

Academic Year: (2021 / 2022)

Review date: 02-11-2021

Department assigned to the subject: Telematic Engineering Department

Coordinating teacher: PUENTE RODRIGUEZ, LUIS ANTONIO

Type: Compulsory ECTS Credits : 6.0

Year : 3 Semester : 2

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

Systems programming
Systems Architecture I

OBJECTIVES

The goal of the course is to enable students to acquire knowledge relative to the fundamental concepts of software systems at the level of the operating system and fundamental concepts on distributed systems.
In order to achieve this goal, the student must acquire the following knowledge:

- To know the basic structure of an operating system.
- To know the software structure based on processes.
- To know the mechanisms of communication between processes.
- To know the software structure based on threads and the characteristics of multi-threaded (or multi-threaded) applications.
- To know the concurrency management mechanisms.
- To know memory management mechanisms.
- To know the architecture of a distributed system.
- To know the basic mechanisms of communication in distributed systems.
- To know the typical architecture of a communications middleware and its fundamental components.

The specific capacities that will be acquired by the student upon successful completion of the course will be:

- To use the basic mechanisms that an operating system offers to support application execution.
- To use the concurrency management mechanisms and memory management mechanisms.
- To design and implement multi-threaded applications both centralised and distributed and their basic synchronization mechanisms.
- To use communication models in distributed systems based on message passing.
- To model and deploy a distributed system.

DESCRIPTION OF CONTENTS: PROGRAMME

1. Architecture of a software system: centralized and distributed
 - Fundamental concepts of a distributed system.
 - Software and hardware aspects

2. The operating system

- 2.1 Functionality and basic services
 - 2.2 Multiprogramming: processes.
 - 2.3 Inter Process communication mechanisms
 - 2.4 Threads and synchronization
 - 2.5 Memory management.

3. Distributed systems models, algorithms and software

- 3.1 Distributed system models
 - 3.2 Algorithms for distributed synchronization, election, and transactions

- 3.3 Consensus algorithms
- 3.4 Middleware models and distributed programming

LEARNING ACTIVITIES AND METHODOLOGY

The activities will be the following:

- Lectures: study and explanation of the main concepts of the subject through the use of oral presentations by the lecturer and discussions of the main problems and the solutions proposed in the literature.
- Exercise classes: sessions where real problems will be presented to the students. The students will analyse these problems and they will propose suitable solutions.
- Labs: concrete small scale problems will be presented to the students. They have to propose solutions based on the concepts explained in the lectures and exercise classes and, with the supervision of the lab teacher, they have to implement their solutions.

ASSESSMENT SYSTEM

** Evaluation: Individual (50%), In pairs (50%).

- The individual evaluation will consist of written partial exams during the term.
- The evaluation of the collaborative work carried out in pairs will be evaluated through assignments.

Written exams:

- * Partial exam about processes (15%).
- * Partial exam about synchronization mechanisms (20%)
- * Partial examination about memory management and distributed systems (15%)

Assignments (consisting of one or more exercises):

- * Lab about processes (15%).
- * Lab about synchronization mechanisms (15%)
- * Lab about distributed systems (20%)

% end-of-term-examination:	0
% of continuous assessment (assignments, laboratory, practicals...):	100

BASIC BIBLIOGRAPHY

- A. Silberschatz, P. Galvin Operating System Concepts., Wiley. 7ª edición..
- B. Kernighan, D. Ritchie The C Programming Language, Prentice Hall, 1998
- Bradford Nichols, Dick Buttlar, Jacqueline Proulx Farrell Pthreads Programming, O'Reilley, 1996
- G. F. Coulouris, J. Dollimore, T. Kindberg. Distributed Systems. Concepts and Design, Addison Wesley. 4th ed., June 2012. ISBN 0201-61918-0..
- M. Kerrisk The Linux Programming Interface, No Starch Press, 2013
- R. Love. Linux System Programming, O'Reilley., 2013.

ADDITIONAL BIBLIOGRAPHY

- A.S. Tanenbaum, M. Van Steel Distributed Systems., Prentice-Hall..
- Brian W. Kernighan The UNIX programming environment, Prentice Hall.
- Brian W. Kernighan, Dennis M. Ritchie The C programming language, Prentice Hall, 1989.
- David Butenhof Programming with POSIX threads, Addison Wesley.
- Gregorio Fernández Conceptos básicos de arquitectura y sistemas operativos: Curso de Ordenadores, Madrid, Sistemas y Servicios de Comunicación..
- S. Oaks, H. Wong Java Threads., O`Reilly.