COURSE DESCRIPTION CARD - SYLLABUS

Course name
Distributed Algorithms [S2Inf1-SRC>AROZ]

Course

Field of study
Computing

Year/Semester
1/1

Area of study (specialization)
Distributed and cloud systems

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
30

Laboratory classes
0

Other
0

Tutorials
15

Projects/seminars
0

Number of credit points

4,00

Coordinators

dr hab. inż. Anna Kobusińska prof. PP
anna.kobusinska@put.poznan.pl

Lecturers

Prerequisites

Learning objectives of the first cycle studies defined in the resolution of the PUT Academic Senate that are verified in the admission process to the second cycle studies. The learning objectives are available at the website of the faculty www.cat.put.poznan.pl. In particular, students starting this course should have basic knowledge of operating systems, distributed processing and computer networks. Students should also be capable of continuous learning and knowledge acquisition from selected sources, understand the need to expand their competences, as well as express the readiness for collaborating as part of a team.

Course objective

The objective for this course is to give the students knowledge in the field of distributed operating systems, presentation of theoretical and practical aspects of the design of distributed operating systems, as well as developing students" skills in solving processing problems in a distributed environment.

Course-related learning outcomes

Knowledge:
1. students posesses well-grounded knowledge on key issues in the field of algorithms and their complexity, computer system architecture, operating systems, network technologies, programming

languages and paradigms
2. students have advanced and detailed knowledge related to selected issues from the field of it, such as: architecture and classification of distributed systems and communication environments, replication mechanisms and protocols, resource management, detection of distributed deadlock
3. students have knowledge about development trends and the most important new achievements of it field, and other related scientific disciplines
4. students have advanced and detailed knowledge of the life cycle processes of hardware or software information systems
5. students know advanced methods, techniques and tools used in solving complex engineering tasks and conducting research in a selected area of computer science related to processing in distributed operating systems
can apply a systemic approach, taking into account also non-technical aspects
4. students can assess the usefulness and the possibility of using new achievements (methods and tools) and new it products
5. students are able to assess the usefulness of methods and tools for solving an engineering task related to the construction or evaluation of an information system and its components, as well as assess the limitations of these methods and tools
6. studnets can - using, among others conceptually new methods - solve complex it problems, including non-standard problems and research problems

Skills:
1. students can obtain information from literature, databases and other sources (both in polish and english), integrate and interpret them, provide their critical evaluation, draw conclusions and formulate and exhaustively justify opinions
2. students can use analytical, simulation and experimental methods to formulate and solve engineering tasks and simple research problems
3. students can integrate knowledge from various areas of computer science (and, if necessary,

Social competences:
1. students understand the importance of using the latest knowledge from the field of computer science in solving research and practice problems
2. students understand the importance of popularizing the latest achievements in the field of computer science

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Learning outcomes presented above are verified as follows:
The knowledge acquired during lectures is verified during a problem-based written exam that consists of 4-5 open questions. The maximum number of points per question is 10. To pass the exam students must obtain at least 50% of the total points.
The skills acquired during the exercises are verified in the following way:
- assessment of the students" preparation for classes ("entrance" test),
- continuous assessment during each class (oral answers),
- assessment of knowledge and skills obtained during solving 2-3 tasks carried out as homework or in the form of a colloquium
It is possible to get additional points for activity during classes, especially during discussing additional aspects of the considered problems. Passing score: 50% of total points.

Programme content

Lectures cover the following topics: basic characteristics of distributed systems, systems architectures and their classification, basic design problems. Communication methods in a distributed environment, distributed Shared Memory (DSM) systems, replication in distributed mobile systems, resource management, distributed deadlock detection

Course topics

Lectures cover the following topics:
- Introduction: basic characteristics of distributed systems, systems architectures and their classification,

basic design problems.
- Communication methods in a distributed environment: system model, basic communication mechanisms, broadcast (reliable, uniform, causal, FIFO, total order, epidemic and gossip protocols), publish-subscribe model .
- Distributed Shared Memory (DSM) systems: consistency models (atomic, sequential, causal, PRAM, synchronized access models), consistency protocols providing the discussed consistency models.
- Replication in distributed mobile systems: client-centric consistency models (session guarantees), consistency protocols.
- CAP theory, different approaches to theory interpretation, examples illustrating the application, more precise definition - PACELC theory.
- Synchronization: physical clocks, clock synchronization algorithms, mutual exclusion, election algorithms.
- Resource management: characteristics of distributed resources, scheduling of distributed processes, problems of load balancing, load balancing algorithms and their classification.
- Distributed deadlock detection: deadlock models (AND, OR, "k out of n"), deadlock detection protocols (for the AND, OR - diffusion processing, "k out of r" model for a synchronous environment , "k out of n" model r " for an asynchronous environment , two-phase deadlock detection algorithm)
During tutorials students discuss the following topics: the application of message broadcasting mechanisms, the proofs of the correctness of broadcasting algorithms, and the communication complexity of these algorithms. They also analyze the differences between data-centric and client-centric consistency models. Moreover, mutual exclusion and deadlock detection protocols and proofs of their correctness are presented.

Teaching methods

1. Lectures: multimedia presentation, illustrated with examples given on the blackboard.
2. Tutorials: a multimedia presentation illustrated with examples given on the blackboard and carrying out the tasks given by the teacher - practical exercises.

Bibliography

Basic
1. Distributed Operating Systems - Concepts and Design, P. K. Sinhal, IEEE Press, 1997
2. Distributed Systems: Principles and Paradigms, A. S. Tanenbaum, M. van Steen, Prentice-Hall, Inc, 2007
3. Ocena stanu globalnego w systemach rozproszonych, J.Brzeziński, OWN, 2001
4. Distributed algorithms, Nancy A. Lynch, Morgan Kaufmann Publishers, San Francisco 1996
5. Distributed Operating Systems, The Logical Design, A. Gościński, Addison Wesley, 1991
Additional
1. Computer Networks, A. S. Tanenbaum, Pearson Education, Inc, 2003
2. Modern Operating Systems, A. S. Tanenbaum, Prentice-Hall, Inc, 2006
3. Advanced Concepts in Operating Systems -Disitributed, Database, and Multiprocessor Operating Systems, M. Singhal, N. G. Shivaratri, McGraw Hill, 1994
4. Distributed Computing Principles Algorithms and Concepts, M. Singhal, A.D Kshemkalyani, Cambridge University Press, 2008
5. Operating Systems Concepts, A. Silberschatz, J. Peterson, P. Galvin, Addison Wesley, 1991
6. Introduction to Distributed Algorithms, G. Tel, Cambridge University Press,, 1994

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