COURSE DESCRIPTION CARD - SYLLABUS

Course name
Computer architecture and operating systems [S1Cybez1>AKiSO]

Course

Field of study
Cybersecurity

Year/Semester
1/2

Area of study (specialization)
–

Profile of study
general academic

Level of study
first-cycle

Course offered in
Polish

Form of study
full-time

Requirements
compulsory

Number of hours

Lecture
30

Laboratory classes
30

Other
0

Tutorials
0

Projects/seminars
0

Number of credit points

5,00

Coordinators

dr inż. Dariusz Wawrzyniak
dariusz.wawrzyniak@put.poznan.pl

Lecturers

Prerequisites

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

Course objective

The lecture covers issues related to the functioning of operating systems and related elements of computer system architecture in the management of basic resources for application processing (including their protection), such as the processor, memory, input-output devices, and file system. Laboratory classes deal with the use of a Unix-like operating system in terms of the text interface and include file system manipulation, process handling, the organization of pipelined processing (including filters), writing scripts for the command interpreter, and elements of environment configuration.

Course-related learning outcomes

Knowledge:
1. Has theoretical knowledge of operating system related issues.

2. Has basic knowledge of computer architectures.
3. 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 perform an analysis of the functionality and requirements of information processing systems with respect to operating system services.
3. Is able to obtain information from the literature, databases, and other sources (both in the native language and in English).

Social competences:
1. Understands the need for lifelong learning and improving one's competence.
2. Is able to interact and work in a group, taking different roles in it.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Formative assessment
With respect to lectures, verification of the assumed learning outcomes is achieved by answering questions regarding the previously discussed material.
In respect of laboratories, verification of the assumed learning outcomes is achieved by evaluation of the student's preparation for individual sessions of laboratory classes and evaluation of skills related to the laboratory exercises.
Total assessment
In respect of lectures, verification of the assumed learning outcomes is achieved by assessing the acquired knowledge based on an examination test comprising several open-ended or close-ended multiple-choice questions with varied scoring, with a total of about 100 points; to get a passing grade, a student must earn a minimum of 50% of the total points (i.e. about 50 points).
With respect to laboratories, the verification of the established learning outcomes is achieved by assessing the knowledge and skills related to the material of laboratory classes through a final colloquium or 2 - 3 partial colloquia, containing single-choice, multiple-choice or open-ended questions.
Activity during classes is rewarded with additional points, in particular for:
• discussion of additional aspects of the issue;
• effective application of the acquired knowledge to the given problem to be solved;
• comments leading to the improvement of teaching materials or the teaching process.
In each form of the course assessment, the grade depends on the number of points the student earns relative to the maximum number of required points. Earning at least 50% of the possible points is a prerequisite for passing. The relationship between the grade and the number of points is defined by the Study Regulations. Additionally, the course completion rules and the exact passing thresholds will be communicated to students at the beginning of the semester through the university’s electronic systems and during the first class meeting (in each form of classes).

Programme content

The lecture program covers the following issues:
1. definition and functions of an operating system, classification of operating systems, structure of system software and its relationship to hardware;
2. the principle of the operating system kernel, including the hardware architecture components necessary for its stable functioning;
3. general concept of computer system resource management;
4. processor management and processor time allocation scheduling;
5. operational memory management, including support at the hardware architecture level;
6. virtual memory implementation;
7. input/output device management;
8. file system: logical and physical organization, implementation examples.
The laboratory exercises program focuses on the use of a Unix-like operating system in the following ways:
1. basic concepts and principles of working in the system;
2. file system operation;

3. protection of information in files;
4. operation of processes;
5. filters and pipeline processing;
6. application programs;
7. configuration of the working environment;
8. batch processing (scripts).

Course topics

The lecture deals with the following topics:
1. definition and functions of an operating system, classification of operating systems, structure of system software, and its relationship to hardware;
2. elements of computer system architecture, key elements of the operating system kernel: command cycle execution, interrupt system, interrupt sources;
3. the principle of the operating system kernel and the hardware support for its implementation;
4. the concept of process and the general concept of computer system resource management in a multitasking operating system;
5. processor management: processor time allocation scheduling, ranking criteria; processor allocation scheduling algorithms;
6. operating memory management: evolution of memory organization, memory allocation, process image creation in memory;
7. hardware support in memory organization: paging and segmentation;
8. virtual memory implementation: missing page error, performance efficiency, page replacement problem, page replacement algorithms;
9. management of input/output devices: classification of input/output devices, structure of the I/O mechanism, CPU interaction with input output devices, caching and spooling;
10. file system:
a. logical organization: file definition and attributes, file access methods, file operation interface, logical directory structure;
b. physical organization: disk block allocation (continuous, string, and index), free space management (bit vector, linked list, clustering, counting), directory implementation (linear list, hash table, index structure);
c. implementation of file operations (cache cache, integrity problem, concurrent file access);
d. implementation examples: CP/M, DOS, ISO 9660, Unix, NTFS.
The laboratory classes cover the following topics related to the use of a Unix-like operating system:
1. introduction: login, command interpreter, terminal as a device, users, groups, file system organization including hierarchical directory structure, use of help;
2. basic operations on files and directories (copying, relocating, deleting, linking), filename patterns, and searching the file system (find);
3. access rights and information protection rules: interpretation of access rights - r, w, x - to regular files and directories, methods of changing access rights, effective user/group ID;
4. process handling: listing processes, terminating, changing priority, and the ways of launching processes (sequential, concurrent, background, conditional, with stream redirection, pipeline);
5. filters: head, tail, more, grep, cut, tr, sort, uniq, sed, etc.;
6. utility programs: cat, cmp, comm, wc, vi editor, etc.;
7. shell: local and environment variables, aliases, and functions;
8. scripts: conditional constructs, loops, select statements, handling special variables (including positional parameters).
The classes are held in a computer laboratory. The exercises are carried out individually by the students.

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, H. Bos, Systemy operacyjne, wyd. 5, Helion, 2024.
4. J. Marczyński, UNIX użytkowanie i administrowanie, wyd. 2, Helion, Gliwice, 2011.

Additional:
1. G. Nutt, Operating Systems. A Modern Perspective, Addison Wesley Longman, Inc., 2002.
2. B. Goodheart, J. Cox, Sekrety magicznego ogrodu. UNIX System V Wersja 4 od środka, WNT, W-wa, 2001.
3. U. Vahalia, Jądro systemu UNIX, WNT, W-wa, 2001.
4. P. Yosifovich, A. Ionescu, M.E. Russinovich, D.A. Solomon, Windows od środka, wyd. 7, Helion, 2018.
5. C. Sobaniec, System operacyjny Linux - przewodnik użytkownika, Nakom, Poznań, 2002.
6. A.S. Tanenbaum, Strukturalna organizacja systemów komputerowych. Wydanie V, Helion, Gliwice, 2006.
7. D. Wawrzyniak, Systemy operacyjne i sprzęt informatyczny, W: Informatyka gospodarcza, A. Gąsiorkiewicz, K. Rostek, J. Zawiła-Niedźwiecki (red.), Wydawnictwo C.H. Beck, Warszawa, 2010.

Breakdown of average student's workload

	Hours	ECTS
Total workload	125	5,00
Classes requiring direct contact with the teacher	60	2,50
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	65	2,50