

Konzept

Betriebssysteme, ©Wolfgang Schröder-Preikschat

Lernziele

- zu vertiefen, wie ein Betriebssystem funktioniert
 - was ein Betriebssystem *ist*, thematisierte Systemprogrammierung 1
- zu erfahren, welchen

<i>Schwerpunkt Nebenläufigkeit</i>

 dabei besitzt
 - Betriebssysteme als nebenläufige/parallele Systemprogramme auffassen
- zu ermöglichen, ein Betriebssystem von der Pike auf zu entwickeln

Lehrinhalte (1)

- Erscheinungsformen** Entstehungsgeschichte, Betriebsart, Zweck (2)
- Unterbrechungen** ... synchrone –, asynchrone –, partielle Interpretation ... (1)
- Synchronisation** blockierende –, nicht-blockierende –, wartefreie – (2)
- Programmfäden** Koroutinen, Prozessorstatus, Kontextwechsel (1)
- Prozesse** Zustand, Kontrollblock, Abfertigung (*dispatching*) (1)

Lehrinhalte (2)

- Einplanung (*CPU scheduling*)** lang-, mittel- und kurzfristige –, kooperative –, verdrängende –, deterministische –, probabilistische – (2)
- Kooperation** Semaphor, Lese-/Schreibprobleme, Monitor (2)
- Kommunikation** synchrone –, asynchrone –, aktive Nachrichten (1)
- Verklemmungen** ... Bedingungen, Vorbeugung, Vermeidung, Erkennung ... (1)
- Architekturen** Monolith, geschichtetes System, Mikrokern (1)

Einordnung

- Studiengänge
 - Informatik, Hauptstudium, ab 5. Semester (s.u.)
 - Wirtschaftsinformatik, ab 7. Semester (Wahl)
 - Computational Engineering, Bachelor-Studium, ab 5. Semester (Wahl)
- *Wahlpflicht* für „Betriebssysteme und Verteilte Systeme“ . . .
als $\left\{ \begin{array}{ll} \text{Prüfungsfach} & (8 \text{ SWS}) \\ \text{Schwerpunktfach} & (12 \text{ SWS}) \end{array} \right\} \text{Prüfung/Schein } (\{,un\} \text{benotet})$
- sehr empfohlen bei vorgesehener Mitarbeit in F&E-Vorhaben des Lehrstuhls

Voraussetzungen

- Rechnerorganisation, **Systemprogrammierung 1** [1]
- C/C++, Assembler
- sich an hardware-naher Programmierung erfreuen können
 - „Furchtlosigkeit“ vor nur schwer erkund- und fassbaren Sachverhalten
- ein gewisses Maß an Durchhaltevermögen

Organisation

- **integrierte Lehrveranstaltung**, $2 \times 1,5$ Stunden wöchentlich.....4 SWS
VL Vorstellung und detaillierte Behandlung des Lehrstoffs
Ü Vertiefung, Besprechung der Übungsaufgaben, Tafelübungen
- **Rechnerübung**, 1,5 Stunden wöchentlich.....0 SWS
– am Rechner betreutes Bearbeiten der Übungsaufgaben
- **Vor-/Nacharbeit**, N Stunden wöchentlich, $0 < N \leq 163,5$0 SWS

Leistungskontrolle

- **Schein**
 - *Ausstellung* bei erfolgreicher Bearbeitung aller Aufgaben
 - *Rücksprache* bei nicht-erfolgreicher Bearbeitung einer Aufgabe
- **benoteter Schein**..... *Voraussetzung*: Schein (s.o.)
 - abschließendes Gespräch („Scheinprüfung“) über den Übungsstoff
- **Prüfung** bei Wahl für „Betriebssysteme und Verteilte Systeme“
 - Inhalt ist Vorlesung- und Übungsstoff (d.h. die „Scheinaufgaben“)

Lehrkörper

- Wolfgang Schröder-Preikschat Professor
– <http://www4.informatik.uni-erlangen.de/~wosch>
- Olaf Spinczyk Assistent
– <http://www4.informatik.uni-erlangen.de/~os>
- Andreas Gal Mitarbeiter
– <http://www4.informatik.uni-erlangen.de/~gal>

Empfohlene Literatur

- [1] F. Hauck. Systemprogrammierung 1. <http://www4.informatik.uni-erlangen.de>, 2001. Vorlesungsskript.
- [2] R. G. Herrtwich and G. Hommel. *Kooperation und Konkurrenz — Nebenläufige, verteilte und echtzeitabhängige Programmsysteme*. Springer-Verlag, 1989. ISBN 3-540-51701-4.
- [3] H. Kopetz. *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Kluwer Academic Publishers, 1997. ISBN 0-7923-9894-7.
- [4] W. Schröder-Preikschat. *The Logical Design of Parallel Operating Systems*. Prentice Hall International, 1994. ISBN 0-13-183369-3.
- [5] A. Silberschatz and P. B. Galvin. *Operating System Concepts*. Addison-Wesley, 1994. ISBN 0-201-59292-4.
- [6] A. S. Tanenbaum. *Structured Computer Organization*. Prentice Hall, 1990.