h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Modulbeschreibung

Advanced Systems Programming

Module numbers:	30.2650 [PVL 30.2651; Module 30.26500]
Language:	english
Study programme:	Bachelor KMI 2024/2021 - Wahlpflichtkatalog I Bachelor 2021 - Wahlpflichtkatalog I Bachelor dual KITS 2021 - Wahlpflichtkatalog I Bachelor dual KoSI 2021 - Wahlpflichtkatalog I Bachelor 2014 - Katalog I: Anwendungs- und systemorientierte Module Bachelor KMI 2014 - Katalog I: Anwendungs- und systemorientierte Module Bachelor dual KoSI 2014 - Katalog I: Anwendungs- und systemorientierte Module
Type of course:	V+P = Lecture+Practical
Weekly hours:	2+2
Credit Points:	5
Exam:	written exam
PVL (e.g. Practical):	graded (benotet [Erfolgreiche Teilnahme am Praktikum: Die Prüfungsvorleistung ist erbracht worden, wenn die benoteten Praktikumsabgaben - Übungsaufgaben und ein Projekt in Kleingruppen - mit Note 4.0 oder besser bestanden wurden])
PVL percentage:	50%
Required knowledge:	• Experience with modern C++ development (C++17) • Experience writing native software under Linux
Learning objectives:	The students are able to understand, design and implement hardware-efficient systems software. Students will learn the fundamentals of a modern systems programming language (Rust) and how it compares to the widely used systems programming language C++. Students will understand how to balance performance, safety and maintainability while writing systems software. By focusing on two different systems programming languages, good programming skills and a deep understanding of common systems programming concepts are encouraged.
Content:	• What is Systems Programming and how does it compare to Application Programming? • Zero-overhead abstractions in C++ and Rust and how they help to write fast, readable and maintainable code • The fundamentals of memory management and memory safety • Error handling concepts in C++ and Rust for writing robust systems software • System level I/O and Network Programming • Fearless concurrency • Profiling and tracing: How to measure, evaluate and tweak performance • Tools for developing, debugging and maintaining systems software Students will gain extensive hands-on experience in systems programming by analyzing open-source code and developing their own systems in the lab.
Literature:	• Computer systems: a programmer's perspective. Vol.3 - Bryant, R. E., David Richard, O. H. • The Rust Programming Language - Steve Klabnik, Carol Nichols • A Tour of C++ (2nd Edition) - Bjarne Stroustrup
Responsibility:	Lars-Olof Burchard