

Academic Year: ( 2019 / 2020 )

Review date: 29-04-2019

Department assigned to the subject: Department of Telematic Engineering

Coordinating teacher: ESTEVEZ AYRES, IRIA MANUELA

Type: Compulsory ECTS Credits : 6.0

Year : 3 Semester : 2

**STUDENTS ARE EXPECTED TO HAVE COMPLETED**

Systems programming  
Systems Architecture I

**COMPETENCES AND SKILLS THAT WILL BE ACQUIRED AND LEARNING RESULTS.**

The goal of the course is to enable the student to acquire knowledge relative to the fundamental concepts software systems at the level of the operating system and fundamental concepts on distributed systems. 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 execution threads, the characteristics of multi-threaded applications, and the mechanisms to manage concurrency problems.
- To know the architecture of a distributed system.
- To know the basic communication mechanisms of distributed systems.
- To know the mainstream 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 deploying a distributed system.

The general capabilities acquired by the students will be:

- Ability to use and apply knowledge of telecommunications technologies and engineering. The students will work on this capability in the lab sessions and in the exercise sessions. (PO a)
- Ability to use the techniques and modern engineering tools necessary for the professional practice (PO k)
- Recognition of the need of a life-long learning and the ability to obtain and apply the required and suitable information through the use of technical literature related to the subject field in spanish and in english (PO i)

**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 (PO a and k).
- Exercise classes: sessions where real problems will be presented to the students. The students will analyse these problems and they will propose suitable solutions. (PO a and k)
- 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. (PO a and k)

### ASSESSMENT SYSTEM

\*\* Evaluation: Theory (45%), Lab (55%).

The theory will be assessed with a partial exam during the semester and the final exam.  
The lab will be evaluated with lab submissions during the semester and laboratory tests.

Theory:

- \* Partial exam about processes (10%).
- \* Final exam (35%). This exam has no minimum grade.

Lab:

- \* Lab submissions:
  - Lab about processes (10%)
  - Lab about threads, concurrency and synchronization (15%)
  - Lab about distributed systems (15%)
- \* Laboratory test about threads, concurrency and synchronization (15%)

<b>% end-of-term-examination:</b>	35
<b>% of continuous assessment (assignments, laboratory, practicals...):</b>	65

### 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.