uc3m | Universidad Carlos III de Madrid

Distributed Systems

Academic Year: (2022 / 2023) **Review date: 25-05-2022**

Department assigned to the subject: Computer Science and Engineering Department

Coordinating teacher: GARCIA CARBALLEIRA, FELIX

Type: Electives ECTS Credits : 6.0

Year : 3 Semester : 2

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

Operating Systems (Course 2 - Semester 2)
Computer networks (Course 3 - Semester 1)

SKILLS AND LEARNING OUTCOMES

¿ Understand the basic concepts and paradigms of distributed computing, with a focus on standards-based message exchange techniques
¿ Understand the principle of usability of remote or web-based services,
¿ Design and implement distributed storage services.

OBJECTIVES

The main objective of the course is to describe the main concepts needed for designing and developing distributed systems and applications.

DESCRIPTION OF CONTENTS: PROGRAMME

The basic concepts of this course are: concurrency; interprocess communication; middleware; RPC; distributed file systems; distributed applications; fault tolerance; web services;

Content:

1. Introduction
   - Basic concepts
   - Interconnection networks
   - Advantages of distributed systems
   - Distributed computing paradigms
   - Design of distributed systems

2. Process communication and synchronization
   - Communication mechanisms in shared memory systems
   - Communication mechanisms in distributed memory systems
   - POSIX services
   - Threads in Python

3. Message passing and client-server applications
   - Communication model using message passing
   - Design aspects
   - POSIX queue messages
   - Client-server applications
   - Design of concurrent servers

4. Communication using sockets
   - Communication model with sockets
   - POSIX sockets API
   - Python sockets API
   - Design guide of client-servers applications using sockets

5. Remote procedure call
   - RPC behavior
   - Interface definition language
   - Marshaling and message transfer

- ONC-RPC
- RPCs in Python

6. Web services
   - HTTP protocol
   - SOAP
   - Development of web services

7. Distributed services
   - Name services
   - Synchronization in distributed systems
   - Physical and logical clocks
   - Distributed mutual exclusion
   - Algorithms of election
   - Multicast

8. Distributed storage systems
   - Distributed file systems structure
   - File and directory services
   - Implementation of distributed file systems
   - Example: NFS
   - Shared disks file systems
   - Parallel file systems
   - Storage area networks

9. Fault tolerant in distributed systems
   - Fault tolerant concepts
   - Software fault tolerance
   - Fault detectors
   - Replication
   - Protocols of consensus

LEARNING ACTIVITIES AND METHODOLOGY

Learning outcomes:

R1 Knowledge and understanding: have basic knowledge and understanding of the scientific and technological foundations of Computer Engineering, as well as a specific knowledge of computer science, computer engineering and information systems.
R2 Engineering Analysis: Be able to identify Computer Engineering problems, recognize their specifications, establish different resolution methods and select the most appropriate one for their solution, taking into account the social, human health, environmental, and commercial constraints applicable in each case.
R3 Engineering Design: Be able to perform engineering designs according to their level of knowledge and understanding that meet the required specifications in collaboration with other engineers and graduates. Design encompasses devices, processes, methods and objects, and specifications broader than strictly technical, including social awareness, health and safety, environmental and commercial considerations.
R5 Engineering Applications: Graduates will be able to apply their knowledge and understanding to solve problems, conduct research, and design devices or processes in the field of Computer Engineering according to criteria of cost, quality, safety, efficiency, environmental friendliness, and ethical implications. These skills include the knowledge, use and limitations of computer systems, process engineering, computer architectures, and the use of computer systems in the field of computer engineering. They aim to achieve the specific cognitive competences of the subject, as well as the transversal competences capacity of analysis and abstraction.
* Practical classes: 1 ECTS. They aim to initiate the development of the specific instrumental competences, as well as the transversal competences problem solving and application of knowledge.
* Continuous evaluation exercises: 1,5 ECTS. Initiated during the practical classes and completed outside of them, they aim to complete the development of the specific instrumental competences and to initiate the development of the specific attitudinal competences, as well as the transversal competences problem solving and application of knowledge.
* Practical work: 2 ECTS. Developed without the presence of the teacher, they aim to complete and integrate the development of all the specific and transversal competences, in the resolution of two practical cases where the approach to the problem, the choice of the method of resolution, the results obtained and their interpretation are well documented.
* Tutorials: TUTORIALS. Individualized assistance (individual tutorials) or in group (collective tutorials) to the students by the professor.

* Final exam: 0.5 ECTS. It aims to influence and complement the development of specific cognitive and procedural skills. It reflects especially the use of the master classes.

ASSESSMENT SYSTEM

The evaluation includes the following parts:

1. The continuous assessment (80 %) includes:
 - Exercises and programming projects: 40%.
 - Laboratory projects: 40%.

2. The percentage of the final exam is: 20%. The final exam will include theoretical and practical concepts.
The minimum value for the final exam will be 4.
The minimum value for all lab projects will be 4.

A student follows the continuous assessment when the student:

- makes all exercises and laboratory projects, the minimum value for each lab will be 2.
- the mínimum values for all lab projects is 4.

The final exam in the extraordinary period will include the theoretical and practical concepts of the course.

% end-of-term-examination:	20
% of continuous assessment (assigments, laboratory, practicals…):	80

BASIC BIBLIOGRAPHY

- G. Coulouris, J. Dollimore, T. Kindberg, G. Blair Distributed Systems, Concepts and design. 5ª edition. 2011, Addison-Wesley.

ADDITIONAL BIBLIOGRAPHY

- Distributed Systems: principles and paradigms Andrew S. Tanenbaum , Maarten van Steen, Pearson.
- F. García, J. Carretero, A. Calderón, J. Fernández, J. M. Pérez Problemas resueltos de programación en C, Thomson.
- L. H. Etzkorn Introduction to Middleware: Web Services, Object Components, and Cloud Computing, CRC Press, 2017
- Pankaj Jalote Fault Tolerance in Distributed Systems, Prentice-Hall.
- Richard Stevens UNIX Network Programming, Prentice Hall.