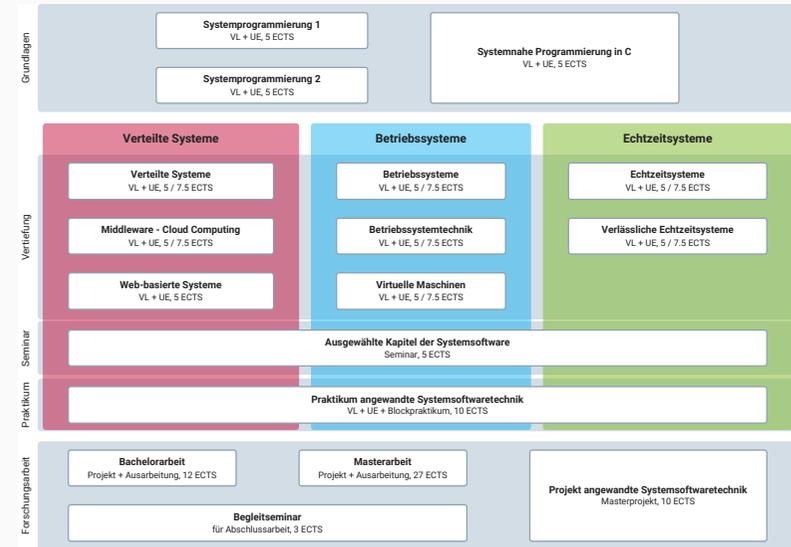
SP

Konzept

A-1/8

Konzept

Lehrkanon



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

Systemprogrammierung (SP) ~ geteiltes Modul

↪ Systemprogrammierung I (SP1) ↪ Teile A und B 5 ECTS

↪ Systemprogrammierung II (SP2) ↪ Teil C 5 ECTS

- SP1 geht in die **Breite**, liefert einen funktionalen Überblick
- SP2 geht in die **Tiefe**, behandelt ausgewählte Funktionen im Detail
- beide Hälften sind Grundlage vor allem der „Betriebssysteme“-Säule

Grundlagen der Systemprogrammierung (GSP)

↪ Systemprogrammierung I (SP1) 5 ECTS

- Export für spezifische Studiengänge

Studiengänge und Zuordnung

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

■ **Alternative** zu Systemnahe Programmierung in C (SPiC):

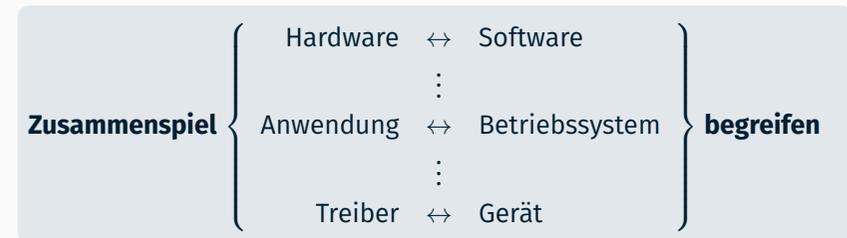
Abschluss	Studiengang	SP1	SP2
Bachelor	Mathematik, Nebenfach Informatik	×	
	Technomathematik	×	

Konzept

Lehrziele

Lernziele

- Vorgänge in Rechensystemen **ganzheitlich** verstehen



- imperative Systemprogrammierung (in C) in Grundzügen kennenlernen
 - im Kleinen für **Dienstprogramme** praktizieren
 - im Großen durch **Betriebssysteme** erfahren
- Beziehungen zwischen funktionalen und nicht-funktionalen Systemmerkmalen erfassen



Quelle: fotalia.com

Lehrveranstaltungsformen

- **Vorlesung** — Detaillierte Behandlung des Lehrstoffs
 - Organisation (der Systemsoftware) von Rechensystemen
 - Grundlagen von Betriebssystemen
 - maschinennahe Programme
- **Übung** — Vertiefung, Aufgabenbesprechung, 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 ↔ Prozess

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

Konzept

Vorkenntnisse

Voraussetzungen zum Verständnis des Lehrstoffs

- obligatorisch: **Grundlagen der Programmierung** ↔ AuD bzw. GdP
 - Datentypen, Kontrollkonstrukte, Prozeduren
 - statische und dynamische Datenstrukturen
 - „Programmierung im Kleinen“↔ vor allem für die Übung, weniger für die Vorlesung
- wünschenswert: **Technische Informatik** ↔ GTI, etwas GRa
 - „Von-Neumann-Architektur“
 - Operationsbefehle, Befehlsoperanden, Adressierungsarten
 - Unterbrechungssteuerung (Pegel kontra Flanke)
 - Assemblerprogrammierung
 - Pseudo- und Maschinenbefehle (IA32)
 - Binär-, Oktal-, Hexadezimalcode
 - CPU, DMA, FPU, IRQ, MCU, MMU, NMI, PIC, TLB
- aktuelle Sekundärliteratur
 - Heinrich Müller und Frank Weichert [2]: Vorkurs Informatik (aus GdP)
 - William Stallings [3]: Computer Organization and Architecture

SP

Konzept

A - 1 / 15

Abhängigkeiten zwischen den Vorlesungsteilen

Systemprogrammierung I

- **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 II

- **Teil C** ■ setzt Kenntnisse dieser Operationsprinzipien voraus
 - vermittelt **interne Funktionsweisen** von Betriebssystemen

- Erlangung der benötigten Kenntnisse:

- durch Vorlesungsteilnahme
 - empfohlene sequentielle Belegung der Vorlesungsteile
- durch Lehrbuchlektüre, aus anderen Lehrveranstaltungen, ...

SP

Konzept

A - 1 / 16

Gliederung

Einleitung

Konzept

Lehrkanon

Lehrziele

Vorkenntnisse

Arrangement

Veranstaltungsbetrieb

Leistungsnachweise

Ausklang

SP

Arrangement

A - 1 / 17

Arrangement

Veranstaltungsbetrieb

Unterrichtstermine und -sprache

- Vorlesungs-, Übungs- und Rechnerzeiten:
 - auf `sys.cs.fau.de` dem Reiter „Lehre“ folgen
 - Sondertermine am Semesteranfang für den *Crash*-Kurs über C

- Unterrichtssprache:



- Vorlesung und Übung
- Fachbegriffe
- Sachwortverzeichnis (in Arbeit und Überarbeitung)
 - `www4.cs.fau.de/~wosch/glossar.pdf`

Vorlesung

Kumulatives Lernen

- Aneignung von neuem Wissen
 - selbständig die jeweils nächste Vorlesung vorbereiten
 - an der Präsentation teilnehmen, ihr zuhören, Fragen stellen
 - behandelte Themen untereinander diskutieren und nachbereiten
- mit bisherigem/anderem/zukünftigen Wissen in Beziehung bringen:
 - GdP** ■ Grundlagen der Programmierung in einer **Hochsprache**
 - PFP** ■ Grundlagen der parallelen Programmierung
 - GRA** ■ Rechnerorganisation oder -architektur
 - Grundlagen der Programmierung in **Assemblersprache**
- im Hörsaal präsentiertes Lehrmaterial: **Vorlesungsfolien**
 - stehen animiert und in Handzettelform zur Verfügung
 - PDF: auf `sys.cs.fau.de` dem Reiter „Lehre“ folgen
 - Anzahl und „Füllungsichte“ sind bewusst eher hoch gehalten:
 - i obligatorischer und optionaler (Anhang) Vorlesungsstoff
 - ii schriftlich fixierte Gedankenstränge als Hilfe zur Nachbearbeitung
 - Anhänge und **ergänzende Materialien** sind keine Klausuraufgaben

Übung

Lernen durch Handeln

- Wissen durch **direkte Erfahrung** vertiefen

Tugendhaftes Verhalten und fachliches Können wird weniger durch einfache Belehrung als durch praktisches Nachmachen, Üben, Anwenden erlernt. (Aristoteles [1])

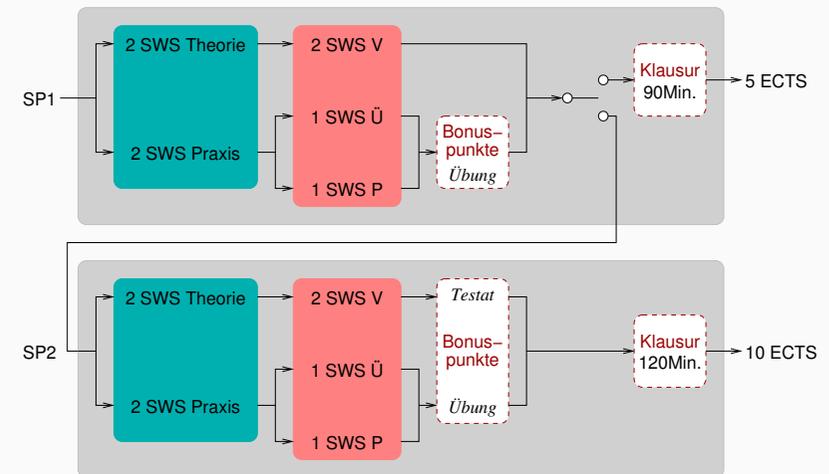
- Diskussion der Übungsaufgaben, Lösungsansätze ausarbeiten
- Vorlesungsstoff festigen, offene Fragen klären
- **Tafelübung** unter Anleitung einer/s Übungsleiterin/s
 - Anmeldung durch **WAFFEL**¹ (URL siehe Webseite von SP)
 - Übungsaufgaben sind in Gruppen zu bearbeiten: Kannvorschrift
 - ist abhängig von der Teilnehmeranzahl
 - Gruppenpartner müssen in derselben Übung sein
- **Rechnerarbeit** in Eigenverantwortung
 - ohne Anmeldung, reservierte Arbeitsplätze stehen zur Verfügung
 - bei Fragen sich an die Übungsleiter/innen von SP wenden

¹Abk. *Webanmeldefrickelformular Enterprise Logic*

Arrangement

Leistungsnachweise

Studien- und Prüfungsleistungen



SP

Arrangement

A - 1 / 21

Kür

- **Übungsaufgaben:** 6 (SP1) + 5 (SP2) Programmieraufgaben
 - abgegebene Programme werden korrigiert und mit Punkten bewertet
 - unzureichende Erklärung der Lösung ergibt 0 Punkte
 - Nichtanwesenheit impliziert unzureichende Erklärung
- ein **Antestat**² (auch: „Miniklausur“) zum Aufwärmen für SP2
 - geprüft wird Stoff von Vorlesung und Übung, 30 Minuten
 - Fragen zu Teil A und Teil B der Vorlesung
 - Trockenaufgabe als Lückentest in der Programmiersprache C
 - mit Aufgabenanteilen als Mehrfachauswahl (*multiple choice*)

Notenbonus für die Klausur (auch: „Maxiklausur“)

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

²Allgemein eine mündliche oder schriftliche Prüfung in naturwissenschaftlichen Studienfächern am Anfang eines Semesters. Schriftlich ausgeführt im Fall von SP.

SP

Arrangement

A - 1 / 22

Kür und Pflicht

- **Notenbonus** nur auf Basis der Übungen **des letzten SP-Moduls**
 - beeinflusst die Punkte-Notenskala der Klausur nicht, er wird allerdings 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 regulär erworbenen Klausurpunkten erfolgen
- **Klausur:** Termin noch offen, Anfang vorlesungsfreie Zeit
 - GSP** ■ Struktur analog Testat (S. 27), 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“

- ⇨ Besuch der Vorlesung, zuhören und Fragen stellen
- ⇨ Teilnahme an den Tafelübungen, Übungsaufgaben bearbeiten
- ⇨ Im Team entwickeln, aber selbst zum Laufen bringen

SP

Arrangement

A - 1 / 23

Gliederung

Einleitung

Konzept

Lehrkanon

Lehrziele

Vorkenntnisse

Arrangement

Veranstaltungsbetrieb

Leistungsnachweise

Ausklang

SP

Ausklang

A - 1 / 24

Ausklang

Kontakt

sys.cs.fau.de/*

Dozenten

- Jürgen Kleinöder (~jklein)
- Rüdiger Kapitza (~rrkapitz)

Mitarbeiter

- Luis Gerhorst (~gerhorst)
- Thomas Preisner (~preisner)

Tutoren

- Ferdinand Schober, Dana Elsmann, Christian Halder, Philip Kaludercic, Johannes Konrad, Stefan Schmitt, Julian Zboril, Frederik Zeug, Hannes Schulze, Lukas Bilstein



www.augsburger-puppenkiste.de

SP

Ausklang

A - 1 / 25

Fragen und Hinweise



Quelle: qmediasolutions.com

■ Hinweise

- Anmeldung für die Übungen heute ab 16:00 Uhr via Waffel

SP

Ausklang

A - 1 / 26

Ausklang

Bibliographie

Literaturverzeichnis (1)

- [1] ARISTOTELES:
Nikomachische Ethik.
c. 334 BC
- [2] MÜLLER, H. ; WEICHERT, F. :
Vorkurs Informatik – Der Einstieg ins Informatikstudium.
Springer Fachmedien Wiesbaden GmbH, 2017. –
392 S. –
ISBN 978-3-658-16141-5
- [3] STALLINGS, W. :
Computer Organization and Architecture.
Pearson, 2021. –
896 S. –
ISBN 9781292420103