

# Echtzeitsysteme

## Übungen zur Vorlesung

Fabian Scheler, Peter Ulbrich, Niko Böhm

Friedrich-Alexander-Universität Erlangen-Nürnberg  
Lehrstuhl Informatik 4 (Verteilte Systeme und Betriebssysteme)  
[www4.informatik.uni-erlangen.de](http://www4.informatik.uni-erlangen.de)

20. Oktober 2008

- **Vorlesung:**

**Prof. Dr. Wolfgang Schröder-Preikschat**

(wosch@informatik.uni-erlangen.de,  
www4.informatik.uni-erlangen.de/~wosch)

- **Übung:**

**Fabian Scheler**

(scheler@informatik.uni-erlangen.de,  
www4.informatik.uni-erlangen.de/~scheler)

**Peter Ulbrich**

(ulbrich@informatik.uni-erlangen.de,  
www4.informatik.uni-erlangen.de/~ulbrich)

**Niko Böhm**

(boehm@informatik.uni-erlangen.de,  
www4.informatik.uni-erlangen.de/~boehm)

# Wann und wo?

- Vorlesung: Montag 14:15 - 15:45 H4
- Übung:
  - Tafelübung:

Montag	10:00 - 12:00	0.031
Dienstag	14:00 - 16:00	00.156
Donnerstag	12:00 - 14:00	00.156
  - Rechnerübung
    - Donnerstag 10:00 - 12:00 02.151a NON-Graphik
    - nach Vereinbarung in der manlobbi

## unbenoteter Schein

- ausreichende Bearbeitung der Übungsaufgaben
- regelmäßige und aktive Teilnahme am Übungsbetrieb
- ggf. Rücksprache über die Scheinvergabe (falls eine oder mehrere Übungsaufgaben nicht ausreichend bearbeitet wurden)

## benoteter Schein

- unbenoteter Schein
- Rücksprache (Termin vereinbaren! - Mail)

## Prüfung

- mündliche Prüfung
- Inhalt ist Vorlesung **und** Übung
- Einzelprüfung (Termin vereinbaren! - Mail)

- Anmeldung über das Waffel:

`https://waffel.informatik.uni-erlangen.de/signup/?course=30`

- Folien, Hinweise, Aufgabenstellungen, Dokumentation ... zu Vorlesung und Übung:

- Im Internet

`www4.informatik.uni-erlangen.de/Lehre/WS08/V_EZS/`

- Im CIP-Pool

`/proj/i4ezs`

- Subversion URL

`https://www4.informatik.uni-erlangen.de:8088/i4ezs/<group_name>`

- Fragen an die Mailingliste

`i4ezs@immd4.informatik.uni-erlangen.de`

**Zielsetzung** Erweiterung eines Betriebssystemgerüsts zu einem Echtzeitbetriebssystem

**Themen** Typische Aufgabenstellungen, die in Echtzeitsystemen anfallen:

- Ablaufplanung
- Behandlung von periodischen und aperiodischen/sporadischen Ereignissen
- Fadensynchronisation

**Gruppen** Die Bearbeitung der Aufgaben erfolgt in 2er-Gruppen, jeweils ein Übungspartner implementiert

- die Aufgabenstellung
- entsprechende Testfälle
- Gruppeneinteilung per Mail an [scheler@informatik.uni-erlangen.de](mailto:scheler@informatik.uni-erlangen.de)

**Abgabe** Erstellung eines entsprechenden Subversion-Tags  
durch `make abgabe EXERCISE=aufgabe{1,2,3,4,5}`

Vorgaben sind Gerüste

- und sollen helfen, sich zu orientieren
- und keine starre Schablone

Schnittstellen sind **verbindlich!**

- Schedule-Table Scheduler
- Aperiodic Task Support
- Multi-Level-Queue Scheduler
- Thread Synchronization
- Deferrable Server

# Übungsaufgaben - Ablauf

- 1 Ausgabe der Vorgabe und Besprechung der Aufgabenstellung (immer Montag 16:30)
- 2 während der Bearbeitung
  - beliebig viele Abgaben möglich
  - Abgaben werden getestet und das Ergebnis per Mail mitgeteilt
  - Referenztestfälle werden **nicht** herausgegeben
- 3 Abgabe ist ca. 2,5 Wochen später (immer Freitags 12:00)
- 4 Abgaben werden gegen Referenztestfälle getestet
  - alle Testfälle erfolgreich  $\mapsto$  Aufgabe erfolgreich
  - Testfälle fehlgeschlagen  $\mapsto$  Fehlerbehebung
  - Herausgabe der Referenztestfälle
- 5 Abgabe wird 1 Woche später erneut getestet
  - sofern die Abgabe erneuert wurde
  - alle Testfälle erfolgreich  $\mapsto$  Aufgabe erfolgreich
  - Testfälle fehlgeschlagen  $\mapsto$  Aufgabe nicht erfolgreich
  - Test-Arena

- 1 Testen der Testfälle gegen eine Referenzimplementierung
  - Referenzimplementierung gilt als fehlerfrei
  - Sind die Testfälle *sinnvoll*?
- 2 Testen: jeder gegen jeden
  - jede Implementierung wird gegen jeweils alle Testfälle aller anderen Gruppen getestet
- 3 Ergebnis:
  - Gruppe abc findet n Fehler in der Abgabe der Gruppe xyz
  - Rangliste - Sortierung nach irgendeinem Kriterium, z.B.:  
*bei anderen gefundene Fehler – eigene Fehler*