COURSE DESCRIPTION CARD - SYLLABUS

Course name
Operating Systems and Applications for Embedded Systems [S2Inf1-PB>SOPSW]

Course

Field of study
Computing

Year/Semester
1/2

Area of study (specialization)
Edge Computing

Profile of study
general academic

Level of study
second-cycle

Course offered in
Polish

Form of study
full-time

Requirements
compulsory

Number of hours

Lecture
15

Laboratory classes
15

Other
0

Tutorials
0

Projects/seminars
15

Number of credit points

4,00

Coordinators

dr inż. Mariusz Naumowicz
mariusz.naumowicz@put.poznan.pl

Lecturers

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, verification of the assumed learning outcomes is carried out by presenting a selected issue during the lecture,
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 program covers issues related to the construction of operating systems for embedded systems and writing dedicated applications for these systems.

Course topics

The lectures cover the following topics:
- construction of 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 sources,
- limiting and extending the functionality of operating systems,
- loading the operating system to the device: firmware, bootloader, BIOS, UEFI,
- working with GIT repositories,
- drivers, programming I/O devices,
- software development for embedded systems, cross compilation.

Laboratory and design classes are conducted in the form of 2-hour meetings held in the laboratory, preceded by an instructional session at the beginning of the semester. Exercises and projects are carried

out by two-person teams of students.
The laboratory program covers the following topics:
- preparation and configuration of the Linux programming environment for building operating systems,
- building and configuring boot programs for dedicated devices,
- compiling and running the Linux kernel for dedicated devices,
- preparing your own file system using Busybox and integration with the Linux kernel,
- preparing your own operating system based on Buildroot,
- preparing your own operating system based on Yocto.

The topics of the projects are related to the content of the laboratories and concern the creation of your own operating systems and dedicated applications supporting peripheral devices.

Some of the above-mentioned program content is carried out as part of the student's own work.

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	4,00
Classes requiring direct contact with the teacher	45	2,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	55	2,00