Operating Systems

Academic Year: (2020 / 2021) **Review date: 31/01/2021 19:14:28**

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)

Programming
Computer Structure

OBJECTIVES

The goal of the course is to allow the student knowing the functioning of the operating system as a expanded machine, its services for the system and its components, the major entities (processes, memory, files, etc.), concurrency, and the relations of the operating systems with the sw and hw of the computer.

To achieve this goal, the student must acquire the following program outcomes (PO): a, b, c, d, e, f, g, h, i, k.

Related to the following competences:

1.- General competences
- Analysis and synthesis (PO: b, e, g)
- Planning and organization (PO: b, c, h)
- Problem solving (PO: a, e, k)
- Collaborative work (PO: d)
- Capacity to apply theoretical concepts (PO: a, b, c, e, f, i, k)

2.- Specific competences (CECRI5, CECRI10).
2.1.- cognitive (PO: a, b, c, e, f, h, k)
- Operating system concepts
- Knowing the main features, functionality, and structure of the Operating Systems.
- Concurrency concepts
- Operating system programming and design of applications based on OS services.
- Resource management in operating systems

2.2.- Instrumental (PO: b, c, e, k)
- Programming with operating system calls
- Programming concurrent applications
- Desinning utilities on the operating system
- Using tools to monitorize and management of the operating systems

2.3 Attitude (PO: e, g, h, i)
- Creativity
- Critical vision of the operating systems
- Quality aspects and operating systems
- Motivation
- Interest for finding new solutions with operating systems

DESCRIPTION OF CONTENTS: PROGRAMME

Program:

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

P3.- 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 Memory Management
- 7.1 Memory partitioning
- 7.2 Memory management algorithms
- 7.3 Virtual memory

T8 Files and Directories
- 8.1.- study the files, their attributes and operations, logical view.
- 8.2.- Services for files.
- 8.3.- Interpretation of names.
- 8.4.- Services for directories.
- 8.5.- volumes, partitions and filesystems.

T9 Security and Protection
9.1.- security mechanisms in operating systems.
9.2.- Security in Linux
9.3.- Security in Windows

LEARNING ACTIVITIES AND METHODOLOGY

- Theory: lectures & exercises 1.5 ECTS. (PO: a, b, e, f, g, h, i, k)

* Operating system theoretical concepts related to the program, professional aspects, importance of the subject, critical vision of the operating systems and quality aspects, information acquisition and lifelong learning recognition.
* OS problems formulation and resolution. Analysis and synthesis. Applying technical knowledge to solve operating systems problems.
* Examples during the lectures to show the students the professional and legal responsibilities due to system failures because of OS, and their economic repercussions.
* Examples in the lectures to show the students the impact of choosing an OS solution in the enterprise economic context.
* Communication skills are enhanced through reading of materials and written exams.

- Projects. 1.5 ECTS. (PO: a, b, c, d, e, g, k)
* Several projects are made along the course applying computer systems principles to the field of computer engineering. Partial teacher support.
* Projects are developed through a design problem under initial specifications, where the students have to analyze requirements and provide a working solution.
* Students are required to use OS tools and provide solutions to real-world problems. Use of professional tools for Linux and Windows OS for solving OS projects.
* They develop collaborative work, capacity to apply theoretical concepts, and capacity to make an experiment in time meeting desired needs.
* Communication skills are enhanced through writing of memory projects in English and Spanish.

- Academic activities with the teacher. Lab experiments. 1 ECTS. (PO: b, d, e, g, k)
* Students must design and execute lab experiments with teacher support, such as OS monitoring, system installation, etc.
* To extract conclusions, they must also analyze, and interpret data.
* Students are required to use OS tools and provide solutions to real-world problems. Use of professional tools for Linux and Windows OS for solving OS projects.
* They develop collaborative work, capacity to apply theoretical concepts, and capacity to make an experiment in time meeting desired needs.

- Student work. 1.5 ECTS. (PO: a, b, c, d, e, f, g, h, i, k)
* Self-study to understand the theoretical concepts and how to apply them to solve operating systems problems.
* Homework, individually or cooperatively, to finish the OS projects requested along the course.
* Information acquisition and study to know new solutions with operating systems and to recognize the importance of OS in the Computer Engineering field, and the changing world of OS and the need of lifelong learning to be an active professional.

- Exams. 0.5 ECTS. (PO: a, b, c, d, e, f, g, h, i, k)

* Assessment described below

ASSESSMENT SYSTEM

% end-of-term-examination/test:	35
% of continuous assessment (assigments, laboratory, practicals…):	65

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:

% end-of-term-examination/test:	35
% of continuous assessment (assigments, laboratory, practicals…):	65

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) Partial activities: 20%. (PO: a, b, e, f, g, h, i, k)
* Partial assessments of theory concepts and OS problem solving abilities. It covers 50% of the program.
* Extra projects or exercises requested in class.
* Other activities requested along the course. Must be delivered on time.

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.

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 pass the ordinary evaluation, the student must pass 50% considering the scores of the trajectory chosen.

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

- Abraham Silberschatz Operating System Concepts, 10th edition, Wiley & Sons, 2018

- J. Carretero, F. Garcia, F. Pérez. Problemas de Sistemas Operativos: de la base al diseño. 2ª Edición, Amazon, 2015

ADDITIONAL BIBLIOGRAPHY

- A.Silberschatz, P.B. Galvin, G. Gagner Operating Systems Concepts, Nineth Edition, John Wiley & Sons, Inc..

- F. García, J. Carretero, A. Calderón, J. Fernández, J. M. Pérez. Problemas resueltos de programación en C, Thomson, 2003. ISBN: 84-9732-102-2..