COURSE DESCRIPTION CARD - SYLLABUS

Course name
Operating Systems and Applications for Embedded Systems

Course

Field of study
Computing
Area of study (specialization)
Edge computing
Level of study
Second-cycle studies
Form of study
full-time

Year/Semester
1/2
Profile of study
general academic
Course offered in
Polish
Requirements
elective

Number of hours

Lecture
15
Laboratory classes
15
Other (e.g. online)
Tutorials
Projects/seminars
15

Number of credit points

4

Lecturers

Responsible for the course/lecturer:
dr inż. Mariusz Naumowicz
email: mariusz.naumowicz@put.poznan.pl
tel. +48 61 665-2364
Faculty of Computing and Telecommunications
Piotrowo 3, 60-965 Poznań

Responsible for the course/lecturer:

Prerequisites

The student starting the course should have basic knowledge of operating systems and electronics. They should also understand the need to expand their competences and be ready to cooperate as part of the team.

Course objective

-To provide students with knowledge related to modern embedded systems and operating systems dedicated to these systems.

- Familiarizing students with modern methods of designing, testing and prototyping embedded systems.

- Developing students' skills in solving complex design problems in the field of embedded systems and operating systems.

- Developing teamwork skills in students.

Course-related learning outcomes

Knowledge

1. Has advanced and detailed knowledge of the processes occurring in the life cycle of IT systems, especially the hardware layer of the systems – [K2st_W5]

2. Knows advanced methods, techniques and tools used in solving complex engineering tasks and conducting research in a selected area of computer science - [K2st_W6]

Skills

1. Can, when formulating and solving engineering tasks, integrate knowledge from various areas of computer science (and, if necessary, also knowledge from other scientific disciplines) and apply a system approach, also taking into account non-technical aspects - [K2st_U5]

2. Can correctly use the selected method of estimating the labor consumption of software development - [K2st_U7]

3. Can make a critical analysis of the existing technical solutions and propose their improvements (improvements) - [K2st_U8]

4. Can evaluate the usefulness of methods and tools for solving an engineering task consisting in building or evaluating an information system or its components, including the limitations of these methods and tools; - [K2st_U9]

Social competences

1. understands that in computer science knowledge and skills very quickly become obsolete - [K2st_K1]

2. Understands the importance of using the latest knowledge in the field of computer science in solving research and practical problems - [K2st_K2]

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Formative assessment:

a) in the field of lectures: on the basis of answers to questions about the material discussed in previous lectures,

b) in the field of laboratories: on the basis of the assessment of the current progress in the implementation of tasks,

Summative assessment:
a) in the field of lectures, verification of the assumed learning outcomes is carried out by an pass (an electronic test on the Moodle platform);

b) in the field of laboratories, verification of the assumed learning outcomes is carried out by means of a design test and an assessment of the tasks performed during each laboratory meeting;

Getting extra points for activity during classes, especially for:
- discussion of additional aspects of the issue,

- the effectiveness of applying the acquired knowledge while solving a given problem,

- the ability to cooperate as part of a team practically carrying out a detailed task in the laboratory.

Programme content

The lecture program includes the following topics:
Building the system kernel in embedded systems. Resource and process management system. Threads and processes: management, synchronization, communication. Multithreading. Interrupt handling. Hardware-dependent operating systems. Building operating systems from source. Limiting and extending the functionality of operating systems. Loading the operating system into the device: firmware, bootloader, BIOS, UEFI. Working with GIT repositories. Drivers, programming of I / O devices. Software development for embedded systems, cross compilation.

Laboratory classes are conducted in the form of 2-hour lab exercises, preceded by a 2-hour instructional session at the beginning of the semester. Exercises are carried out by 2-person teams.
The program of laboratory classes includes the following topics:
Preparing and configuration of programming software using Eclipse software and GCC compilation. Building and configuring bootloaders for dedicated devices. Builds and runs the Linux kernel for dedicated devices. Projects related to the project in the Buildroot and Yocto project. Creating and managing projects in GIT and Redmine. Programming of peripheral devices in dedicated companies.

Teaching methods

1. Lecture with multimedia presentation (diagrams, formulas, definitions, etc.) supplemented by the content of the board.
2. Laboratory exercises: multimedia presentation, presentation illustrated with examples given on the board and performance of tasks given by the teacher - practical exercises.

3. Projects: case studies, problem-solving consultation, project presentation.

Bibliography

Basic

1. Andrew S. Tanenbaum, Herbert Bos, Systemy operacyjne. Wydanie IV. Helion, 2015. ISBN: 9788328314221.
2. Daniel P. Bovet, Marco Cesati, Understanding the Linux Kernel. 3rd Edition, Helion, 2005. ISBN: 9780596554910.

Additional

1. Alex Gonzalez, Embedded Linux Projects Using Yocto Project Cookbook, Packt Publishing, 2015. ISBN: 1784395188.

Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4,0
Classes requiring direct contact with the teacher	45	2,0
Student's own work (literature studies, preparation for laboratory classes, preparation for tests, technical reports preparation) [1]	55	2,0

1 delete or add other activities as appropriate