UNIVERSITÀ DEGLI STUDI DI PALERMO

DEPARTMENT	Matematica e Informatica
ACADEMIC YEAR	2022/2023
BACHELOR'S DEGREE (BSC)	COMPUTER SCIENCE
SUBJECT	OPERATING SYSTEMS
TYPE OF EDUCATIONAL ACTIVITY	B
AMBIT	50166-Discipline Informatiche
CODE	16784
SCIENTIFIC SECTOR(S)	ING-INF/05
HEAD PROFESSOR(S)	VALENTI CESARE FABIO Professore Associato Univ. di PALERMO
OTHER PROFESSOR(S)	
CREDITS	6
INDIVIDUAL STUDY (Hrs)	86
COURSE ACTIVITY (Hrs)	64
PROPAEDEUTICAL SUBJECTS	05880 - PROGRAMMING AND LABORATORY - INTEGRATED COURSE
MUTUALIZATION	
YEAR	2
TERM (SEMESTER)	2° semester
ATTENDANCE	Not mandatory
EVALUATION	Out of 30
TEACHER OFFICE HOURS	**VALENTI CESARE FABIO** Wednesday 14:30 18:30 da concordare via email

DOCENTE: Prof. CESARE FABIO VALENTI

PREREQUISITES	Basic knowledge about computer architectures and main data structures.
LEARNING OUTCOMES	- Knowledge and ability to understand. The student acquires the ability to recognize and organize independently the basic topics of the course, to use the knowledge learned in the specific field of application, with particular reference to the interfacing and modification of operating systems. - Ability to apply knowledge and understanding. The student will be able to understand and analyze the functions of an operating system and the relative choices adopted by its programmers. - Making judgments. The student will be able to provide a critical interpretation of the results obtained in relation to the investigated subject and the methodologies used. - Ability to communicate. The student learns to synthesize and expose what he learnt during the course; to adapt the language of technology to the reference context and to the interlocutor, who may be non-expert. - Learning ability. Ability to upgrade through scientific publications regarding the fields of operating systems and algorithms. Ability to attend, using the knowledge acquired during the course, second level masters, in-depth courses, specialized seminars on operating systems.
ASSESSMENT METHODS	Optional ongoing evaluation for the self-evaluation of the student. Practice test and oral examination to verify skills and disciplinary knowledge provided by the course. Both the ripeness of the examinee and the clarity of the presentation will be evaluated. Excellent (29-30 / 30 cum laude) - excellent knowledge about the topics, technical language with excellent ability, good analytical ability, the student is able to apply knowledge to solve independently all proposed problems, providing also original solutions. Very good (27-28) - very good knowledge about the topics, excellent ability of the technical language, the student is able to apply knowledge to solve independently all proposed problems. Good (24-26) - good knowledge about the topics with reasonable ability of the technical language, the student is able to apply knowledge to solve most proposed problems. Sufficient (18-23) - the student does not have full ability of the main topics but he has the underlying knowledge; just sufficient ability of the technical language with little ability to apply independently the knowledge acquired. Insufficient - the student does not have an acceptable knowledge about the contents of the topics covered during the course.
EDUCATIONAL OBJECTIVES	The course has the following objectives: - fundamentals of an operating system - structure of an operating system - process management - memory management - data space management - binary representation of information Apply knowledge of computing theory and mathematics to solve problems and present comprehensively the results and methods of the solution to either a professional or non-professional audience. Predict the behavior of systems under random events using knowledge of probability and expectation and inform users of its potential behavior. Assess the security of a system using the knowledge of confidentiality, availability, and integrity with an understanding of risks, threats, vulnerabilities, and attack vectors, and relate its societal and ethical impact to the system's constituents.
TEACHING METHODS	Lectures and lab activities.
SUGGESTED BIBLIOGRAPHY	Qualsiasi testo di livello universitario che descriva tutti gli argomenti trattati durante il corso. Le lezioni saranno supportate da presentazioni che NON sostituiscono i seguenti testi consigliati. / Any university-level book that describes all the topics discussed during the course. The lessons will be supported by presentations that do NOT supersede the following recommended texts. Testo principale / main textbook Silberschatz A., Galvin P.B., Gagne G. "Sistemi Operativi: concetti ed esempi", 10 Ed. Pearson, 2019. ISBN-10:8891904554 Per l'attivita' di laboratorio / Lab activities Kerrisk M. "The Linux Programming Interface: A Linux and UNIX System Programming Handbook", No Starch Press, 2010. ISBN-10:1593272200 Per consultazione / Further reading Tanenbaum A.S.. Bos H. "I moderni sistemi operativi", 4 Ed. Pearson, 2019. ISBN-10:8891906255

SYLLABUS

Hrs	Frontal teaching
2	Description of the main architectures and components of modern hardware and operating systems. User interfaces. API and system calls. Process management and communications.
3	Processes and threads in detail. Scheduling and synchronization.
3	Single- and multi-core processors. Competition. Models for multi-threading. Thread groups.
3	Locality of reference. Systems with and without preemption. Comparison of the main scheduling algorithms. Load balancing. Real-time scheduling.
3	Critical section. Peterson's solution. Barriers; atomic instructions; lock mutex; semaphores; monitors.
3	Classic problems of synchronization: bounded-buffer; readers–writers; dining–philosophers. Transactional memory.
3	Deadlock, livelock and starvation. Resource allocation graph. Prevention, avoidance and detection of deadlocks. Recovering from deadlock.
3	Main memory. Logical and physical addresses. Loading, allocation and protection. Fragmentation. Paging. Swapping.
3	Virtual memory. Page-replacement algorithms. Page faults. Thrashing. Working set. Kernel allocation.
3	Mass-storage memory. HDD, SSD, NAND flash. Scheduling. Error detection and correction. RAID structure.
3	Structures and allocations of the file system. Files and directories. Protection. Free-space management. Recovery.

Hrs	Workshops
4	Graphical user interface and command line interface.
4	Locality of reference and page faults simulation.
4	Simulation of scheduling algorithms.
6	Processes, threads, fork-join.
6	Typical synchronization issues.
4	Semaphores and mutex.
4	Advanced binary file management.