

Modulbezeichnung	<b>Entwicklung von Betriebssystemen</b>								
Modulverantwortliche(r)	Prof. Dr. J. Peleska								
Modulart	Pflicht/Wahl <input type="checkbox"/> Wahlpflicht <input checked="" type="checkbox"/>								
Spezialisierungsbereich	Systemsoftware / Eingebettete Systeme								
Dauer des Moduls	1 Semester								
Kreditpunkte	6 CP								
Arbeitsaufwand	<table> <tr> <td>Berechnung des Workloads</td> <td></td> </tr> <tr> <td>Präsenz</td> <td>56 h</td> </tr> <tr> <td>Übungsbetrieb/Prüfungsvorbereitung</td> <td>124 h</td> </tr> <tr> <td>Summe</td> <td>180 h</td> </tr> </table>	Berechnung des Workloads		Präsenz	56 h	Übungsbetrieb/Prüfungsvorbereitung	124 h	Summe	180 h
Berechnung des Workloads									
Präsenz	56 h								
Übungsbetrieb/Prüfungsvorbereitung	124 h								
Summe	180 h								
Turnus des Moduls	i. d. R. angeboten alle 2 Semester								
Voraussetzung für die Teilnahme	Keine <input type="checkbox"/> Folgende Inhaltliche Voraussetzungen: Betriebssysteme, Programmierkenntnisse in C/C++								
Lehr- und Lernformen	Seminar <input type="checkbox"/> Vorlesung <input checked="" type="checkbox"/> Tutorium <input checked="" type="checkbox"/> Praktikum <input type="checkbox"/> Projekt <input type="checkbox"/>								
Lernziele	<p>Die Studierenden: - verfügen über vertiefte Kenntnisse über die Interna von Betriebssystemen - sind in der Lage, selbständig neue Betriebssystemdienste oder sogar ganze Systeme zu entwickeln.</p> <p>Die hier erworbenen Kenntnisse sind heute vor allem in den Anwendungsbereichen Mobilkommunikation, Mobile Computing, Wearable Computing und Steuerung sicherheits-relevanter Anwendungen wichtig: Hier werden zur Zeit viele neue Betriebssysteme oder Komponenten davon entwickelt, um den spezialisierten Anforderungen der Anwendungsdomäne Rechnung zu tragen.</p>								
Lerninhalte	<p>Als „Anschauungsobjekt“ für die Vorlesung dient das Linux-Betriebssystem, dessen frei verfügbare Quellen eine gute Grundlage bieten, um Implementierungstechniken zu verstehen und die Erweiterung des Kernels zu erlernen.</p> <ol style="list-style-type: none"> <li>1. Implementierung von Systemaufrufen unter Linux</li> <li>2. Entwicklung von Kernel-Modulen</li> <li>3. Scheduling Policies</li> <li>4. Architektur, Objekte und Methoden des virtuellen Dateisystems.</li> <li>5. Die Dateisysteme ext2 und ext3</li> <li>6. Interrupts und ihre Behandlung</li> <li>7. Entwicklung von Treibern</li> </ol> <p>Die Übungen vertiefen den Stoff anhand von Aufgaben aus den Bereichen Systemprogrammierung – Entwicklung von Systemaufrufen und Integration in den LinuxKernel – Entwicklung von Dateisystemen – Entwicklung von Linux Kernel-Modulen.</p>								
Prüfungsformen	i.d.R. Bearbeitung von Übungsaufgaben und Fachgespräch oder mündliche Prüfung								

Literatur

- Robert Love: Linux Kernel Development, Second Edition, Novell Press, Indianapolis, USA, 2005.
- U. Vahalia: Unix Internals - The New Frontiers, Prentice Hall 1996.
- Wolfgang Maurer: Linux Kernelarchitektur. Konzepte, Strukturen und Algorithmen von Kernel 2.6, Hanser (2005).
- Krzysztof R. Apt and Ernst-Rüdiger Olderog: Verification of Sequential and Concurrent Programs., Springer, 1991
- J. Corbet, A. Rubini and G. Kroah-Hartman: Linux Device Drivers., O'Reilly, 2005.
- Thomas H. Cormen, Charles E. Leiserson and Ronald L. Rivest: Introduction to Algorithms., The MIT Press, Cambridge Massachusetts, McGraw-Hill Book Company, New York, 1999