POZNAN UNIVERSITY OF TECHNOLOGY

EUROPEAN CREDIT TRANSFER AND ACCUMULATION SYSTEM (ECTS)

pl. M. Skłodowskiej-Curie 5, 60-965 Poznań

COURSE DESCRIPTION CARD - SYLLABUS

Course name

Operating Systems

Course

Field of study	Year/Semester
Bioinformatics	2/3
Area of study (specialization)	Profile of study
	general academic
Level of study	Course offered in
First-cycle studies	Polish
Form of study	Requirements
full-time	elective

Number of hours

Lecture	Laboratory classes	Other (e.g. online)
30	30	
Tutorials	Projects/seminars	

Number of credit points

4

Lecturers

Responsible for the course/lecturer:
Dariusz Wawrzyniak
Institute of Computing Science, PUT
ul. Piotrowo 2, 60-965 Poznań
e-mail: Dariusz.Wawrzyniak@cs.put.poznan.pl

Responsible for the course/lecturer:
Anna Kobusińska
Institute of Computing Science, PUT
ul. Piotrowo 2, 60-965 Poznań
e-mail: Anna.Kobusinska@cs.put.poznan.pl

Prerequisites

The student starting this module should have a basic knowledge of the computer structure and its working principle, selected elements of discrete mathematics, imperative programming skills, including implementation of simple algorithms and their complexity assessment. With respect to social skills, the student should show attitudes as honesty, responsibility, perseverance, curiosity, and creativity.

Course objective

1. To acquaint students with theoretical and practical problems of the design and implementation of operating systems, especially resource management.

2. To acquaint students with the concept of processor scheduling, memory management, IO handling and file system organisation.

3. To develop basic skills of computer system management, including resource and information protection.

4. To develop basic skills of efficient data processing organisation and optimal resource utilisation.

Course-related learning outcomes

Knowledge

1. Has theoretical knowledge of operating systems related issues.
2. Has basic knowledge of algorithms and data structures, as well as computational complexity theory.
3. Has basic knowledge of combinatorial optimisation algorithms and their application.
4. Has basic knowledge of computer systems life cycle.

Skills

1. Is able to design a computer program based on operating system services, following a given specification, using appropriate methods, techniques and tools.
2. Is able to carry out an analysis of functionality and requirements of information processing systems in respect of operating system services.
3. Is able to gain information from literature, databases and other information sources (both in the native language and English).

Social competences

1. Understands the need for learning throughout their lives and enhance their competence.
2. Is able to collaborate and cooperate in a team fulfilling different roles.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Formative assessment

a) Lectures:

- based on answers to questions related to the issues discussed at previous lectures.

b) Laboratory classes:

- evaluation of the student’s preparation for each laboratory session, and their skills associated with the performance of laboratory tasks,
- evaluation of knowledge and skills acquired at the laboratory classes based on two written tests in the semester.

Total assessment

a) Lectures:

- Evaluation of acquired knowledge based on the written exam consisting of 4 – 5 open-end questions with about 20 – 30 points to score for each question, agregating to 100 points for the whole exam. To get a passing grade in the exam a student must earn a minimum of 50% of the maximum score (i.e. 50 points).
- Discussion (on demand) of correct answers to the exam questions.

b) Laboratory classes:

- calculation of the evaluation in the form of a weighted arithmetic average: the weight of each of the two written tests conducted in a semester is 5, the weight of entrance tests is 2, and the weight obtained in the result of the evaluation of student’s knowledge necessary to prepare, and carry out the lab tasks is 1.

Additional elements cover:
• discussing more general and related aspects of the class topics,
• effective use of the knowledge gained during solving the given problem,
• comments leading to the improvement of the teaching materials and teaching process.

Programme content

The lecture covers the following topics:

The definition and the functions of the operating system, the classification of operating systems, system software structure and its relationship with the hardware, the principle of operation of the system kernel. The overall concept of computer system management. Processor management: CPU scheduling, scheduling criteria and algorithms. Memory management: the evolution of memory organization, memory allocation, creation of process image in memory, paging and segmentation, virtual memory. Management of I/O devices: classification of input/output devices, the structure of the I/O mechanism, the interaction between CPU and I/O devices, buffering and spooling. File system — logical organization: the definition of the file and its attributes, access methods to a file, the interface for file operations, logical directory structure. File system — physical organization: disk block allocation (contiguous, chained, and indexed), free space handling (bit vector, linked list, grouping, counting), the implementation of a directory (linear list, hash table, index structure), implementation of file operations (buffer cache, the problem of integrity, concurrent access to a file).

Laboratory classes are divided into fifteen two-hour periods, conducted in the computer laboratory under the control of a Unix-like operating system. The first laboratory session is devoted to introducing the students to the principles and the evaluation of the laboratory classes. Tasks during the classes are conducted by each student individually.

The laboratory classes cover the following topics: kernel routines for file and process handling, inter-process communication through named and unnamed pipes. Signal handling.

Teaching methods

1. Lectures: presentation of slides (multimedia showcase), discussion of problems, solving tasks on blackboard.
2. Classes: solving tasks, practical exercises, discussion, conducted in a computer laboratory (under the control of Unix-like operating system), teamwork.

Bibliography

Basic

1. A. Silberschatz, G. Gagne, P.B. Galvin Podstawy systemów operacyjnych, WN PWN, W-wa, 2021.
2. W. Stallings, Systemy operacyjne. Architektura, funkcjonowanie i projektowanie, wyd. 9, Helion, 2018.
3. A. S. Tanenbaum, Systemy operacyjne, wyd. 4, Helion, 2015.
4. M. J. Rochkind, Programowanie w systemie Unix dla zaawansowanych, WNT, Warszawa, 2007.

Additional

1. M. Ben-Ari, Podstawy programowania współbieżnego i rozproszonego, WNT, W-wa, 2016.
2. Z. Weiss, T. Gruźlewski, Programowanie współbieżne i rozproszone w przykładach i zadaniach, WNT, W-wa, 1993.

Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4,0
Classes requiring direct contact with the teacher	60	2,5
Student's own work (literature studies, preparation for laboratory classes/tutorials, preparation for tests/exam, project preparation) [1]	40	1,5

[1] delete or add other activities as appropriate