COURSE OUTLINE

(1) GENERAL

SCHOOL	School of Engineering		
ACADEMIC UNIT	Informatics and Computer Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE		**SEMESTER**	4th
COURSE TITLE	Operating Systems II		

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**
	4	4
Add rows if necessary. The organisation of teaching and the teaching methods used are described in detail at (d).		

COURSE TYPE general background, special background, specialised general knowledge, skills development	Specialised general knowledge, skills development
PREREQUISITE COURSES:	
LANGUAGE OF INSTRUCTION and EXAMINATIONS:	Greek
IS THE COURSE OFFERED TO ERASMUS STUDENTS	Yes (in English)
COURSE WEBSITE (URL)	https://eclass.uniwa.gr/courses/CS122/

(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 course aims to present the basic elements and mechanisms of Multiprocessor, Multicomputer, Network and Real-Time Operating Systems, as well as to cover advanced concepts and mechanisms of communication and process synchronization in a computer system, such as e.g. the theory of deadlock prevention and avoidance, inter-process communication tools and mechanisms, thread management etc. A more specific goal from a practical point of view is to acquaint the students with practical issues and applications of concurrent programming, process communication and thread programming.

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

- Recognize the basic characteristics of operating systems in multiprocessor shared/distributed memory and network environments (multiprocessors, multi-computers and distributed systems with limited resources), as well as their main differences comparing to conventional operating systems.
- To explain how the basic concepts and mechanisms of an operating system (CPU scheduling, memory management, process synchronization, etc.) are specialized in multi-processor systems.
- Describe the characteristics and differences of real-time operating systems comparing to traditional interactive and batch processing operating systems.
- Understand the special requirements and significant advantages of the thread computing model for programming in single or multi-core environments.
- To delve into the necessity of advanced process synchronization and communication

mechanisms both in theory (e.g. deadlock avoidance and prevention) and in practice (e.g. communication via sockets).

- To take advantage of the possibilities offered by the virtualization technique for more efficient use of computer systems.
- To recognize the special requirements that govern resource-limited computing environments (mobile devices, sensor nodes etc.) in terms of operating system services.
- To use in practice basic system programming mechanisms and tools, concurrent processes and threads, in a unix/linux type operating system.

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
Adapting to new situations
Decision-making
Working independently
Team work
Working in an international environment
Working in an interdisciplinary environment
Production of new research ideas

Project planning and management
Respect for difference and multiculturalism
Respect for the natural environment
Showing social, professional and ethical responsibility and sensitivity to gender issues
Criticism and self-criticism
Production of free, creative and inductive thinking
……
Others…
…….

- Autonomous work
- Team work
- Adaptation to new situations
- Work in an interdisciplinary environment
- Production of new research ideas
- Promotion of free, creative and inductive thinking
- Search for, analysis and synthesis of data and information, using the necessary technologies

(3) SYLLABUS

Theory:

- Multiprocessor and Multicomputer Operating Systems (special requirements and operating system design issues for multiprocessor environments, process communication and synchronization methods, process scheduling algorithms, memory schemes and memory management methods, load distribution across multiple processors, etc.).
- Advanced concepts of process communication-synchronization (theory of deadlock prevention and avoidance, inter-process communication through pipes and sockets, etc.).
- Thread Management in multiprocessing environments (special requirements and scheduling algorithms, special requirements and synchronization-communication mechanisms, etc.).
- Real-Time Operating Systems (basic concepts and design issues, special algorithms for CPU scheduling, memory organization-management and disk scheduling, application in embedded systems and multimedia systems, etc.).
- Special requirements of operating systems of mobile devices and environments with limited resources (sensor networks, etc.).
- Virtual machines - virtualization: definitions, categorization, virtualization techniques, the concept of hypervisor and supported types, etc.
- Network and distributed resource operating systems (definitions, categorization, design issues, services, etc.).

Lab:

Unix/Linux: System programming and concurrent processes (child processes, use of fork, wait, exec, signals/traps, etc.). Special mechanisms to support communication and process synchronization (shared memory segments, semaphores, message queues, etc.). Programming and thread management using the Posix threads library. Interprocess communication and programming using pipes and sockets.

(4) TEACHING and LEARNING METHODS - EVALUATION

<table>
<tr><td><b>DELIVERY</b>
Face-to-face, Distance learning, etc.</td><td>Face-to-face. Use of distance learning (if required)</td></tr>
<tr><td><b>USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY</b>

Use of ICT in teaching, laboratory education, communication with students</td><td>Use of ICT in Course Teaching and Laboratory Education,
Use of ICT in Communication with Students
Post course material on the University's e-learning platform (e-class).
Use of email and e-class in communication with students.</td></tr>
<tr><td><b>TEACHING METHODS</b>
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</td><td><table>
<tr><th>Activity</th><th>Semester workload</th></tr>
<tr><td>Lectures</td><td>26</td></tr>
<tr><td>Problem Solving in Class</td><td>13</td></tr>
<tr><td>Laboratory Education</td><td>13</td></tr>
<tr><td>Lab exercises</td><td>20</td></tr>
<tr><td>Non-guided study</td><td>28</td></tr>
<tr><td>Course total</td><td>100</td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
</table></td></tr>
<tr><td><b>STUDENT PERFORMANCE EVALUATION</b>
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.</td><td>Final Grade = (70% * Grade of the Theory_Part) + (30% * Grade of the Laboratory Part)

<i>Evaluation Process of Theory Part</i>: Final written exam at the end of the semester

<i>Evaluation Process of Laboratory Part</i>: Preparation of laboratory exercises / assignments and oral or written examination

The evaluation process is disclosed to the students in class and online, via e-class.</td></tr>
</table>

(5) ATTACHED BIBLIOGRAPHY

1. Silberschatz A., Galvin P., Gagne G., *Operating Systems Concepts*, 10th edition, Wiley, 2018.
2. Stallings W., *Operating Systems: Internals and Design Principles*, 9th edition, Pearson, 2017.
3. Tanenbaum A., Bos H., *Modern Operating Systems*, 4th edition, Pearson, 2015.
4. Κάβουρας Ι., Λειτουργικά Συστήματα, 5η έκδοση, Εκδόσεις Κλειδάριθμος, 2000 (in greek).
5. Gary Nutt, Operating Systems, 3rd edition, Addison-Wesley, 2003.
6. Robert Love, Linux Kernel Development, 3rd edition, Addison-Wesley, 2010.
7. Robert Love, Linux System Programming, 2nd edition, O'Reilly Media, 2013.
8. Kernighan B., Pike R., The Unix Programming Environment, Prentice Hall, 1985.
9. Glass G., Ables K., Unix for Programmers and Users, Prentice Hall, 1998.
10. Stevens W.R., Unix Network Programming, Vol. 2: Interprocess Communications, 2nd ed., Prentice Hall, 2000.
11. Butenhof D.R., Programming with POSIX Threads, Addison-Wesley, 2000.
12. Robbins K., Robbins S., Unix Systems Programming: Communication, Concurrency and Threads, 2nd ed., Prentice Hall, 2003.