COURSE OUTLINE

(1) GENERAL

SCHOOL	ENGINEERING		
ACADEMIC UNIT	ELECTRICAL AND COMPUTER ENGINEERING DEPT.		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	**ECE_K540**	**SEMESTER**	**5**
COURSE TITLE	Operating Systems		

INDEPENDENT TEACHING ACTIVITIES *if credits are awarded for separate components of the course, e.g. lectures, laboratory exercises, etc. If the credits are awarded for the whole of the course, give the weekly teaching hours and the total credits*	**WEEKLY TEACHING HOURS**	**CREDITS**
Lectures	2	
Seminars / Practice exercises	1	
Laboratory	1	
Add rows if necessary. The organisation of teaching and the teaching methods used are described in detail at (4).	4	5

COURSE TYPE *general background, special background, specialised, general knowledge, skills development*	Special Background.
PREREQUISITE COURSES:	There are no prerequisite courses. Recommended background knowledge: Functional Programming, Computer Architecture
LANGUAGE OF INSTRUCTION and EXAMINATIONS:	Greek
IS THE COURSE OFFERED TO ERASMUS STUDENTS	Yes
COURSE WEBSITE (URL)	https://www.ece.uop.gr/

(2) LEARNING OUTCOMES

Learning outcomes

The course learning outcomes, specific knowledge, skills and competences of an appropriate level, which the students will acquire with the successful completion of the course are described.

Consult Appendix A

- *Description of the level of learning outcomes for each qualifications cycle, according to the Qualifications Framework of the European Higher Education Area*
- *Descriptors for Levels 6, 7 & 8 of the European Qualifications Framework for Lifelong Learning*

and Appendix B

- *Guidelines for writing Learning Outcomes*

The main purpose of the course is to understand the operation of modern Operating Systems (OS), to get acquainted with the most popular OS and to introduce into the basic concepts of system programming.

Keywords: Types of OSs, Process Synchronization, Mutual Exclusion, Memory Management, Scheduling, System Programming.

Learning Outcomes

Upon successful completion of the course, student will be able to:

At the knowledge level:

- describe in detail the functions of an OS.
- describe the different ways of processes scheduling and explain their differences and the advantages-disadvantages of each.
- understand the problem of mutual exclusion and process synchronization and the tools available to implement them.
- understand the used memory management strategies such as paging, segmentation and their combinations.
- know the way of operation / organization of virtual memory and distinguish the various page replacement algorithms that support the modern operating systems.

.

At the skill level:

- be able to use modern popular LS effectively
- be able to use the basic synchronization and mutual exclusion mechanisms of a modern operating system.
- be able to apply the most efficient memory management techniques considering the user requirements and the system architecture.
- be able to apply the most efficient time scheduling techniques taking into account the specific requirements.

At the level of abilities:

- evaluate the various OSs and select the most appropriate one, depending on the user and application requirements.
- have the ability to design and implement (in introductory level) concurrent/parallel programs/applications at the level of programming languages and/or operating systems.
- have the ability to design and implement applications in the OS shell.
- have the ability to design and implement basic system programming applications

General Competences

Taking into consideration the general competences that the degree-holder must acquire (as these appear in the Diploma Supplement and appear below), at which of the following does the course aim?

Search for, analysis and synthesis of data and information, with the use of the necessary technology	*Project planning and management*
Adapting to new situations	*Respect for difference and multiculturalism*
Decision-making	*Respect for the natural environment*
Working independently	*Showing social, professional and ethical responsibility and sensitivity to gender issues*
Team work	*Criticism and self-criticism*
Working in an international environment	*Production of free, creative and inductive thinking*
Working in an interdisciplinary environment	*……*
Production of new research ideas	*Others…*
	…….

- Search for, analysis and synthesis of data and information, with the use of the necessary technology
- Working independently
- Team Work
- Working in an international environment
- Production of free, creative and inductive thinking
- Production of new research ideas

(3) SYLLABUS

The course is developed in the following 13 lectures:

1. Introduction to Operating Systems. Definitions, basic principles, structure, types, history.
2. Processes and threads. Interrupts and the kernel.

3. CPU scheduling. Scheduling requirements, performance evaluation criteria.
4. CPU scheduling. Scheduling algorithms and their performance.
5. Familiarity with scheduling algorithms using case studies and simulation use.
6. Concurrent and parellel processes. Precedence graphs, critical regions.
7. Mutual exclusion. Techniques and tools for mutual exclusion.
8. Process synchronization. Techniques and tools for process synchronization.
9. Familiarity with process synchronization tools, offered by modern OSs and/or the most popular programming languages, System programming.
10. Memory management. Introduction, swapping, memory management with fixted and variable partitions, fragmentation.
11. Memory management. Paging, segmentation. Compined systems, two-level paging.
12. Memory management. Virtual memory, introduction, page replacement algorithms, thrasing.
13. Memory management. Case study.

(4) TEACHING and LEARNING METHODS - EVALUATION

DELIVERY *Face-to-face, Distance learning, etc.*	Face to face in class and in the laboratory. Distance learning support via e-Class system.
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY *Use of ICT in teaching, laboratory education, communication with students*	• Supporting the learning process through the e-Class platform (for notification of the course regulations, for distribution of slides, laboratory exercises, supplementary material, announcements, links, bibliography, etc.). • During the lectures of the theoretical part, a projector and presentations in electronic form are used, which are also posted on the eclass from the beginning of the semester. • During the lectures, a computer is used to write and execute code. • Use of specialized software like popular OS, simulators, programming languages (C, Python, Java)
TEACHING METHODS *The manner and methods of teaching are described in detail.* *Lectures, seminars, laboratory practice, fieldwork, study and analysis of bibliography, tutorials, placements, clinical practice, art workshop, interactive teaching, educational visits, project, essay writing, artistic creativity, etc.* *The student's study hours for each learning activity are given as well as the hours of non-directed study according to the principles of the ECTS*	

Activity	***Semester workload***
Theory Lectures	26
Seminars / Practice exercises	13
Laboratory Exercises	13
Preparation of laboratory exercises	13
Projects preparation	25
Independent study of lectures and bibliography	35
Course Total	***125 hours (5 ECTS)***

STUDENT PERFORMANCE EVALUATION *Description of the evaluation procedure* *Language of evaluation, methods of evaluation, summative or conclusive, multiple choice questionnaires, short-answer questions, open-ended questions, problem solving, written work, essay/report, oral examination, public presentation, laboratory work, clinical examination of patient, art interpretation, other* *Specifically-defined evaluation criteria are given, and if and where they are accessible to students.*	A. Written final exam that includes: • Solve exercises • Multiple choice questions • Short answer questions B. Preparation of laboratory exercises and project. <u>Remarks:</u> • The final grade results from the weighting of the theory and work grades with coefficients determined at the beginning of the semester and announced to the students via e-class. Indicatively it will be about 60% - 40% • Laboratory exercises and assignments will be submitted electronically and students will be asked to take an oral exam on them. • The exam material and the evaluation process are communicated to the students in the lecture hall and in the e-class.

(5) ATTACHED BIBLIOGRAPHY

- *Suggested bibliography:*

- Silberschatz A., Galvin P., Gagne G., *Λειτουργικά Συστήματα*, 9η έκδοση, Εκδόσεις Μ. Γκιούρδα (μετάφραση), 2013.
- Stallings W., *Λειτουργικά Συστήματα*, 9η έκδοση, Εκδόσεις Α. Τζιόλα & Υιοί ΑΕ (μετάφραση), 2018.
- Tanenbaum A., *Σύγχρονα Λειτουργικά Συστήματα*, 3η έκδοση, Εκδόσεις Κλειδάριθμος ΕΠΕ (μετάφραση), 2009.

- *Related academic journals and communities:*

- ACM SIGOPS Operating Systems Review
- ACM Symposium on Operating Systems Principles
- IEEE Computer Society Special Technical Community on Operating Systems (STCOS)