

# Lehrveranstaltung

## BVS1 - Betriebssysteme und Verteilte Systeme 1

---

Version: 3 | Letzte Änderung: 01.04.2022 09:35 | Entwurf: 0 | Status: vom verantwortlichen Dozent freigegeben

### ^ Allgemeine Informationen

Langname	Betriebssysteme und Verteilte Systeme 1
Anerkennende LModule	<a href="#">BVS1_BaET</a> , <a href="#">BVS1_BaTIN</a>
Verantwortlich	Prof. Dr. Cartsten Vogt Professor Fakultät IME
Niveau	Bachelor
Semester im Jahr	Wintersemester
Dauer	Semester
Stunden im Selbststudium	60
ECTS	5
Dozenten	Prof. Dr. Cartsten Vogt Professor Fakultät IME
Voraussetzungen	prozedurale Programmierung Architektur von Digitalrechnern (Grundkenntnisse) Internetprotokolle (Grundkenntnisse)
Unterrichtssprache	deutsch
separate Abschlussprüfung	Ja

## Abschlussprüfung

### Details

Die Studierenden sollen in einer schriftlichen Klausur folgende Kompetenzen nachweisen: 1.) Sicherer Umgang mit grundlegenden Begrifflichkeiten, Konzepten und Techniken, 2.) Anwendung programmiersprachlicher und abstrakterer Konstrukte zur Lösung von Anwendungsproblemen bei der nebenläufigen und verteilten Programmierung, 3.) Prüfung von Lösungsvorschlägen auf Korrektheit. Typische Aufgabenformen zu 1.) sind Multiple-Choice-Fragen, Lückentexte, Bewertung von Aussagen hinsichtlich ihrer Korrektheit, zu 2.) Lösung kleinerer umgangssprachlich formulierter Probleme durch Programmstücke oder in abstrakterer Form und zu 3.) das Finden von Fehlern in Aussagen und vorgegebenen Programmstücken.

## Mindeststandard

Mindestens 50% der möglichen Gesamtpunktzahl.

## Prüfungstyp

Die Studierenden sollen in einer schriftlichen Klausur folgende Kompetenzen nachweisen: 1.) Sicherer Umgang mit grundlegenden Begrifflichkeiten, Konzepten und Techniken, 2.) Anwendung programmiersprachlicher und abstrakter Konstrukte zur Lösung von Anwendungsproblemen bei der nebenläufigen und verteilten Programmierung, 3.) Prüfung von Lösungsvorschlägen auf Korrektheit. Typische Aufgabenformen zu 1.) sind Multiple-Choice-Fragen, Lückentexte, Bewertung von Aussagen hinsichtlich ihrer Korrektheit, zu 2.) Lösung kleinerer umgangssprachlich formulierter Probleme durch Programmstücke oder in abstrakterer Form und zu 3.) das Finden von Fehlern in Aussagen und vorgegebenen Programmstücken.

## ^ Vorlesung / Übungen

## Lernziele

---

### Kenntnisse

Grundlagen von Betriebssystemen und Verteilten Systemen  
Einordnung und Aufgaben eines Betriebssystems im Rechner-System  
zu verwaltende Betriebsmittel  
Nebenläufigkeit in Hard- und Software  
Komponenten und Eigenschaften Verteilter Systeme  
Software-Strukturen  
Betriebssystemkern  
Hierarchische Strukturen  
Virtuelle Maschinen  
Client-Server-Systeme  
Peer-to-Peer-Systeme

---

Das UNIX/Linux-Betriebssystem  
Geschichte und Standards  
Schalenstruktur  
Kern mit Programmierschnittstelle  
Shell mit Benutzerschnittstelle  
wichtige Benutzerkommandos  
Aufbau des Dateisystems  
Programmierung in C

---

Nebenläufigkeit  
Prozesse und Threads  
grundlegende Eigenschaften  
Prozesse in UNIX  
Threads in Java  
Synchronisation  
grundlegende Synchronisationsbedingungen  
wechselseitiger Ausschluss  
Reihenfolge  
Mechanismen zur Durchsetzung  
Interruptsperrung  
Spinlocks

Signale  
Semaphore  
Monitore  
Deadlocks

---

Kommunikation  
Grundbegriffe  
speicher- vs. nachrichtenbasierte Kommunikation  
Mailboxen und Ports  
synchrone vs. asynchrone Kommunikation  
lokale Kommunikation  
Shared Memory  
Message Queues  
Pipes  
Kommunikation in verteilten Systemen  
Protokolle  
Sockets

---

## Fertigkeiten

Umgang mit den Schnittstellen eines Betriebssystems  
zeichenorientierte Benutzerschnittstelle (Konsole)  
Programmierschnittstelle

---

Steuerung nebenläufiger Aktivitäten in einem Betriebssystem  
von der Benutzerschnittstelle aus  
durch Funktionen der Programmierschnittstelle

---

Synchronisation nebenläufiger Ausführungen durch Synchronisationsmechanismen

---

Nutzung verschiedener Kommunikationsmechanismen  
lokale Mechanismen  
Mechanismen in Rechnernetzen

## Aufwand Präsenzlehre

Typ	Präsenzzeit (h/Wo.)
Vorlesung	2
Übungen (ganzer Kurs)	1
Übungen (geteilter Kurs)	1
Tutorium (freiwillig)	0

## Separate Prüfung

## ^ Praktikum

### Lernziele

---

#### Kenntnisse

Befehle der zeichenorientierten UNIX/Linux-Benutzerschnittstelle  
Nutzung durch Eingabe über die Tastatur  
Nutzung durch Einbettung in Shell Scripts  
insbesondere zur Steuerung nebenläufiger Prozesse

---

C-Funktionen der UNIX/Linux-Programmierschnittstelle  
zum Zugriff auf Dateien und Geräte  
zur Erzeugung und elementaren Steuerung von Prozessen  
zur Synchronisation von Prozessen  
zur Kommunikation von Prozessen (lokal und im Netz) - je nach verfügbarer Zeit

---

#### Fertigkeiten

Anwendung der unter "Kenntnisse (fachliche Inhalte)" genannten Aspekte auf praxisbezogene Szenarien durch selbstständige Arbeit in kleinem Team.

### Aufwand Präsenzlehre

Typ	Präsenzzeit (h/Wo.)
Praktikum	1
Tutorium (freiwillig)	0

### Separate Prüfung

#### Prüfungstyp

praxisnahes Szenario bearbeiten (z.B. im Praktikum)

#### Details

Die Studierenden schließen sich zu Kleingruppen zusammen. Jede Kleingruppe absolviert mehrere "Praktikumsrunden" mit zugewiesenen Laborterminen. In jeder Runde werden Programmieraufgaben gelöst.

Zur Vorbereitung eines Labortermins muss ein "Vorbereitungsblatt" praktisch gelöst werden. Die dabei erworbenen Kenntnisse werden zu Beginn des Termins geprüft (kurzer schriftlicher Eingangstest, persönliches Gespräch mit dem Betreuer). Wird diese Prüfung nicht bestanden, so muss ein Folgetermin wahrgenommen werden; im Wiederholungsfall führt dies zum Nichtbestehen des Praktikums. Im Erfolgsfall wird ein "Laborarbeitsblatt" mit weiteren Aufgaben unter Aufsicht (und ggf. mit Hilfestellung) bearbeitet.

## **Mindeststandard**

Erfolgreiche Teilnahme an allen Laborterminen, d.h. insbesondere selbstständige (ggf. mit Hilfestellung) Lösung der Programmieraufgaben.