Technology
Arts Sciences
TH Köln

Course Manual GSP

Fundamentals in System Programming

Version: 2 | Last Change: 16.09.2019 10:26 | Draft: 0 | Status: vom verantwortlichen Dozent freigegeben

− General information

Long name	Fundamentals in System Programming
Approving CModule	GSP_BaTIN
Responsible	Prof. Dr. Lothar Thieling Professor Fakultät IME
Valid from	summer semester 2021
Level	Bachelor
Semester in the year	summer semester
Duration	Semester
Hours in self-study	60
ECTS	5
Professors	Prof. Dr. Lothar Thieling Professor Fakultät IME
Requirements	basic skills in procedural programming structure and mode of operation of a simple computer basics in digital systems finite state machines and state transition diagrams
Language	German
Separate final exam	Yes

Literature

Märtin: Rechnerarchitektur, Fachbuchverlag Leipzig (Carl Hanser)

Oberschelp/Vossen: Rechneraufbau und Rechnerstrukturen, Oldenbourg Verlag

Vogt, C: C für Java-Programmierer

Tanenbaum, Goodman: Computerarchitektur, Pearson Studium (Prentice Hall)

Final exam

Details	The students should demonstrate the following competencies in a written exam: 1.) Safe handling of basic concepts and mechanisms. 2.) Programming under C. 3.) Development of simple hardware drivers. 4.) Development of problem solutions using a microcontroller and real-time operating system.
Minimum standard	At least 50% of the total number of points

Exam Type	EN Klausur

− Lecture / Exercises

Learning goals

Goal type	Description
Knowledge	basics of C-programming constants, variable,s data types expressions, statements, control structures preprocessor expressions pointers and pointer arithmetic array, structures funtions standard libraries multi-file programs with access to libraries software development tools compiler linker debugger simulator
Knowledge	hardware-related I/O programming in C functioning of digital ports access to digital ports memory-mapped I/O isolated I/O access to I/O ports using pointers access to I/O ports using driver libraries implementation of driver libraries in C bit-wise I/O and analysis of data using C
Knowledge	programming applications for measurement and control in C implementation of finite state machines in C (mealy and moore) optimization of cyclic queries on I/O-data
Knowledge	real-time operating system requirements and comparison to "normal" operating systems cooperative and preemtive multitasking priority and states of a task mutex, semaphores event-driven multitasking intertask communication via queues deadlocks and race conditions

Special requirements

none

Accompanying material	lecture foils (electronic), set of exercise (electronic), tool chaine (compile, link, debug, simulate), set of example-codes, self-study tutorials for the tool chain
Separate exam	No

Knowledge	I/O interfaces of a computer system and its use by means of C digital ports timer/counter (pulse width modulation, cyclic interrupt generation) analog-to-digital converter serial port using the I/O interfaces from C
Knowledge	interrupts interrupt sources and types (external, internal, hardware, software) interrupt management interupt vector table interrupt service routine time sequence of the interrupt handling mechanisms for handling concurrent interrupts prioritization interruption problem specific use of these mechanisms use of interrupt driven I/O interfaces in C
Knowledge	C runtime system subroutine call in assembler stack and assembler instructions for stack manipulation program state backup and recovery using stack passing parameters to C-function using stack managing local variables using stack dynamic behavior of the stack interpretation of the stack contents using a debugger
Skills	explain the operation of a mikrocontroller-system (hardware and real-time operating system)
Skills	interpretation of detailed technical specifications of the I/O interfaces, so that meaningful configurations can be created
Skills	implementation of C driver libraries for various I/O interfaces including the interrupt support
Skills	specifying system behavior (derived from text documents)
Skills	development of problem solutions for measurement and control, which can be realized in C

Skills	describe and analyze the C runtime system

Expenditure classroom teaching

Type	Attendance (h/Wk.)
Lecture	2
Exercises (whole course)	1
Exercises (shared course)	1
Tutorial (voluntary)	0

– Practical training

Learning goals

Goal type	Description
Skills	refer to "Vorlesung/Übung->Lernziele->Fertigkeiten"
Skills	targeted use of the software development environment
Skills	manage complex tasks as a small team
Skills	development of problem solutions for measurement and control, which can be realized in C using mikrocontroller and real-time operating systems

Expenditure classroom teaching

Type	Attendance (h/Wk.)
Practical training	1
Tutorial (voluntary)	0

Special requirements

none

Accompanying material	problem and task description (electronic), tool chaine (compile, link, debug, simulate), set of example-codes, self-study tutorials for the tool chain
Separate exam	No