SYLLABUS

Operating Systems

University year 2025-2026

1. Information regarding the programme

1.1. Higher education institution	Babeş-Bolyai University
1.2. Faculty	Mathematics and Computer Science
1.3. Department	Computer Science
1.4. Field of study	Computers and information technology
1.5. Study cycle	Bachelor
1.6. Study programme/Qualification	Information Engineering
1.7. Form of education	Full time

2. Information regarding the discipline

2.1. Name of the dis	cipline	Operating S	Operating Systems			MLE5007
2.2. Course coordinator Assoc. prof. phd. Sanda-Ma			a-Maria AVRAM			
2.3. Seminar coordinator			Assoc. prof. phd. Sanda	Assoc. prof. phd. Sanda-Maria AVRAM		
2.4. Year of study	2	2.5. Semester	4	2.6. Type of evaluation E	2.7. Discipline regime	mandatory

3. Total estimated time (hours/semester of didactic activities)

3.1. Hours per week	5	of which: 3.2 course	2	3.3 seminar/laboratory/project	3
3.4. Total hours in the curriculum	70	of which: 3.5 course	28	3.6 seminar/laboratory/project	42
Time allotment for individual study (ID) and self-study activities (SA) hours					
Learning using manual, course support, bibliography, course notes (SA) 25					25
Additional documentation (in libraries, on electronic platforms, field documentation) 10					10
Preparation for seminars/labs, homework, papers, portfolios and essays					22
Tutorship					5
Evaluations					15
Other activities:				3	
3.7. Total individual study hours 80					
3.8. Total hours per semester150					
3.9. Number of ECTS credits6					

4. Prerequisites (if necessary)

4.1. curriculum	
4.2. competencies	Minimum knowledge of standard C programming.

5. Conditions (if necessary)

5.1. for the course	Class room equipped with video projector.
5.2. for the seminar /lab activities	• Laboratory with computers connected to the Internet and UNIX operating system or access to a UNIX server

6.1. Specific competencies acquired ¹

Professional/ essential competencies	 Problem solving using specific computer science and computer engineering tools Designing hardware, software and communication components
Transversal competencies	 Honorable, responsible, ethical behavior, in the spirit of the law, to ensure the professional reputation Demonstrating initiative and pro-active behavior for updating professional, economical and organizational culture knowledge

6.2. Learning outcomes

Knowledge	 The graduate has knowledge of the basics of programming specific to operating systems and has basic knowledge in programming The graduate knows the concepts related to software modeling and can implement functional and non-functional requirements described in specific documents The graduate knows the methods of testing and verifying software systems
Skills	 The graduate has the necessary skills to install and configure operating systems The graduate is familiar with tools used for testing, debugging, and validating software applications
Responsibility and autonomy:	 The graduate is able to identify complex issues and examine related issues to design and implement solutions The graduate performs testing and qualitative evaluation of functional and non-functional characteristics of systems

7. Objectives of the discipline (outcome of the acquired competencies)

7.1 General objective of the discipline	• Assimilation by the student of the main concepts underlying operating systems.
_	

¹ One can choose either competences or learning outcomes, or both. If only one option is chosen, the row related to the other option will be deleted, and the kept one will be numbered 6.

	• Learning the main facilities offered by the Unix operating system.
7.2 Specific objective of the discipline	• Shell programming and text file processing skills under Unix.
-	 Managing multitasking applications using Unix processes.

8. Content

8.1 Course	Teaching methods	Remarks
 1-3 Unix OS: external interfaces General operating system structure Regular expressions, file specification, generic specification Filters; general principles sort, awk, sed, sed, grep sh, csh, ksh, bash; general introduction Useful shell commands and external process management Shell programming; shell applications The structure of directories in Unix system Mount-ing concept Symbolic and hard links 	Exposure: description, explanation, examples, discussion of case studies	
 4-7 Unix operating system: system calls, internal structures Files and processes under Unix I/O in C POSIX: open, close, lseek, read, write, dup, dup2 File protection Processes under Unix; structure of a process Process management system calls: fork, wait, exit, exec* Communication between processes: pipe, popen, FIFO POSIX threads 	Exposure: description, explanation, examples, discussion of case studies	
 8-9 Filesystems General disk management and file systems Scheduling magnetic disk access DOS disk and file system internal structure; FAT table WindowsNT & 2000 disk and file system internal structure; NTFS mechanism, MFT file Unix disk and file system internal structure; i-node mechanism 	Exposure: description, explanation, examples, discussion of case studies	
 10-14 General Theory of operating systems Types of computer systems and operating systems. I/O channel, multiple buffers. Multiprogramming. General structure and functions of an operating system Processes: specification, concurrency, semaphores, deadlock Process scheduling Memory management Scheduling swap between internal and secondary memory 	Exposure: description, explanation, examples, discussion of case studies	

Bibliography

In English:

- 1. Albing, C., Vossen, J.P., Newhman, C., bash Cookbook: Solutions and Examples for bash Users, O'Reilly, USA, 2007.
- 2. Kernighan, B.W., Dennis, R.M., The C Programming Language, Prentice Hall, Massachusetts, 2012.
- 3. Stallings, W., Operating Systems: Internals and Design Principles, Pearson Education Limited, Essex, 2015.
- 4. Raymond, E.S., The Art of UNIX Programming, Addison-Wesley, Pearson Education Limited, USA, 2004.
- 5. Tanenbaum, A., Herbert, B., Modern Operating Systems, Pearson Education Limited, Essex, 2015.

In Romanian:

6. Boian, F., Vancea, A., Boian, R., Bufnea, D., Sterca, A., Cobarzan, C., Cojocar, D., Sisteme de operare, Ed. Risoprint, Cluj-Napoca, 2006.

8.2 Seminar	Teaching methods	
1. Unix commands and text-editors	Dialogue, case studies,	
2. sed and grep	examples	
3. awk		
4. UNIX Processes		
5. Communications between Unix processes: pipe		
6. Communications between Unix processes: FIFO		
7. Revision		
8.2 Laboratory	Teaching methods	Remarks
1-2. Unix commands for working with files	Dialogue, debate, case	
3. Shell 1	studies, examples, proofs	
4. sed and grep utilities		
5. awk utility		
6. shell Programs	~	
7-8. C programs; working with Unix files		
9. UNIX Processes		
10. Communications between Unix processes: pipe		
11. Communications between Unix processes: FIFO	-	
12. Unix-Threads		
13. Closing lab activities		
14. Practical exam		

Bibliography

- 1. Albing, C., Vossen, J.P., Newhman, C., bash Cookbook: Solutions and Examples for bash Users, O'Reilly, USA, 2007.
- 2. Kernighan, B.W., Dennis, R.M., The C Programming Language, Prentice Hall, Massachusetts, 2012.
- 3. **Stallings, W.**, Operating Systems: Internals and Desing Principles, Pearson Education Limited, Essex, 2015.
- 4. Raymond, E.S., The Art of UNIX Programming, Addison-Wesley, Pearson Education Limited, USA, 2004.
- 5. Tanenbaum, A., Herbert, B., Modern Operating Systems, Pearson Education Limited, Essex, 2015.

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

- This course exists in the study program of all major universities in Romania and abroad.
- This course provides the basic knowledge that any system administrator or programmer must have.

10. Evaluation

Activity type	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage of final grade	
10.4 Course	- knowledge of the basic principles of the field	Final exam (regular session)	40%	
10.5 Seminar/ laboratory	- applying these concepts in problem-solving	Lab assignments (during the semester)	20%	
	- developing shells and creating Unix processes	Practical exam (last week of the semester)	40%	
10.6 Minimum standard of performance				

• At least grade 5 (from a scale of 1 to 10) for all types of examination.

- Seminar attendance of minimum 75% (at least 5 seminars out of 7)
- Lab attendance of minimum 90% (at least 12 out of 14 labs)
- Knowledge of theoretical and practical aspects of **shell** concepts and **processes**:
 - shell: working with files, control structures (especially for), access to command line parameters;
 - processes: one-way communication via *pipe* or *FIFO*.

11. Labels ODD (Sustainable Development Goals)²

Not applicable.

Date:	Signature of course coordinator	Signature of seminar coordinator
15.04.2025	Assoc. prof. phd. Sanda-Maria AVRAM	Assoc. prof. phd. Sanda-Maria AVRAM

² Keep only the labels that, according to the *Procedure for applying ODD labels in the academic process*, suit the discipline and delete the others, including the general one for *Sustainable Development* – if not applicable. If no label describes the discipline, delete them all and write *"Not applicable.*".

Date of approval:

...

Signature of the head of department

Assoc. prof. phd. Adrian STERCA