

## *Technische Informatik II (T2INF2005)*

Formale Angaben zum Modul		
Studiengang	Studienrichtung	Vertiefung
Informatik	-	-

Modulbezeichnung	Sprache	Nummer	Version	Modulverantwortlicher
Technische Informatik II	Deutsch	T2INF2005	1	Dr. -Ing. Alfred Strey

Verortung des Moduls im Studienverlauf			
Semester	Voraussetzungen für die Teilnahme	Modulart	Moduldauer
-	Technische Informatik I (T2INF1006), Programmieren (T2INF1004)	Kernmodul	2

Eingesetzte Lehr- und Prüfungsformen	
Lehrformen	Vorlesung, Übung
Lehrmethoden	Lehrvortrag, Diskussion

Prüfungsleistung	Benotung	Prüfungsumfang (in min)
Klausur	Standardnoten	150

Workload und ECTS			
Workload insgesamt (in h)	davon Präsenzzeit (in h)	davon Selbststudium (in h)	ECTS-Punkte
240,0	96,0	144,0	8

Qualifikationsziele und Kompetenzen	
<b>Sachkompetenz</b>	Die Studierenden lernen den Aufbau und die Arbeitsweise von digitalen Rechenanlagen kennen. In einem Übungsteil wird ihnen die systemnahe Programmierung anhand eines Beispielprozessors vermittelt. Abgerundet wird dieses hardwarznahe Wissen durch die Unit "Betriebssysteme", welche die Arbeitsweise von Rechenanlagen aus Sicht der Software beleuchtet. Die Studierenden sind somit in der Lage das Zusammenwirken von Hard- und Software in einem Rechner zu verstehen.
<b>Selbstkompetenz</b>	Die Studierenden sind in der Lage die Leistungsfähigkeit von Rechenanlagen zu beurteilen und selbst systemnahe Programme zu schreiben. Die rasche Weiterentwicklung auf diesem Sektor mitzuverfolgen und zu verstehen welche Vor- bzw. Nachteile die Markteinführung einer neuen IT-Technologie hat, ist ihnen jederzeit möglich. Auch sind sie in der Lage zu verstehen wie die neue Technologie arbeitet bzw. sie können sich das dazu notwendige neue Wissen jederzeit selbst erarbeiten.
<b>Sozial-ethische Kompetenz</b>	-

## Lerneinheiten und Inhalte

Lehr- und Lerneinheiten	Präsenz	Selbststudium
<b>Rechnerarchitekturen 1</b>	<b>36,0</b>	<b>54,0</b>
<ul style="list-style-type: none"> <li>- Einführung</li> <li>- Historie (mechanisch, analog, digital)</li> <li>- Architektur nach von Neumann</li> <li>- Systemkomponenten im Überblick</li> <li>- Grobstruktur der Prozessorinterne</li> <li>- Rechenwerk</li> <li>- Addition: Halbaddierer, Volladdierer, Wortaddierer, Bedeutung des Carrybits, Ripple + Look-ahead Carry</li> <li>- Subtraktion: Transformation aus Addition, Bedeutung des Carrybits</li> <li>- Multiplikation: Parallel- und Seriell-Multiplizierer</li> <li>- Division: Konzept</li> <li>- Arithmetische-logische Einheit (ALU)</li> <li>- ALU mit Rechenregister und Ergebnisflags (CCR, Statusbits)</li> <li>- Steuerwerk</li> <li>- Aufbau und Komponenten</li> <li>- Befehlsdekodierung und Mikroprogrammierung</li> <li>- Struktur von Prozessorbefehlssätzen</li> <li>- Klassifizierung und Anwendung Prozessorregistern (Daten, Adressen und Status)</li> <li>- Businterface</li> <li>- Buskomponenten: Daten-, Adress- und Steuerleitungen</li> <li>- Buszyklen: Lese- und Schreib-Zugriff, Handshaking (insbesondere Waitstates)</li> <li>- Busarbitrierung</li> <li>- Busmultiplexing</li> <li>- Fundamentalarchitekturen</li> <li>- Konzept Systemaufbau und Komponenten: CPU, Hauptspeicher, I/O: Diskussion Anbindung externer Geräte (Grafik, Tastatur, Festplatten, DVD, ...)- Halbleiterspeicher</li> <li>- Wahlfreie Speicher: Aufbau, Funktion, Adressdekodierung, interne Matrixorganisation</li> <li>- RAM: statisch, dynamisch, aktuelle Entwicklungen</li> <li>- ROM: Maske, Fuse, EPROM, EEPROM, FEPR0M, aktuelle Entwicklungen</li> <li>- Systemaufbau</li> <li>- Aufteilung des Adressierungsraumes</li> <li>- Entwerfen von Speicherschemata und der zugehörigen Adress-Dekodierlogik</li> <li>- Vitale System-Komponenten: Stromversorgung, Rücksetzlogik, Systemtakt, Chipsatz</li> <li>- Schaltkreise: Interrupt- und DMA-Controller, Zeitgeber- und Uhrenbausteine</li> <li>- Schnittstellen: Parallel und seriell, Standards (RS232, USB, ...)</li> <li>- Performancekonzepte</li> </ul>		
<b>Betriebssysteme</b>	<b>36,0</b>	<b>54,0</b>
<ul style="list-style-type: none"> <li>- Einführung</li> <li>- Historischer Überblick</li> <li>- Betriebssystemkonzepte</li> <li>- Prozesse und Threads</li> <li>- Einführung in das Konzept der Prozesse</li> <li>- Prozesskommunikation</li> <li>- Übungen zur Prozesskommunikation: Klassische Probleme</li> <li>- Scheduling von Prozessen</li> <li>- Threads Speicherverwaltung</li> <li>- Einfache Speicherverwaltung ohne Swapping und Paging</li> <li>- Swapping</li> <li>- Virtueller Speicher</li> <li>- Segmentierter Speicher Dateien und Dateisysteme - Dateien</li> <li>- Verzeichnisse</li> <li>- Implementierung von Dateisystemen</li> <li>- Sicherheit von Dateisystemen</li> <li>- Schutzmechanismen</li> <li>- Neue Entwicklungen: Log-basierte Dateisysteme Ein- und Ausgabe</li> <li>- Grundlegende Eigenschaften der I/O-Hardware</li> <li>- Festplatten</li> <li>- Terminals</li> <li>- Die I/O-Software Anwendung der Prinzipien auf reale Betriebssysteme: UNIX und Windows Windows NT, 2000, XP, Windows7,</li> </ul>		
<b>Systemnahe Programmierung 1</b>	<b>24,0</b>	<b>36,0</b>
<ul style="list-style-type: none"> <li>- Befehlssatz und Maschinenprogrammierung</li> <li>- Programmiermodell: Befehlssatz und Adressierungsarten</li> <li>- Umsetzung von Kontrollstrukturen, Auswertung von Ergebnisflags</li> <li>- Unterprogrammstruktur mit Hilfe des Hardwarestapels: Mechanismen, Aufruf</li> <li>- Konventionen</li> <li>- Konzept und Umsetzung (HW- und SW-Interrupts): Diskussion von HW- und SW-Mechanismen und Automatismen, IR-Vektortabelle; Spezialfall: Bootvorgang</li> <li>- Diskussion User- und Supervisor-Modus von Prozessoren</li> <li>- Praktische Übungen</li> <li>- Einführung eines Beispielprozessors</li> <li>- Aufbau des Übungsrechners</li> <li>- Einarbeitung und Softwareentwicklungs- und Testumgebung für den Übungsrechner</li> <li>- Selbständige Entwicklung von Maschinenprogrammen mit steigendem Schwierigkeits- und Strukturierungsgrad</li> </ul>		

## Besonderheiten und Voraussetzungen

### Besonderheiten

-

### Voraussetzungen

-

## Literatur

- Elektronik 4: Digitaltechnik, K. Beuth, Vogel Fachbuch, 2006
- Elektronik 5: Mikroprozessortechnik, H. Müller / L. Walz, Vogel Fachbuch, 2005
- Computerarchitektur, A. S. Tanenbaum, Person Studium, 2005
- Rechneraufbau und Rechnerstrukturen, W. Oberschelp / G. Vossen, Oldenbourg Wissenschaftsverlag, 2006
- Mikroprozessortechnik und Rechnerstrukturen, T. Flik, Springer, 2004
- Technische Informatik 2, W. Schiffmann / R. Schmitz, Springer, 2005
- Tanenbaum A.S.: Moderne Betriebssysteme, Pearson Studium, 2001
- Tanenbaum A.S.: Verteilte Betriebssysteme, Prentice Hall, München, London, New York, 1995
- Tanenbaum A.S., van Steen Marten: Verteilte Systeme – Grundlagen und Paradigmen - , Pearson Studium, 2002
- Mullender S.: Distributed Systems, Addison-Wesley, New York, 1993
- Quinn B., Shute, D.: Windows Sockets Network Programming, Addison-Wesley, Reading, 1995
- Gray J.S.: Interprocess Communications in UNIX, Prentice Hall PTR, NJ, 1998
- Siegert H.-J., Baumgarten U.: Betriebssysteme, Oldenbourg MünchenWien, 1998
- Templeman Julian, Olsen Andy: Visual C++ Schritt für Schritt, Microsoft Press, 2001
- Petzold Charles: Windows Programmierung mit C#, Microsoft Press, 2002
-