Università degli Studi di L'Aquila - Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica

Course catalogue

Page compiled on 19/08/2019

Programme of Integrated course "Sistemi Operativi con Laboratorio"		
This course is composed of 2 Modules: 1) Laboratorio di Sistemi Operativi, 2) Sistemi Operativi		
Programme of Module "Laboratorio di Sistemi Operativi"		
• Code: F1I021 • Type of course unit: Compulsory (Bachelor Degree in Computer Science curriculum General) • Level of course unit: Undergraduate Degrees • Semester: 1		
Number of ects credits: (Bachelor Degree in Computer Science) 6 (workload 150 hours)		
Teachers: Marco Autili (marco.autili@univaq.it)		
1	**Course objectives**	The objective of this course is to provide a complete introduction to Unix-like systems. Contents are organized as follow: PART I - UNIX System Architecture PART II - Command Line (Bash shell) PART III - Bash Scripting PART IV - Programming in UNIX-like Environment The course offers a basic, yet complete, knowledge of the following practical aspects: Unix-like systems architecture, command line interaction, shell scripting, files and directories, system calls, system-programming, process management, and concurrent programming.
2	**Course content and learning outcomes (dublin descriptors)**	Topics of the module include: • PART I - UNIX System Architecture • PART II - Command Line (Bash shell) • PART III - Bash Scripting • PART IV - Programming in UNIX-like Environment On successful completion of this module, the student should : • acquire the theoretical knowledge necessary to understand how operating systems can implement their main functionalities; • acquire the practical capabilities necessary to develop system programs and bash scripts in Unix-like environments; • will be able to develop concurrent programs using semaphores, mutex and condition variables for synchronization; • acquire methodologies to evaluate different operating systems bny integrating all the notions acquired during the course; • be able to communicate with competence and correctness of language the issues related with operating systems and systems programming; • be able to autonomously learn and study specific additional subjects related to operating systems.
3	**Course prerequisites**	Topics treated by the Operating Systems module, algorithms and data structures, computer architecture, design and programming of simple software solutions to elementary problems, programming in the C language. Ability to integrate classroom study room with personal study. Reading comprehension of English.
4	**Teaching methodsand language**	Lectures and exercises **Language**: Italian **Reference textbooks** • W. Richard Stevens, Stephen A. Rago, ***Advanced Programmingin the UNIX Environment***. Addison-Wesley Professional Computing Series. (vol. 3rd Edition) 2013.

5	**Assessment methods**	For the 1st session, the exam consists of (1) a midterm written exam + a final written exam or (2) a total written exam. The exam is passed if the score of the total written exam or the final score as the average of the midterm written exam + the final written exam is greater or equal to 18. For later sessions, the exam consists of only a total written exam.

Programme of Module "Sistemi Operativi"

- Code: F1I020
- Type of course unit: Compulsory (Bachelor Degree in Computer Science curriculum General)
- Level of course unit: Undergraduate Degrees
- Semester: 1

Number of ects credits: (Bachelor Degree in Computer Science) 6 (workload 150 hours)

Teachers: Vittorio Cortellessa (Vittorio.Cortellessa@univaq.it)

1	**Course objectives**	KNOWLEDGE: basic concepts common to all the operating systems, mechanisms and policies of operating systems, system overhead vs solution efficiency tradeoff SKILLS : ability to relate different topics, ability to solve problems never faced in classroom, but solvable through logic deductions and reasoning (i.e., ability to analyze and synthesize concepts), ability to work during the course time and do not delaying the refinement of the knowledge, improved ability to pose questions in the classroome to originate discussion EXPECTED BEHAVIORS : interest for an integrated knowledge of different aspects of computer science, awareness of relationships among computer subsystems, hence awareness of the fact that a satisfactory behavior of a computer may derive from the combination of very different (sometimes unexpected) factors.
2	**Course content and learning outcomes (dublin descriptors)**	Topics of the module include: • General concepts, computer system and operating system structures • Processes and CPU scheduling • Process synchronization and deadlock management • Memory management • The virtual memory • The file system On successful completion of this module, the student should : • Students shall know the basic concepts that are common to all general-purpose operating systems, in particular the ones related to the management of CPU and central memory. Students shall also be able to relate these concepts in order to synthesize the intrinsic tradeoffs that underlie a virtual machine (i.e. a computer plus its operating system). • Students shall definitely be able to solve complex problems related to the synchronization among concurrent processes. Beside, they shall also be able to apply the operating system policies studied in the course (such as CPU schedulers, pagers, etc.) to specific examples. • Students shall be able to select the best solutions (among those studied in the course) for specific examples. • Students shall be able to critically explain why the existing operating systems operate in their ways, basing also on the historical notions that they have received in the course and that help to understand the current state-of-art. • On the basis of the knowledge and capacities acquired in this course, the students shall be able in the future to tackle any actual operating system, by just studying its handbook. This is expected because their knowledge is independent on any specific existing system in this course module, with the goal of providing general instruments suitable for a continuous learning in this domain.
3	**Course prerequisites**	KNOWLEDGE : fundamentals of programming, algorithms and data structures, computer architecture, reading and understanding english language SKILLS : ability to integrate

		classroom and homework study, ability to pose questions in the classroome to originate discussion.
4	**Teaching methodsand language**	The module includes 54 hours of frontal lectures plus 30 hours of on-demand clarifications in the teacher's office. The frontal lectures are partitioned in theory and exercises. **Language**: Italian **Reference textbooks** • A. Silberschatz, P.B. Galvin, G. Gagne, ***Operating System Concepts***. John Wiley & Sons.
5	**Assessment methods**	Pre-Assessment There is no formal pre-assessment, but Course pre-requisites are clearly stated on the Module website. Fulfilment of such pre-requisites is verified by formative assessment. Formative Assessment The formative assessment is performed via interactive interaction beteween teacher and students during lectures. Students are aware since the beginning of the Course that they will be involved (in turns) in: - Questioning and discussion, by means of open oral questions to the class or to single students. Summative Assessment Written test followed by an optional oral exam. An optional mid-term written test is also be provided, which is meant to cover the first part of the course, in order to help the students to split the workload. The written test is aimed at: (1) verification of theoretical competences, and in particular of knowledge and comprehension of Course contents; (2) verification of skills in understanding and solving significant exercises, and in explaining the proposed solutions. This in order to verify the ability of application of techniques learnt during the Course, of analysis of problems and synthesis of suitable solutions, and of evaluation of alternative solutions. Criteria of evaluation will be: the level of knowledge and practical ability; the property of use of the technical/mathematical language; the clarity and completeness of explanations. The oral exam will occur within one week of the written test and will typically cover the areas of the written answers that need clarification plus additional subjects proposed by the teacher. The oral test can be required: (i) by the student, to improve final marks; (ii) by the teacher, in presence of significant mistakes/misunderstandings in the written test. Assessment breakdown: 100% mid-term plus end-of-semester summative assessment. The written test (2 hours) consists, in general, in: (a) Three exercises to solve; (b) A list of about 3-4 questions to answer. The oral test (max 1 hour) consists of questions on the written exam and extra ones. The final marks of the Operating Systems with Laboratory 12 CFU Module are obtained as the average among the marks of the Operating Systems and Operating Systems Laboratory 6 CFU Modules.