

OPERATING SYSTEMS

COURSE OUTLINE

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electrical and Computer Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0811.8.009.0	SEMESTER	2nd
COURSE TITLE	Operating Systems		
INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	WEEKLY TEACHING HOURS	CREDITS	
	0	4	
Total	0	4	
COURSE TYPE general background, special background, specialised general knowledge, skills development	Compulsory Elective		
PREREQUISITE COURSES	None		
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English		
OFFERED TO ERASMUS STUDENTS	Yes (in English)		
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/ECE147		
2. LEARNING OUTCOMES			
Learning outcomes			
The knowledge which students acquire upon successful completion of the course relates to design and implementation of modern operating systems (OSes). More specifically, the students are taught concepts related to the lifecycle of processes and threads, SystemV/POSIX shared memory, and resource sharing, focusing on inter-process communication and synchronization primitives (IPC message queues, pipes, UNIX signals, and POSIX locks, semaphores, barriers, and condition variables). They are also exposed to techniques that can detect or avoid hazards, such as data race and protocol deadlock, and guided to examine cost-efficient solutions of classical OS problems, such as producer-consumer, readers-writers, dining philosophers, and sleeping barber. Students are finally introduced to complex OS kernel functions and high-level services, related to job scheduling, virtual-to-physical address translation, memory management, file system operations, and I/O device management. The skills, which students develop upon successful course completion, relate to: • Understanding the design, implementation, operation, and high-level services of modern OSes • Understanding fundamental issues related to resource management in Linux and real-time OSes • Programming in consistent and efficient manner concurrent systems utilizing IPC • Embedding OS API functions (IPC, POSIX, SystemV) The abilities, which students develop upon successful course completion, relate to: • Enabling new problem-solving abilities, e.g. for avoiding hazards in concurrent systems • Learning shell & systems programming (C/C++ & GNU/Linux) and GNU software development			
General Competences			

- Search, analysis and synthesis of data and information, using the necessary technologies
- Adapt solutions to new situations (resource sharing, congestion, contention etc)
- Autonomous work
- Teamwork
- Decision making
- Promoting liberal, creative and inductive/deductive thinking\
- Work in an interdisciplinary environment

3. SYLLABUS

Theoretical Lectures

The theoretical part concentrates on the following topics:

- Introduction to OS Design, Architecture and Services, Examples, Open vs Free Software
- Processes & Threads, IPC, Shared Memory, Hazards (Data Races, Deadlocks), IPC functions
- Classical OS Problems and Solutions
- Job Scheduling Algorithms, Implementations
- Virtual-to-Physical Address Translation
- Memory Management, Paging Systems, Replacement Algorithms, Implementations
- File Systems, Kernel support, Examples
- I/O Device Management, Device Drivers Types
- Special OS Topics

Lab

Students are introduced to Linux OS implementation. Hands-on activities relate to shell programming and systems programming, focusing on task management and efficient use of different IPC functions. In addition through simple demos, students are exposed to sophisticated OS topics, such as kernel scheduling policies, paging and address translation (GNU/Linux page maps, analysis), file systems (simpleFS), and I/O device management (UART-to-SPI TTY driver of an LCD).

4. TEACHING and LEARNING METHODS - EVALUATION

DELIVERY Face-to-face, Distance learning, etc.	Eclass for Optional Exercises. Project Presentations/Demonstration in the Lab	
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	Using Eclass	
TEACHING METHODS The manner and methods of teaching are described in detail.	Activity	Semester workload
	Course total	
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	All announcements related to the syllabus, including complementary reading material, solved exercises, and optional homework problems, are permanently posted in the course web page (ECLASS). The course grade incorporates the following evaluation procedures: 1. Final Oral Exam on theoretical/practical problems (50%) 2. Project (50%) or Project	

5. ATTACHED BIBLIOGRAPHY

Bibliography:

- Andrew S. Tanenbaum, and H. Bos, "Modern Operating Systems", 3rd Edition, Prentice Hall, 2018.
- W.R. Stevens and S.A. Rago, "Advanced Programming in the UNIX Environment", 3rd edition, Addison-Wesley (2013), ISBN 978-0321637734.
- A. Silberschatz, P. B. Galvin, and G. Gagne, "Operating System Concepts", 9th Edition, Wiley (2013), ISBN 978-1118093757, <http://os-book.com>

Other Important Sources

- Eclass - <http://eclass.hmu.gr> (notes, examples, open source code)

Relevant Scientific Journals:

- ACM Symposium on Operating Systems Principles (SOSP)
- Operating Systems Design and Implementation (OSDI)
- Workshop on Hot Topics in Operating Systems (HotOS)
- Architectural Support for Programming Languages and Operating Systems (ASPLOS)
- Linux Foundation conferences, e.g., Open Source Forum
- Linux Plumbers Conference

