UNIVERSITÀ DEGLI STUDI DI PALERMO

DEPARTMENT	Matematica e Informatica
ACADEMIC YEAR	2019/2020
BACHELOR'S DEGREE (BSC)	COMPUTER SCIENCE
SUBJECT	OPERATING SYSTEMS
TYPE OF EDUCATIONAL ACTIVITY	B
AMBIT	50166-Discipline Informatiche
CODE	16784
SCIENTIFIC SECTOR(S)	INF/01
HEAD PROFESSOR(S)	VALENTI CESARE FABIO Professore Associato Univ. di PALERMO
OTHER PROFESSOR(S)	
CREDITS	9
INDIVIDUAL STUDY (Hrs)	153
COURSE ACTIVITY (Hrs)	72
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 of C programming is required. Notions of computer architectures can also be useful for a better acquisition of some of the imparted knowledge.
LEARNING OUTCOMES	The student should acquire 1) proficiency in technical language, and its use in different contexts; 2) knowledge of structure and functions of OS; 3) basic knowledge and techniques for OS programming (resource sharing, mutual exclusion, protection); 4) ability to recognise and use basic OS notions. Knowledge and understanding The student acquires the ability to recognise and organise the basic elements of knowledge, and to apply them (especially when programming) to problem solving in OS. Applying knowledge and understanding The student should be able to understand and analyse the functions of an OS, and to understand and analyse choices from the OS programmers. Making judgements The student should be able to critically discuss results obtained and methodologies used during the learning process. Communication The student should be able to: 1) synthesize and discuss the course topics; 2) adapt the technical language according to context, even to the non expert. Lifelong learning skills The student should be able to peruse scientific literature and to critically discuss and explain the problems encountered in a professional role with the non initiated.
ASSESSMENT METHODS	The final exam is in semi-structured written form. There are 25 multiple-choice questions, each with 3 possible options, and 2 open questions. Each multiple-choice question gives 1 point for correct answer, 0 points otherwise. Each open questions gives a score ranging between 0 (unanswered or completely off) to 3 (correct, concise, and with good use of technical language). The maximum score for the written test is 31, which allows to pass the exam cum laude, after a brief interview. The questions cover the whole syllabus, with a number of questions per topic that reflects the relative weight of that topic in terms of lecture hours. This choice allows to compare different student performances, and it ensure the highest level of objectivty, yet allowing the student to express the learned concepts in appropriate technical language. The overall time spent in examination is reduced to a minimum. The score ranges are qualified as follows: 18-20: knowledge of the subject and operating systems skills are sufficient; 21-23: knowledge of the subject and operating systems skills are fair; 24-25: knowledge of the subject and operating systems skills are good; 26-27: knowledge of the subject and operating systems skills are very good; 28-30 cum laude: knowledge of the subject and operating systems skills are excellent.
EDUCATIONAL OBJECTIVES	The educational objectives of the lectures are as follows: fundamentals of Operating Systems (OS) OS structure processes memory management data examples of programming via message-passing binary representations of information Lectures also give the student some instruments for critical and autonomous thinking.
TEACHING METHODS	Teaching is based on standard lectures, and on computer exercises that are used to develop the practical skills.
SUGGESTED BIBLIOGRAPHY	Testo principale / main textbook A. Silberschatz, P.B. Galvin, G. Gagne, "Sistemi Operativi: concetti ed esempi", 10/Ed. Pearson, 2019. Altri materiali / Other materials Slide e materiali utilizzati a lezione e scaricabili on line. / Lecture slides and handouts are available on line. Per consultazione / Further reading Tanenbaum, A. and Bos, H. (2014). Modern Operating Systems. Prentice Hall.

SYLLABUS

Hrs	Frontal teaching
3	Introduction, definitions
4	Parallel systems, distributed systems; Elements of computer architecture; Interrupt.
2	Communication between processors and I/O devices; Memory: types, hierarchy, caching
5	Hardware protection, user mode and kernel mode; Structures of operating systems; system calls
5	Representation of processes; PCB, code; Scheduling; CPU Scheduling Algorithms and their evaluation
5	Main memory management; Allocation methods
2	Pagination
6	Secondary storage; supported types; scheduling of secondary storage
4	file system
6	Concurrency: definitions, causes, derived problems;
6	Concurrency: race conditions; critical sections: software and hardware approaches, system and language support; classical problems (5 philosophers, producer/consumer, readers/writers)
6	Deadlock: definitions; conditions; detection and recovery; prevention
6	Concurrent programming with message-passing: blocking and non-blocking, synchronous and asynchronous communication
4	Concurrent programming: practical examples and implementation with MPI
4	Binary numeric representation of information: definitions, number representation (quantities) and base conversions; Binary numeric representation of one-dimensional time-varying signals (e.g., sound); Binary numeric representation of time-varying signals on multi-dimensional domain (e.g., images, video);
4	Compression: lossless techniques (e.g., RLE) and perceptual (lossy) techniques (e.g., JPEG)