

<b>03-IMVP-VPP</b>	<b>Verteilte und Parallele Programmierung mit Virtuellen Maschinen</b>
	<i>Distributed and Parallel Programming with Virtual Machine</i>

**Lehrform** (*teaching format*) / **SWS** (*hours per week*): 4K

**Kreditpunkte** (*credit points*): 6

**Turnus** (*frequency*): i.d.R. jedes SoSe

**Inhaltliche Voraussetzungen** (*content-related prior knowledge/skills*): KEINE

**Sprache** (*language*): Deutsch

**Lehrende** (*teaching staff*): PD Dr. Stefan Bosse

<b>Studiengang</b> ( <i>degree program</i> )	<b>Module</b>	<b>Semester</b>
Informatik (Master)	IMVP, IMVP-AI, IMVP-SQ	ab 1.Sem.
Systems Engineering I/II (Master)	M07-VT-AuR, M07-VT-ESS	ab 1./2.Sem.
Produktionstechnik	(auf Anfrage)	ab 1.Sem.
Informatik (Bachelor)	(nur <i>Freie Wahl</i> )	

### **Lernergebnisse:**

- Verständnis der Grundprinzipien und Architekturen verteilter (VS) und paralleler Systeme (PS) und Fähigkeit zum Transfer auf technische Systeme, vor allem mit Fokus auf VMs
- Verständnis von Fehlverhalten von PS/VS und Anwendung von geeigneten Programmiermethoden und Fehlerbehebung zur Vermeidung von Fehlern
- Verständnis und Fähigkeit der Programmierung mit Synchronisation und Kommunikation in VS und PS
- Verständnis der Probleme und dem Betrieb von parallelen Systemen im Vergleich zu sequenziellen Systemen (Effizienz, Blockierung, Skalierung, Ressourcenbedarf)
- Praktische Kenntnisse der Programmierung von PS und VS anhand von Programmierübungen mit Lua und der PLVM; Verwendung digitaler WEB Notebooks und Workbooks für die Übungen

### **Learning Outcome:**

- Understanding of the basic principles and architectures of distributed (VS) and parallel systems (PS) and ability to transfer to technical systems, especially with a focus on VMs
- Understanding PS/VS misbehavior and applying appropriate programming methods and troubleshooting to avoid errors
- Understanding and ability of programming with synchronization and communication in VS and PS
- Understanding the problems and operation of parallel systems compared to sequential systems (efficiency, blocking, scaling, resource requirements)
- Practical knowledge of programming PS and VS based on programming exercises with Lua and the PLVM; use of digital WEB notebooks and workbooks for the exercises

### **Inhalte:**

- Einführung von Multiprozessmodellen mit Prozessalgebra und Prozessflussdiagrammen basierend auf Hoare's CSP Modell (Kommunizierende Sequenzielle Prozesse)
- Erweiterung vom CSP Modell um Konkurrenz und geteilte Ressourcen (CCSP); Konfliktlösung; Analyse mit

## Petri Netzen und Zustandsraumdiagrammen

- Einführung von parallelen und verteilten Programmiersprachen (OCCAM, Funktionale Programmierung, Multi-threading, MPI, Lua) und Verknüpfung mit Proezssalgebra und dem CCSP Modell
- Einführung in Virtuelle Maschinen und Speichermanagement, Möglichkeiten der Parallelisierung von sequenziellen Maschinen
- Virtualisierung und parellele/veretilde Virtuelle Maschinen (PVM)
- Zelluläre Automaten als paralleles und verteiltes Rechner- und Softwaremodell
- Synchronisation und Interprozesskommunikation; Protokolle, Fehler wie Blockierung; Eigenschaften von PS/VS
- Synchronisationsobjekte und deren Anwendung in der Programmierung {Mutex, Semaphore, Monitor, Event, Queue, Barriere, Channel} und deren Implementierung (VM)
- Rechnerarchitekturen für parallele und verteilte Systeme (MultiCore CPU, GPU, Cluster, FPGA)
- Möglichkeiten der Parallelisierung und Verteilung gängiger Algorithmen und Programme auf PVMs
- Formale Grundlagen von parallelen Systemen; Metriken; Effizienz; Skalierung
- Verhandlungs- und Wahlalgorithmen und Gruppenkommunikation

## Contents:

- Introduction of multi-process models with process algebra and process flow diagrams based on Hoare's CSP model (Communicating Sequential Processes)
- Extension of the CSP model by competition and shared resources (CCSP); conflict resolution; analysis with Petri networks and state space diagrams
- Introduction of parallel and distributed programming languages (OCCAM, Functional programming, Multi-threading, MPI, Lua) and linking with Proezssalgebra and the CCSP model
- Introduction to virtual machines and memory management, possibilities of parallelization of sequential machines
- Virtualization and parellele/veretilde Virtual Machines (PVM)
- Cellular automata as parallel and distributed computer and software model
- Synchronization and interprocess communication; protocols, errors such as blocking; features of PS / VS
- Synchronization objects and their application in programming {Mutex, Semaphore, Monitor, Event, Queue, Barrier, Channel} and their implementation (VM)
- Computer architectures for parallel and distributed systems (MultiCore CPU, GPU, Cluster, FPGA)
- Possibilities of parallelization and distribution of common algorithms and programs on PVMs
- Formal foundations of parallel systems; metrics; efficiency; scaling
- Negotiation and election algorithms and group communication

---

**Hinweise (remarks):** In der Tabelle sind nur die primären/spezifischsten Module aufgelistet, denen diese Veranstaltung zugeordnet ist.