
| | |
|---|---------------|
| Modulbezeichnung: Betriebssystemtechnik (BST) (Operating Systems Engineering) | 5 ECTS |
|---|---------------|

| | |
|-------------------------|------------------------------|
| Modulverantwortliche/r: | Wolfgang Schröder-Preikschat |
| Lehrende: | Wolfgang Schröder-Preikschat |

| | | |
|------------------------|-----------------------|-----------------------|
| Startsemester: SS 2021 | Dauer: 1 Semester | Turnus: jährlich (SS) |
| Präsenzzeit: 60 Std. | Eigenstudium: 90 Std. | Sprache: Deutsch |

Lehrveranstaltungen:

ACHTUNG: Die notwendigen Veranstaltungen fallen im Sommersemester 2020 aufgrund der Aussetzung aller Präsenzveranstaltungen aus. Das Modul ist somit nicht belegbar.

Betriebssystemtechnik (fällt aus) (SS 2021, Vorlesung, 2 SWS, Wolfgang Schröder-Preikschat)
 Übungen zu Betriebssystemtechnik (fällt aus) (SS 2021, Übung, 2 SWS, Bernhard Heinloth et al.)
 Rechnerübungen zu Betriebssystemtechnik (fällt aus) (SS 2021, Übung, Bernhard Heinloth et al.)

Inhalt:

Schwerpunktthema der Veranstaltung ist die Verwaltung von Prozessadressräumen. Untersucht werden Verfahren und Techniken zur Trennung logischer Adressräume, zum Adressraumgrenzen überschreitenden Zugriff und zum Schutz von Prozessen. Vorgestellt wird die Implementierung von Systemaufrufen und seiten- wie auch segmentbasierte Techniken zur Abbildung logischer/virtueller Adressräume auf reale. Vor diesem Hintergrund werden verschiedene Betriebssystemarchitekturen verglichen und gängige Adressraummodelle von Betriebssystemen erläutert. Weitere Themen bildet die Interprozesskommunikation durch Nachrichtenversenden bei getrennten Adressräumen, aber auch die Nachbildung virtuell gemeinsamen Speichers auf Basis solcher Ansätze.

Lernziele und Kompetenzen:

Studierende, die das Modul erfolgreich abgeschlossen haben:

- erläutern und implementieren Entwurfsprinzipien für Systemaufrufe und diskutieren deren spezifischen Vor-/Nachteile.
- vergleichen grundlegende BS-Architekturen (Monolith, Mikrokern, Makrokern, Exokern) anhand fundamentaler Charakteristika (Robustheit, Performanz, Portierbarkeit) und deren Einfluss auf die Implementierung von Mechanismen (Systemaufrufe, Adressraumschutz).
- unterscheiden Hierarchiekonzepte des Softwareentwurfs (Benutzthierarchie, funktionale Hierarchie) und erläutern deren Implikationen beim Betriebssystementwurf.
- klassifizieren Schutz-, Verwaltungs-, und Virtualisierungstechniken für Programmzustände (Seitennummerierung, Segmentierung, Sprachbasierung, Capabilities) und implementieren diese auf der IA-32-Architektur
- diskutieren Adressraummodelle (Mehradressraummodell, Einadressraummodell, mehrstufige und inverse Seitenabbildungen, Mitbenutzung) und deren Implementierbarkeit auf gängigen Hardwarearchitekturen.
- implementieren Mechanismen und Abstraktionen zur Interprozesskommunikation.
- erläutern das Zusammenspiel zwischen Kommunikation und Synchronisation im Bezug auf die besonderen Herausforderungen bei der Implementierung von Betriebssystemabstraktionen (lost wakeup, lost update, gepufferte/ungepufferte Zugriffe)
- diskutieren Prinzipien der Mitbenutzung von Code und Daten unter Berücksichtigung der Betriebssystem- und Adressraumarchitektur.
- erläutern die Funktionsweise eines Bindeladers und skizzieren effiziente Implementierungstechniken für positionsunabhängige Strukturen.
- erschließen sich typische Probleme (Nebenläufigkeit, Compilerverhalten, Debuggen ohne dedizierte Hilfsmittel) und Fehlerquellen bei der hardwarenahen Softwareentwicklung.
- können in Kleingruppen kooperativ arbeiten.
- können die ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten.
- reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab.
- können offen und konstruktiv mit Schwachpunkten und Irrwegen umgehen.

Literatur:

- Wolfgang Schröder-Preikschat. The Logical Design of Parallel Operating Systems. Prentice Hall.
- Brian W. Kernighan, Dennis M. Ritchie. The C Programming Language. Prentice Hall.
-
- David M. Weiss, Chi Tau Robert Lai. Software Product-Line Engineering: A Family-Based Software Development Process. Addison-Wesley.
- Krzysztof Czarnecki, Ulrich W. Eisenecker. Generative Programming. Methods, Tools and Applications. Addison-Wesley.

Studien-/Prüfungsleistungen:

Betriebssystemtechnik (Vorlesung mit Übungen) (Prüfungsnummer: 406841)

(englische Bezeichnung: Betriebssystemtechnik (Vorlesung mit Übungen))

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: SS 2021, 1. Wdh.: WS 2021/2022

1. Prüfer: Wolfgang Schröder-Preikschat
