Distributed Systems

Academic Year: (2019 / 2020) **Review date: 29-04-2020**

Department assigned to the subject: Computer Science and Engineering Department

Coordinating teacher: GARCIA CARBALLEIRA, FELIX

Type: Compulsory ECTS Credits : 6.0

Year : 5 Semester : 2

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

Operating Systems
Computer networks

OBJECTIVES

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

1. Generic competences:
- Capacity of analysis and synthesis (PO a, b, c, e)
- Capacity of organization and scheduling (PO c, d)
- Problem solving (PO a, b, c, g)
- Capacity to apply theoretical concepts (PO a, c)

2. Specific competences:
a. Cognitive (PO a,c e,j)
- Concepts of a distributed system
- Main hardware platforms used in distributed systems
- Understanding the concept of middleware and the services provided by a middleware
- Techniques for developing distributed applications
- Storage systems used in distributed systems
- Techniques for developing fault tolerant distributed systems
- Knowledge and application of features, functionality and structure of Distributed Systems, Computer Networks and Internet-based design and implement distributed applications (CECRI11)
- Ability to design and implement software and communication systems (CEIC4)

b. Procedimental/Instrumental (PO a, b, c, g, j, k)
- Designing and developing a distributed system and a distributed application.
- Designing and developing distributed applications using the main TCP/IP services, RPC or services provided by middlewares.
- Using tools (programming languages and operating systems) appropriates for developing distributed systems.
- Designing and developing distributed applications using middleware services and remote procedure calls.
- Evaluating the performance of a distributed application
- Researching and looking for solutions to new problems related to distributed systems.

c. Attitude (PO: c, e)
- Creativity
- Critical vision of current distributed systems
- Motivation
- Interest for acquiring new knowledge and information

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 Java

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
    - Java 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
    -Remote method invocation in JAVA (RMI)

6. Web services
    - HTTP protocol
    - SOAP
    - Development of web services with JAX-WS
    - Development of web services with gSOAP

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

The teaching methodology includes:

1. Theoretical lectures 1 ECTS (PO a, b, c, e, j)
   - Distributed systems concepts
   - Basic text books will be also recommended
2. Projects 1.5 ECTS (PO a, b, c, d, e, g, k)
   - Several projects are made along the course applying the concepts shown in theoretical lectures. Partial teacher support in computer labs.
   - Students have to analyze the requirements and provide a working solution
   - The projects are developed in team of 2 students, in order to promote the collaborative work
3. Problem solving with the teacher 1 ECTS (PO a, c, e)
   - By solving exercises and case of studies in a participatory way.
4. Student work 2 ECTS. (PO a, c, e, g, k)
   - Self-study to understand the theoretical concepts
   - Homework for solving proposed exercises
   - Students must to understand and present in public the main aspects obtained in the analysis of a basic paper written in English.
5. Exams 0.5 ECTS (PO a, c, e, g)
   - Midterm and final exam

ASSESSMENT SYSTEM

The evaluation includes the following parts:

1. The continuous assessment (75 %) includes:
   - Programming and laboratory projects: 60%. (PO b, c, d, e, g, k)(CECRI11, CEI4)
   - Analyze and public presentation of a simple and basic paper written in English: 15% (b, g, j)

A student follows the continuous assessment when the student obtains more than 4 in all lab projects and presents the article in public.

2. The percentage of the final exam is: 25%. (PO a, c, e, g, j). 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 labs, the minimum value for each lab will be 2.
- the mínimum values for all lab projects is 4.
- makes the article presentation.

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

The final grade will be increased by 1 point to those students who complete all parts of the continuous assessment, obtain more than 7 in the continuous evaluation, and at least 4 in the final exam.

% end-of-term-examination:	25
% of continuous assessment (assigments, laboratory, practicals…):	75

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.
- Pankaj Jalote Fault Tolerance in Distributed Systems, Prentice-Hall.
- Richard Stevens UNIX Network Programming, Prentice Hall.