

Überblick

Lehrveranstaltungskonzept

- Einordnung
- Studiengänge
- Lernziele und Lehrinhalte
- Voraussetzungen
- Ablauf der Lehrveranstaltung
- Leistungsnachweise
- Kontakt

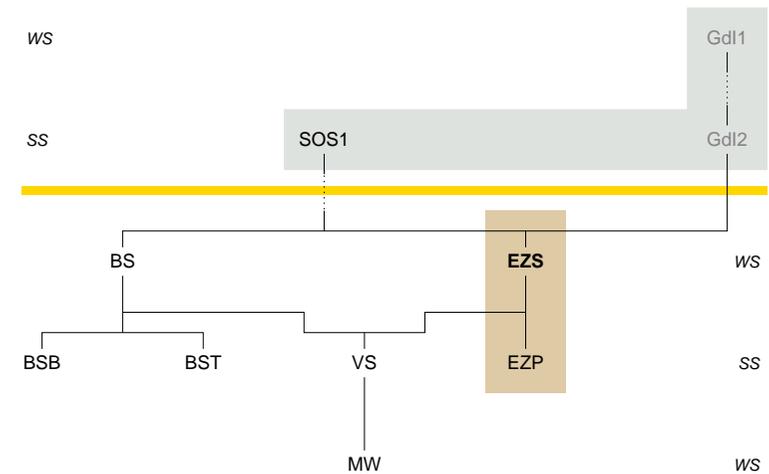
Integrierte Lehrveranstaltung

$$\text{Termine} \left\{ \begin{array}{l} \text{Vorlesung} \quad 1 \\ \text{Übung} \quad 1 \\ \text{Rechner} \quad 1 \end{array} \right\} \equiv 3 \times 1,5 = 4,5 \text{ Zeitstunden wöchentlich}$$

Vor-/Nacharbeit

- ▶ N Stunden wöchentlich: $0 \leq N \leq (163,5 - X)$
- ▶ $X \ll 163,5$ ist das Zeitstundenäquivalent anderer „Pflichten“

Lehrprofil von I4



Fakultative Vertiefung in Informatik

Diplom

- ▶ Informatik, I & K
- ▶ Mechatronik, Maschinenbau
- ▶ Technomathematik, Wirtschaftsinformatik

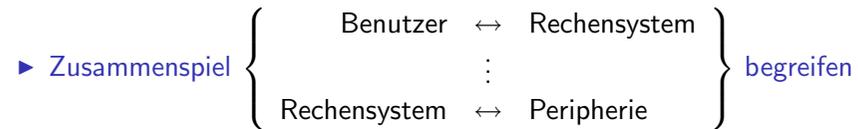
Bachelor

- ▶ Computational Engineering (CE)

Echtzeitsysteme findet man nicht nur in den mehr technisch ausgelegten Bereichen. Entscheidungsprozesse der Wirtschaft sind verschiedentlich ebenfalls echtzeitabhängig.

Lernziele

Rechensystem und seine Umgebung als **Ganzes** verstehen:



Objektorientierte **Echtzeitprogrammierung** (in Grundzügen) erleben

- ▶ betriebssystemnah praktizieren

☞ Vermittlung der Grundlagen von echtzeitfähigen Softwaresystemen

Lehrinhalte

Vorlesung: Vorstellung und detaillierte Behandlung des Lehrstoffs

- ▶ Grundlagen von Echtzeitsystemen
 - ▶ zeit- und ereignisgesteuerte Systeme
 - ▶ periodische und sporadische Aufgaben (engl. *tasks*)
 - ▶ Einplanung und Koordination
- ▶ Echtzeitbetriebssysteme

Übung: Vertiefung, Besprechung der Übungsaufgaben, Tafelübungen

- ▶ Systemprogrammierung in C/C++
- ▶ echtzeitfähige Systemprogramme

Erforderliche Grundkenntnisse

Grundlagen der Systemprogrammierung: **Softwaresysteme**

- ▶ systemnahe Programmierung in C
 - ▶ asynchrone Programmunterbrechungen, Aktivitätsträger, Prozesse
 - ▶ koordinierte Ausführung nebenläufiger Programme
- ▶ Grundzüge von Betriebssystemen

Grundlagen der Rechnerorganisation: **Technische Informatik**

- ▶ maschinennahe Programmierung
 - ▶ Unterbrechungssteuerung (Pegel kontra Flanke)
 - ▶ Assemblerprogrammierung
- ▶ CPU, DMA, IRQ, MCU, NMI, PIC

☞ **Betriebssystemkenntnisse** sind fördernd, daher erwünscht und hilfreich

Vorlesungsbetrieb und Lehrmaterialien

Vorlesungstermine:

- ▶ Montag, 14:15–15:45, H4

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

- ▶ www4.informatik.uni-erlangen.de/Lehre/WS06/V_EZS
- ▶ **kein Skript**, bloß die Folien zum Vorlesungs- und Übungsstoff

Fachbegriffe der Informatik (Deutsch ↔ Englisch)

- ▶ <http://www.babylonia.ork.uk>

Ergänzende Literatur

- ▶ H. Kopetz. *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Kluwer Academic Publishers, 1997. ISBN 0-7923-9894-7.
- ▶ J. W. S. Liu. *Real-Time Systems*. Prentice-Hall, Inc., 2000. ISBN 0-13-099651-3.

- ▶ W. Schröder-Preikschat. *Softwaresysteme 1*. www4.informatik.uni-erlangen.de/Lehre/SS06/V_S0S1
- ▶ W. Schröder-Preikschat. *Echtzeitbetriebssysteme*. In *Software Engineering für Eingebettete Systeme*, Hrsg.: P. Liggesmeyer und D. Rombach, S. 343–376. Elsevier, Spektrum Akademischer Verlag, 2005.

Übungsbetrieb

Anmeldung bevorzugt über WAS (Web-Anmeldesystem):

- ▶ www.was.dienste.uni-erlangen.de/content
- ▶ Beginn des Übungsbetriebs: ab sofort...

Tafel- und Rechnerübungen in wöchentlichen Intervallen

- ▶ Trockenübung ohne praktische Umsetzung ist wenig ergiebig
- ▶ praktische Umsetzung ohne Trockenübung ist zu beschwerlich

☞ Kontinuität und aktive Mitarbeit ist der Schlüssel zum Erfolg

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

☞ „Wieso, weshalb, warum? Wer nicht fragt, bleibt dumm!“

Leistungskontrolle

unbenoteter Schein: obligatorisch für Rücksprache zum benoteten Schein

- ▶ erfolgreiche Bearbeitung aller Übungsaufgaben

unbenotete Abfrage: Bestandteil der Tafelübungen

- ▶ erfolgreiche Erklärung/Behandlung eines Themenkomplexes
 - ▶ z.B. einer Übungsaufgabe oder Fragestellung der Vorlesung
- ▶ zufällige Auswahl, Drankommen ist jedoch garantiert...

Prüfung: mündlich, studienbegleitend

- ▶ Themenschwerpunkte je nach Teilnahme am Übungsbetrieb
 - vorlesungslastig bei bestandenem Schein/bestandener Abfrage
 - übungslastig sonst
- ▶ den unbenoteten Schein zu machen, wird dringend empfohlen

Übungsscheinkriterien

Abgabefristen von Übungsaufgaben sind **strikt** einzuhalten

- ▶ elektronisch gestützter, semi-automatischer Abgabevorgang:
 - ▶ prüft die Aufgabenlösungen auf Plausibilität
 - ▶ testet auf Abschreiben
- ▶ stichprobenartig werden einzelne Aufgaben genauer durchkorrigiert

Scheinvergabe nur bei ausreichender Bearbeitung der Übungsaufgaben

- ▶ Plausibilitätsprüfung und Abschreibetest müssen OK signalisiert haben
- ▶ die Stichprobenkorrektur muss ein positives Ergebnis geliefert haben
- ▶ ggf. entscheidet eine persönliche Rücksprache über Erfolg/Misserfolg

www4.informatik.uni-erlangen.de/*

Dozent

- ▶ Wolfgang Schröder-Preikschat (~wosch)

Mitarbeiter

- ▶ Fabian Scheler (~scheler)

Rücksprache und Prüfung

Rücksprache für einen benoteten Übungsschein, 30 Minuten

- ▶ Thema ist Stoff der Übung aber auch der Vorlesung
- ▶ Einzelgespräch, Termin direkt mit dem Dozenten vereinbaren

Prüfung: mündlich, 15–30 Minuten

- ▶ Thema ist Stoff der Vorlesung aber auch der Übung
- ▶ Einzelprüfung, Termin direkt mit dem Dozenten vereinbaren

aktive Mitarbeit machen Rücksprache bzw. Prüfung „leicht“

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

Fragen...

?