

Modul INF-MSc-409: Betriebssystembau					
Englischer Modultitel: Operating System Construction					
Studiengänge: Masterstudiengang Informatik, Masterstudiengang Angewandte Informatik					
Turnus nach Ankündigung	Dauer 1 Semester	Studienabschnitt 2.-3. Semester	Credits 6	Aufwand 180 (90/150)	
1	Modulstruktur				
	Nr.	Element / Lehrveranstaltung	Typ	Credits	SWS
	1	Betriebssystembau	V	3	2
	2	Übungen zu Betriebssystembau	Ü	3	2
2	Lehrveranstaltungssprache: deutsch				
3	<p>Lehrinhalte</p> <p>Inhalt des Moduls ist Vermittlung grundlegender Konzepte, Methoden und Techniken, welche für den Bau eines Betriebssystems erforderlich sind. Im Rahmen der Übungen entwickeln die Studierenden in einem "bottom-up" Entwurf- und Entwicklungsprozess ihr eigenes Einkernbetriebssystem für die IA-32 Plattform, ausgehend von der "nackten Hardware" über grundlegende Ein-Ausgabemöglichkeiten, Unterbrechungsbearbeitung bis hin zu quasiparalleler Programmausführung.</p>				
4	<p>Kompetenzen</p> <p>Ziel des Moduls ist die Entwicklung eines tief gehenden Verständnisses der Vorgänge in einem Betriebssystem sowie an der Schnittstelle zwischen Systemsoftware und Rechnerhardware. Darüber hinaus wie Systemsoftware praktisch implementiert wird und wie mit schwer zu durchschauenden Problemen, die durch Nebenläufigkeit von Aktivitäten entstehen können, umgegangen werden kann.</p> <p>Studierende, die das Modul erfolgreich abgeschlossen haben:</p> <ul style="list-style-type: none"> • erläutern den Startvorgang eines Rechensystems am Beispiel eines IA32 PCs. • beschreiben die spezifischen Herausforderungen bei der Softwareentwicklung für "bare metal". • beschreiben den Ablauf einer Unterbrechungsbehandlung von der Hardware bis zur (System-)software. • skizzieren Besonderheiten und Strategien der Unterbrechungsbehandlung in Hardware für Mehrkernsystemen am Beispiel des IA32-APICs. • diskutieren die Aufgabenteilung zwischen Hardware und Systemsoftware bei der Unterbrechungsbearbeitung. • unterscheiden die verschiedenen Typen von Kontrollflüssen in einem Betriebssystem anhand des Ebenenmodells. • unterscheiden harte, mehrstufige, und weiche Verfahren zur Unterbrechungssynchronisation in Betriebssystemen und können diese implementieren. • klassifizieren konkrete Konkurrenzsituationen anhand des Ebenenmodells und leiten daraus geeignete Synchronisationsmaßnahmen ab. • schildern die IA32-Architektur und gängige PC-Technologie und deren Schnittstellen zur Systemsoftware. • erläutern grundlegende Bausteine für die Implementierung von Quasi-Parallelität (Fortsetzungen, Koroutinen, Fäden) und grenzen diese gegeneinander ab. • erläutern die Interaktionen zwischen Hardware, Übersetzer und Systemsoftware, die dabei zu beachten sind. • entwickeln den Koroutinenwechsel für einen gegebene Architektur. • erläutern die Implikationen von Quasi-Parallelität auf das Ebenenmodell und die daraus abgeleiteten Synchronisationsmaßnahmen. • beschreiben die Implementierung von (verdrängendem) Scheduling in einem Betriebssystem. • analysieren das Zusammenspiel von Scheduling und Unterbrechungssynchronisation. • nennen Kriterien und Dimensionen des Scheduling von Betriebsmitteln, insbesondere der CPU. • erläutern die konkrete Umsetzung am Beispiel der Scheduler in Linux und Windows. • unterscheiden grundlegende Möglichkeiten der Koordinierung und Synchronisation von Fäden (aktives/passives Warten, nichtverdrängbare kritische Abschnitte). • entwickeln Mechanismen für die Synchronisation auf Fadenebene. 				

	<ul style="list-style-type: none"> • erklären die dabei zu beachtenden Synchronisationsprobleme (lost update, lost wakeup) und geeignete Gegenmaßnahmen. • interpretieren die Bedeutung von Gerätetreibern in der Betriebssystempraxis. • erläutern die Anforderungen an ein Treibermodell. • vergleichen die Umsetzung von Treibermodellen in Windows und Linux. • vergleichen grundlegende BS-Architekturen (Bibliothek, Monolith, Mikrokern, Exokern, Hypervisor) anhand fundamentaler Charakteristika (Robustheit, Performanz, Portierbarkeit) und Mechanismen. • schildern die grundlegenden Paradigmen zur Interprozesskommunikation in Betriebssystemen (speicherbasiert vs. nachrichtenbasiert). • erläutern die grundlegenden Primitiven dieser Verfahren. • skizzieren, wie unter Anwendung dieser Primitiven höhere Synchronisationskonstrukte implementiert werden (Monitore, Leser-/Schreiber-Sperre). • illustrieren die Dualität der Paradigmen. • erschließen sich typische Probleme (Nebenläufigkeit, Compilerverhalten, Debuggen ohne dedizierte Hilfsmittel) und Fehlerquellen bei der hardwarenahen Softwareentwicklung. • können in Gruppen kooperativ und effektiv arbeiten. • können ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten. • reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab. • können offen und konstruktiv mit Schwachpunkten in der Konzeption wie Umsetzung umgehen. 	
5	Prüfungen <i>Modulprüfung: mündliche Prüfung (30 Minuten)</i> <small>BOSS-NR. 66391</small> <i>Studienleistung: -keine-</i>	
6	Prüfungsformen und -leistungen <input checked="" type="checkbox"/> Modulprüfung <input type="checkbox"/> Teilleistungen	
7	Teilnahmevoraussetzungen <i>Erfolgreich abgeschlossen: -keine-</i> <i>Vorausgesetzte Kenntnisse: Betriebssysteme, Rechnernetze und Verteilte Systeme</i> <i>Wünschenswerte Kenntnisse: Ein Basismodul des Forschungsbereichs Eingebettete und Verteilte Systeme</i>	
8	Modultyp und Verwendbarkeit des Moduls Vertiefungsmodul im Masterstudiengang Informatik und Masterstudiengang Angewandte Informatik Forschungsbereich Eingebettete und verteilte Systeme	
9	Modulbeauftragte/r Prof. Dr. P. Ulbrich	Zuständige Fakultät Informatik
		<small>Beschluss Fakultätsrat 13.01.2010 Änderung Fakultätsrat 22.02.2017, 22.05.2019, 18.10.2022</small>