

Modulbezeichnung	<b>Technische Informatik 2</b>								
Modulverantwortliche(r)	Prof. Dr. U. Bormann								
Modulart	Pflicht/Wahl <input checked="" type="checkbox"/> Wahlpflicht <input type="checkbox"/>								
Spezialisierungsbereich									
Dauer des Moduls	1 Semester								
Kreditpunkte	8 CP								
Arbeitsaufwand	<table> <tr> <td>Berechnung des Workloads</td> <td></td> </tr> <tr> <td>Präsenz</td> <td>84 h</td> </tr> <tr> <td>Übungsbetrieb/Prüfungsvorbereitung</td> <td>156 h</td> </tr> <tr> <td>Summe</td> <td>240 h</td> </tr> </table>	Berechnung des Workloads		Präsenz	84 h	Übungsbetrieb/Prüfungsvorbereitung	156 h	Summe	240 h
Berechnung des Workloads									
Präsenz	84 h								
Übungsbetrieb/Prüfungsvorbereitung	156 h								
Summe	240 h								
Turnus des Moduls	angeboten in jedem WiSe								
Voraussetzung für die Teilnahme	Keine <input type="checkbox"/> Folgende Inhaltliche Voraussetzungen: Praktische Informatik 2, Technische Informatik 1								
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	<ul style="list-style-type: none"> <li>• In der Terminologie der Betriebssysteme und nebenläufigen Systeme kommunizieren können.</li> <li>• Abstraktionshierarchien (Speicherverwaltung, Dateisystem) in Bezug auf ihre Auswirkung auf die Systemleistung einschätzen können.</li> <li>• Lösungsvarianten für Systemsoftwarekomponenten und den Umgang mit Nebenläufigkeit bewerten können (s. unten).</li> <li>• Schutzmechanismen in Bezug auf Anwendungssicherheitsziele anwenden können.</li> <li>• Selbständiges Entwickeln von einfachen Systemkomponenten in C++ für Unix.</li> <li>• Die globalen Strategien auf einfache vorgegebene Einzelsituationen übertragen können.</li> <li>• In Gruppen Probleme analysieren, gemeinsam Lösungsstrategien entwickeln und präsentieren können.</li> </ul>								

Lerninhalte	<p>I. Grundlagen der Betriebssysteme</p> <ul style="list-style-type: none"> <li>● Betriebssysteme: Aufgaben, Rechnerbetriebsformen und Elemente von Betriebssystemen, Anmerkungen zur Geschichte und Überblick über die Entwicklung der Betriebssysteme</li> <li>● Prozessverwaltung: Einfache Prozesse, Prozesseigenschaften, Unterbrechungen, Systemaufrufe, Ausnahmen, Echtzeitbetrieb</li> <li>● Speicherverwaltung: Ein-/Auslagerungsverfahren</li> <li>● Dateisystem: Namen, Baumstruktur; Zugriffsoperationen; Abbildung auf reale Geräte; Ein/Ausgabe; Sicherheit (Schutzmechanismen, Zugriffsrechte)</li> <li>● Befehlsinterpretierer</li> </ul> <p>II. Nebenläufigkeit</p> <ul style="list-style-type: none"> <li>● Synchronisation: Semaphore, (bedingte) kritische Abschnitte, Ereignisse, Monitore, synchroner/asynchroner Nachrichtenaustausch, "Rendezvous", Kanäle, verteilte Systeme mit Prozedurfernaufrufen</li> <li>● Verklemmungen, Lebendigkeit, Fairness; Korrektheit</li> <li>● Formale Beschreibung nebenläufiger Systeme, z.B. mit Petri-Netzen (Überblick)</li> <li>● Spezielle nebenläufige Systeme: Speisende Philosophen, Erzeuger/Verbraucher, Leser/Schreiber usw.</li> <li>● Grundlagen der Rechnernetze, Client/Server-Architekturen, lokale und globale Netze (Überblick, Ethernet, IP, TCP, HTTP), Sicherheit (Grundlagen der Kryptographie)</li> </ul>
Prüfungsformen	i.d.R. Bearbeitung von Übungsaufgaben und Fachgespräch
Literatur	Andrew S. Tanenbaum: Modern Operating Systems, 4th Edition, Pearson Studium, 2016 (bzw. die deutsche Übersetzung: Moderne Betriebssysteme, 4. Auflage, Pearson Studium, 2016)