

Modul für den Studiengang Informatik / Softwaretechnik

Modulbezeichnung	Betriebssysteme II
Kürzel für Stundenplan	BeSy2
Semester	4/5
Modulverantwortliche/r	Prof. Dr.-Ing. Krause
Dozent/in	Prof. Dr.-Ing. Krause
Sprache	Deutsch
Zuordnung zum Curriculum	INF, technisches Wahlpflichtfach
Lehrform / SWS	2 V mit integrierten Übungen, 2 P am Rechner
Arbeitsaufwand	V 32h plus die Hälfte für Vor-/Nachbereitung = 48h P 32h plus das Zweifache für Vor-/ Nachbereitung = 96h
Leistungspunkte	5
Voraussetzungen	Mathematik, Grundlagen der Informatik, Programmieren, Rechnerstrukturen, Betriebssysteme I
Lernziele / Kompetenzen	Aufbauend auf der Lehrveranstaltung Betriebssysteme I, vertieft diese Lehrveranstaltung die Kenntnisse über das Innenleben von Betriebssystemen, die Besonderheiten der Systemprogrammierung sowie die nebenläufige Programmierung in Linux und Windows Umgebungen. Die Studierenden entwickeln praktische Kompetenzen in der systemnahen Programmierung, insbesondere im Umgang mit den Mechanismen der Prozesssynchronisation und Prozesskommunikation. Die grundlegenden Konzepte wurden in Betriebssysteme I bereits erarbeitet, die tatsächliche praktische Anwendung erfordert jedoch systemabhängige Detailkenntnisse, die im Rahmen dieser LV praxisnah vermittelt werden.
Inhalt	s.u.
Studien-/ Prüfungsleistungen	Projektarbeit, Klausur (60 Minuten)
Medienformen	Beamerpräsentationen, Tafel, Unterlagen
Literatur	R. Love, Linux Kernel Development, 2nd Edition, Novell Press, 2005. Bovet, D. P. and Cesati, M., Understanding the Linux Kernel, 3rd Edition, 2005, O'Reilly. Mauerer, W., Linux Kernelarchitektur (Kernel 2.6), Nov. 2003, Hanser Fachbuchverlag. A. Rubini, Linux Device Drivers, third Edition, O'Reilly UK, 2005. M. Mark et al, Advanced Linux Programming, New Riders, 2001. J. M. Hart, Windows System Programming, 3rd Edition,

Addison Wesley, 2004.

Mark Russinovich, D. A. Solomon Windows Internals, 4th Edition, Microsoft Press, 2005.

Inhalt

Einführung

Besonderheiten der systemnahen Programmierung, Systemarchitekturen von Linux und Windows

Nebenläufige Programmierung in Linux

Prozesse und Threads in Linux, Datenstrukturen des Betriebssystemkerns und Systemaufrufe zur Erzeugung und Steuerung von Prozessen und Threads; Scheduling und Prozesszustände im Detail; Programmierung mit Mechanismen der Synchronisation und Kommunikation: Semaphore, Message Queues, Shared memory, Sockets, Domain Sockets.

Nebenläufige Programmierung in Windows

Prozesse und Threads in Windows, Datenstrukturen des Betriebssystemkerns, Win32-API: Der Begriff des „Handles“, Funktionen zur Prozesserzeugung und -steuerung; Scheduling und Prozesszustände im Detail; Programmierung mit Mechanismen der Synchronisation und Kommunikation: Mutexe, Semaphore, Critical Sections, Sockets.