COURSE DESCRIPTION CARD - SYLLABUS

Course name
Emulation techniques [S2Inf1-PB>TEMU]

Course

Field of study
Computing

Year/Semester
2/3

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 (e.g. online)
0

Tutorials
0

Projects/seminars
0

Number of credit points

1,00

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

- Provide students with knowledge related to modern techniques of emulation of embedded systems. - Familiarizing students with modern methods of designing, testing and prototyping embedded systems with the use of emulators. - Developing students" skills in solving complex design problems in the field of building and testing embedded systems. - Developing teamwork skills in students.

Course-related learning outcomes

Knowledge:
1. has ordered and theoretically founded general knowledge related to key issues in the field of computer science - [k2st_w2]
2. has advanced detailed knowledge of selected issues in the field of computer science - [k2st_w3]
3. has knowledge of development trends and the most important new achievements of computer

science and other selected related scientific disciplines - [k2st_w4]
4. has advanced and detailed knowledge of the life cycle of hardware or software information systems - [k2st_w5]

Skills:
1. can obtain information from literature, databases and other sources (in polish and english), integrate them, interpret and critically evaluate them, draw conclusions and formulate and exhaustively justify opinions - [k2st_u1]
2. can plan and carry out experiments, including measurements and computer simulations, interpret the obtained results and draw conclusions as well as formulate and verify hypotheses related to complex engineering problems and simple research problems - [k2st_u3]
3. can use analytical, simulation and experimental methods to formulate and solve engineering tasks and simple research problems - [k2st_u4]
4. 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]
5. can assess the usefulness and the possibility of using new achievements (methods and tools) and new it products - [k2st_u6]

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 a test (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 content presented in the course concerns broadly understood software and hardware emulation of embedded systems.

Course topics

The lecture program covers the following topics:
- modern emulation tools and their capabilities,
- use of emulators in industry,
- emulator optimization techniques,
- testing of embedded systems using an emulator,
- adapting emulators to your needs,
- hardware emulation.

Laboratory 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. The exercises are carried out by two-person teams of students.

The laboratory program covers the following topics:
- preparing the programming environment necessary to run selected emulators, building your own embedded system based on existing structures in the emulator and preparing the operating system for the emulated embedded system,
- system analysis using emulator commands and environment automation,
- designing devices using an emulator,
- testing your own solutions in the emulator,
- programming emulated devices,
- automatic system testing.

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.

Bibliography

Basic
1. John L. Hennessy, John L. Hennessy, Computer Architecture: A Quantitative Approach, 4th Edition. Elsevier, 2007. ISBN: 0123704901.
2. Eldad Eilam, Reversing: Secrets of Reverse Engineering, Wiley, 2011. ISBN: 0764574817.
Additional
1. Noam Nisan, Shimon Schocken, The Elements of Computing Systems: Building a Modern Computer from First Principles, The MIT Press, 2005. ISBN: 0262640686.

Breakdown of average student's workload

	Hours	ECTS
Total workload	35	1,00
Classes requiring direct contact with the teacher	30	1,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	5	0,00