

Echtzeitsysteme 2

Analyse, Entwurf, Implementierung

Fabian Scheler
Peter Ulbrich
Michael Stilkerich
Wolfgang Schröder-Preikschat

Lehrstuhl für Informatik IV
Verteilte Systeme und Betriebssysteme
Friedrich-Alexander Universität Erlangen-Nürnberg

<http://www4.cs.fau.de/~{scheler,ulbrich,mike,wosch}>
{scheler,ulbrich,mike,wosch}@cs.fau.de



1

Übersicht

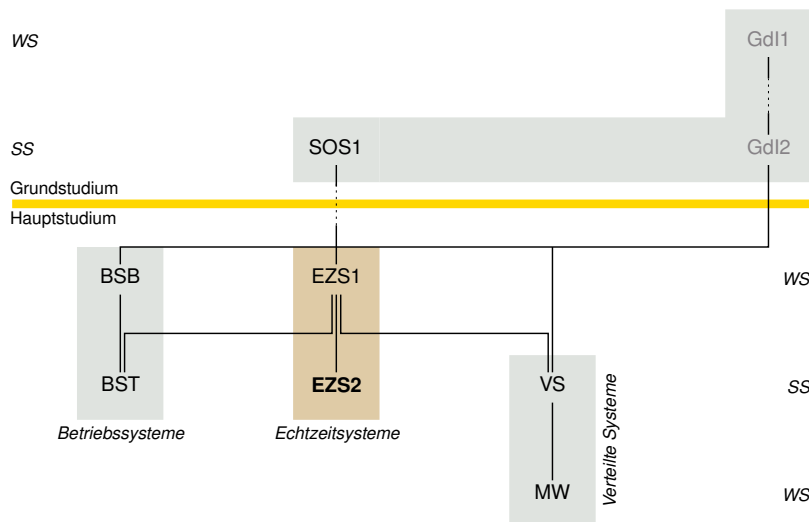
- Einordnung
- Aufbau/Lehrform
- Voraussetzungen
- Leistungsnachweis
- Literatur
- Lernziele & Lehrinhalt
- Experimente



© {scheler,ulbrich,mike,wosch}@cs.fau.de - EZS2 (SS 2007)

2

Lehrprofil von I4



© {scheler,ulbrich,mike,wosch}@cs.fau.de - EZS2 (SS 2007)

3

Integrierte Lehrveranstaltung

- Termine
 - Vorlesung (2 SWS) + Übung (2 SWS)
= 4 SWS (3 Stunden/Woche)
- Vor-/Nacharbeit
 - N Stunden/Woche: $0 \leq N \leq (165 - X)$
 - $X < 165$: Zeitäquivalent anderer Pflichten
- Folien/Handout
 - www4.informatik.uni-erlangen.de/Lehre/SS08/V_EZS2
 - **kein Skript** – Folien zu Vorlesung/Übung



© {scheler,ulbrich,mike,wosch}@cs.fau.de - EZS2 (SS 2007)

4

Ablauf/Material

- **Modus**
 - wöchentlicher Wechsel (2 SWS):
 - Vorlesung = inhaltliche Einführung zu den Projektphasen
 - Übung = Statusbericht der Gruppen zu den Projekten
 - „restlichen“ 2 SWS:
 - Bearbeitung der Projekte
- **Ort und Termin???**
 - Abstimmung
 - Möglichkeit 1: Dienstag 12:00 – 14:00 (0.031)
 - Möglichkeit 2: Vorschlag ...
 - Möglichkeit 3: Vorschlag ...



Übungsbetrieb

- **Anmeldung**
 - über das Waffel
 - <http://waffel.informatik.uni-erlangen.de/signup/?univisid=20669384>
 - Begin: ab sofort ...
- **Projektarbeit**
 - ≤ 3 Mitarbeiter je Gruppe (Steuerung und Bedienpult extra)
 - Einzelkämpferdasein ist nicht zielführend
 - 14-tägige Statusberichte
- **Rechnerübung:**
 - Manlobbi (0.058), West-Side Labs (0.055)
 - freies Arbeiten, Betreuung „on demand“
- **Kontinuität und aktive Mitarbeit als Schlüssel zum Erfolg!**



Voraussetzungen

- **Grundlagen von Echtzeitbetriebssystem: EZS 1**
 - zeit- und ereignisgesteuerte Systeme
 - periodische, aperiodische und sporadische Aufgaben
 - Einplanung und Koordination
- **Systemprogrammierung in C/C++ und Assembler**
 - maschinen- d.h. hardwarenahe Programmierung
- **Betriebssystemkenntnisse sind fördernd, daher erwünscht und hilfreich**



Leistungsnachweis

- **unbenoteter Schein**
 - erfolgreiche Bearbeitung des Projekts
 - Statusberichte / Abschlusspräsentation
 - Poster
- **benoteter Schein**
 - unbenoteter Schein
 - Bewertung der erbrachten Leistungen



Kontakt

- Fabian Scheler
www4.informatik.uni-erlangen.de/~scheler
scheler@informatik.uni-erlangen.de
- Michael Stilkerich
www4.informatik.uni-erlangen.de/~mike
stilkerich@informatik.uni-erlangen.de
- Peter Ulbrich
www4.informatik.uni-erlangen.de/~ulbrich
ulbrich@informatik.uni-erlangen.de
- auf folgenden Wegen
 - persönlich
 - e-Mail (**Mailingliste!**)
 - ICQ, Jabber, ...



Ergänzende Literatur

- *Hermann Kopetz.*
Real-Time Systems: Design Principles for Distributed Embedded Applications.
Kluwer Academic Publishers, 1997.
- *Jane W. S. Liu.*
Real-Time Systems.
Prentice Hall PTR, Upper Saddle River, NJ, USA, 2000.
- *Jim Cooling.*
Software Engineering for Real-Time Systems.
Addison-Wesley, 2003.
- *Phillip A. Laplante.*
Real-Time Systems Design and Analysis.
Wiley, third edition, 2004.
- *W. Schröder-Preikschat.*
Echtzeitsysteme.
www4.informatik.uni-erlangen.de/Lehre/WS07/V_EZS



Lernziele

- Echtzeitprogrammierung in Teams
- Echtzeitanwendung entwickeln
 - Anforderungen/Gegebenheiten analysieren und spezifizieren
 - Echtzeitsystem problemorientiert implementieren
 - Experimente aufbauen und durchführen
- Vertiefung von Grundlagenwissen durch experimentelles Arbeiten



Lehrinhalte - Projektphasen

1. Analyse des physikalischen Objekts
 - Erfassen physikalischer Zusammenhänge
 - Wo existieren Deadlines?
 - Wie sieht das physikalische Modell aus, wo abstrahiert man?
2. Entwicklungsumgebung
 - Was für ein
 - Prozessor
 - Compiler
 - Betriebssystemwird verwendet?
 - Wie funktioniert das Debuggen?
 - Gibt es irgendwelche Besonderheiten?



Lehrinhalte - Projektphasen

3. Komponenten

- Welche Komponenten existieren in meinem System?
- Was tun diese Komponenten? (Spezifikation)
- Wie interagieren diese Komponenten? (Spezifikation)
- Implementierung der Komponenten?

4. Testen: Komponenten

- einfache Testumgebung:
 - übersetzt und bindet Testfall
 - führt Testfall aus
 - überprüft das Ergebnis
- Spezifikation der Testfälle
- Getestet wird: Funktionalität, WCET (und Speicherplatz)



Lehrinhalte - Projektphasen

5. Komposition

- Abbildung: Komponenten → Ereignisbehandlungen
- Planung:
 - Prioritätenvergabe
 - statische Ablaufpläne
- Planbarkeitsanalyse

6. Integration

- Implementierung des kompletten Systems
- Abbildung auf Betriebssystemstrukturen



Lehrinhalte - Projektphasen

7. Testen: komplettes System → Akzeptanztest

- Ausprobieren
- Debuggen

8. Präsentation

- Zusammenfassung des Projekts

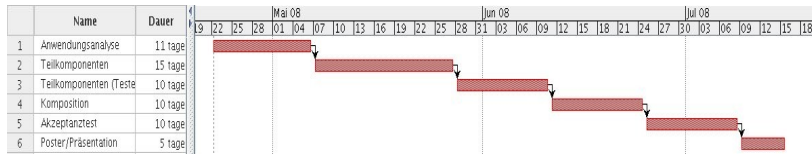


Lehrinhalt - Struktur

- zu jeder Phase gibt es einen Zwischenbericht
 - ca. 10-minütige Präsentation
 - Zu welchen Ergebnissen ist man gekommen?
 - Welche Designentscheidungen wurden getroffen? Warum?
 - Welche Konsequenzen haben diese Entscheidungen?
- Abschlussbericht / Abschlusspräsentation
 - ca. 45-minütige Präsentation / Demonstration
 - Überblick über das Projekt
 - Poster
 - setzt sich aus Zwischenberichten zusammen



Zeitplanung



Phase	Start	Ende
1 Anwendungsanalyse	22.04.2008	06.05.2008
2 Teilkomponenten (Entwurf & Implementierung)	07.05.2008	27.05.2008
3 Teilkomponenten (Testen & WCET)	28.05.2008	10.06.2008
4 Komposition	11.06.2008	24.06.2008
5 Akzeptanztest	25.06.2008	08.07.2008
6 Poster/Präsentation	09.07.2008	15.07.2008

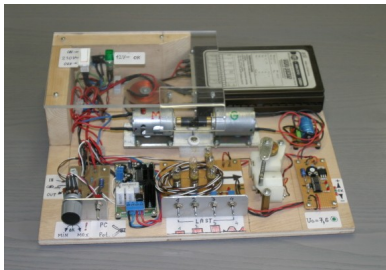
Experiment 1: Hau den Lukas

- Aufgabe 1: den Eisenkern ...
 - anheben und abbremsen
 - schrittweise anheben/fallen lassen
 - gebremst fallen lassen
 - pendeln lassen
- Aufgabe 2: Bedienpult
 - Ablauf wechseln
 - neuen Ablauf programmieren
 - Monitoring/Überwachung
 - Not-Aus



Experiment 2: Generator

- Aufgabenstellung
 - Regelung einer Motor-Generator-Strecke
 - Implementierung
 - des Regelsystems
 - eines Beobachters
 - einer *naiven* Regelung
 - Vergleich der implementierten Regler



Experiment 3: Quadkopter

- Aufgabenstellung
 - Balancierung einer Wippe mit 2 Rotoren
 - Motoren hoch-/runterfahren
 - Neigung der Wippe verändern
 - Implementierung des
 - Regelkreises
 - und des Beobachters
 - digitale Filterung der Messwerte



Experiment 4: Eisenbahn

■ Aufgabe 1: Zug-/Weichensteuerung

- (un)veränderliches Streckennetz
- Weichenstellung
- Signalanlagen

■ Aufgabe 2: Bedienpult

- Verwaltung von
 - Zügen
 - Gleisabschnitten
 - Routen
- Not-Aus

