Operating Systems

Academic Year: (2019 / 2020) **Review date: 21/04/2020 17:02:22**

Department assigned to the subject: Computer Science and Engineering Department

Coordinating teacher: CARRETERO PEREZ, JESUS

Type: Compulsory ECTS Credits : 6.0

Year : 2 Semester : 2

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

Computer Programming
Computer Structure

OBJECTIVES

The aim of this course is that the student knows the role of extended machine operating system, the services offered by the rest of the system and its main components and entities (processes, memory, files, etc.), the concepts of competition and relations with the rest operating system software and computer hardware. To achieve this goal the student must acquire a number of generic skills, knowledge, skills and attitudes.

1 Transversal / Generic Skills
1.1 Capacity for analysis and synthesis
1.2 Ability to organize and plan
1.3 Troubleshooting
1.4 Teamwork
1.5 Ability to apply knowledge in practice
2 Specific Skills
2.1 Cognitive (Know)
- Knowledge of the concepts of SS.OO
- Knowledge of the organization and operating system services as extended machine
- Knowledge of the concepts of concurrency
- Understanding the effect of the operating system on the rest of the system.
- Techniques for programming calls to operating systems
- Knowledge of methods of resource management in an operating system
2.2 Procedural / Instrumental (Know how)
- Programming with calls to operating systems
- Programming concurrent applications
- Design of the operating system utilities
- Use monitoring tools operating systems
2.3 Attitudinal (Be)
- Ability to generate new ideas (creativity)
- Critical attitude regarding current operating systems
- Concern for quality operating systems
- Achievement motivation
- Interest in research and seek solutions to new problems related to operating systems.

DESCRIPTION OF CONTENTS: PROGRAMME

T1.- Introduction to Operating Systems
  1.1.- Basics.
  1.2.- Main features: extended machine, resource manager and user interface
  1.3.- History of operating systems
  1.4.- Structure and operating system components.

1.5.- Operating System Activation

T2 services operating systems.

2.1.- Operating system services. System call.
2.2.- Services associated with the various components of the operating system.
2.3.- System call interface for systems programming.
2.4.- Generation and implementation of programs
2.5.- Static and dynamic libraries

T3.- processes and threads

3.1.- Process Definition.
3.2.- Resources, multiprogramming, multitasking and multiprocessing
3.3.- Lifecycle process: state of processes.
3.4 - Services to manage processes.
3.5.- Definition of thread.
3.6.- Threads: library and kernel.
3.7.- Services for operating system threads.
3.8.- Data structures for processes and threads in the kernel
3.9.- Design and implementation of multiprogramming and multitasking in the kernel

T4.- Scheduling Processes and threads.

4.1.- Scheduling basics.
4.2.- Scheduling and activation
4.3.- Scheduling algorithms (FIFO, SJF, RR, priority, ...).
4.4.- LINUX scheduling: aging.
4.5.- Process scheduling calls.
4.6.- Scheduler data structures in the kernel

T5 Communication between processes

5.1.- Signals and exceptions.
5.2.- Timers.
5.3.- Process communication with pipes .
5.4.- Local message passing.

T6 concurrent processes and synchronization

6.1.- concurrent processes.
6.2.- Mutual exclusion and critical section.
6.3.- Semaphore
6.4.- System Calls for traffic lights.
6.5.- Thread synchronization mechanisms.
6.6.- Mutex and condition variables.
6.7.- System calls to mutex.
6.8.- Cassic concurrency problems.
6.9.- Case study: development of concurrent servers

T7 Files and Directories

7.1.- Understand the concepts of file and directory and its characteristics.
7.2.- study the files, their attributes and operations, logical view.
7.3.- Representation from the point of view of users.
7.4.- Services for files.
7.5.- Interpretation of names.
7.6.- Services for directories.
7.7.- volumes, partitions and filesystems.

T8 Security and Protection

8.1.- security mechanisms in operating systems.
8.2.- Security in Linux
8.3.- Security in Windows

LEARNING ACTIVITIES AND METHODOLOGY

- Theoretical classes: 1.5 ECTS. They aim to achieve specific cognitive skills of the subject. In them the knowledge that students should acquire will be presented. To facilitate their development students receive class notes and have basic reference texts that allow them to complete and deepen those

subjects in which they are most interested.
- Practical classes: 1.5 ECTS. Instrumental develop specific skills and most of the cross, such as teamwork, ability to apply knowledge to practice, to plan and organize and analysis and synthesis. Also they aim to develop the attitudinal specific capabilities. Is the design and development of a shell with concurrency with several new commands to the operating system. They developed in working groups and in computer classes with teacher presence.
- Guided academic activities with presence of the teacher: 1 ECTS
- Resolution of exercises and case studies of participatory in class. It may further include the study of examples of operational systems common answers to the exercises and joint correction, presentation of papers, etc.
- Student study: 1.5 ECTS. Exercises and readings proposed by the teacher. Personal study
- Exercises and exam: 0.5 ECTS. They intended to influence and complement in the development of cognitive and procedural specific capabilities.

ASSESSMENT SYSTEM

% end-of-term-examination/test:	50
% of continuous assessment (assigments, laboratory, practicals…):	50

The evaluation allows to know the degree of satisfaction of the knowledge goal, thus all work of the students will be valuated by using continuous evaluation of their activities by using exercises, exams, projects, and other activities.

The following scoring will be used for continuous evaluation:

a) Ordinary Exam: 35%. (PO: a, b, e, f, g, h, i, k)
* Activities to assess theory concepts and OS problem solving abilities.
* It covers all the program.

b) Ordinary Exam: 20%. (PO: a, b, e, f, g, h, i, k)
* Activities to assess theory concepts and OS problem solving abilities.
* It covers 50% of the program.

c) Projects and exercises: 45%. (PO: a, b, c, d, e, g, k)
* Activities must be delivered on time. They are mandatory.
* Each project is evaluated separately, including solution adopted, functionality completeness, and design.
* Evaluation of the project written memory. Project memory organization and correctness, written exam correctness.
* Evaluation of tools usage.
* Evaluation of the collaborative work of the members distinguishing roles. Responsibility of the result is shared by all members.
* Total score for project is computed by given weight to each activity.

To pass the projects, it is mandatory to deliver of all them, to get a minim score of 2 per project, and a minimum average score of 4 fort all the projects . If those criteria are not covered, the student will loose continuous evaluation.

- Voluntary activities: 10%. (PO: a, b, c, d, e, g, k)
* Activities must be delivered on time. They are not mandatory.
* Extra projects or exercises requested in class.

In the case of copy detection in any project or partial exam, those students implicated will loose continuous evaluation. Copy could be among students or by taking the projects from Internet.

You will also loose continuous evaluation, if you not deliver all the projects, or do not get minimum a score of 2 in every project.

For those students not following the continuous evaluation, the ordinary exam will cover all the program (including projects). It will have a maximum value of 60% over 10.

A minimum score of 35% is required to follow the continuous evaluation.
If the student does not get the minimum, but the average of continuous evaluation and the exam is higher than 50%, the final student sore will be 45%.

To compute the final score for the extraordinary exam, the following situations are possible:
A.- Students following continuous evaluation that did not pass:

% end-of-term-examination/test:	50
% of continuous assessment (assigments, laboratory, practicals…):	50

a- Extraordinary exam will weight 35%
b- Other 65% will come from the score of continuous evaluation.
c- A minimum score of 40% is mandatory to pass the exam and compute the average.

B- Students not following continuous evaluation partially or totally:
a.- Extraordinary exam will weight 100%
b.- It may include all the topics related to the course contents, including theory and projects.
c.- A minimum of 5 is required to pass the exam.

BASIC BIBLIOGRAPHY

- Jesus Carretero y otros Sistemas Operativos: una visi´n aplicada. Segunda edición, McGraw-Hill España, 2007

- Jesus Carretero y otros Problemas de Sistemas Operativos. Segunda edición, Amazon Books Press, 2015