COURSE DATA

Data Subject	
Code	34838
Name	Operating systems
Cycle	Grade
ECTS Credits	6.0
Academic year	2022 - 2023

Study (s)

Degree	Center	Acad. year	Period
1407 - Degree in Multimedia Engineering	School of Engineering	3	First term

Subject-matter

Degree	Subject-matter	Character
1407 - Degree in Multimedia Engineering	4 - Ingeniería de Computadores	Obligatory

Coordination

Name	Department
MARTINEZ DURA, RAFAEL JAVIER	240 - Computer Science

SUMMARY

The course "Operating Systems" is a compulsory 6 ECTS, taught in the first quarter of third degree course in Multimedia Engineering. It is part of the matter "Computer Engineering".

The course covers the operating systems from three complementary viewpoints:

- The operating system as an interface for developing and running applications. From this point of view we consider the basic abstractions provided by the operating system (processes, memory, files and input / output) and the services related to them.
- The operating system as a control system that manages the use of computer resources and relies on the hardware (hardware) to ensure the proper functioning of the system.
- The OS as a program. Therefore it also takes into account aspects such as its internal structure, and the data structures and algorithms used to perform their functions.

Overall Objectives

- Show what an operating system is and what services it offers, providing an overview of the functioning of today's computers and, specifically, the roles played by the operating system.
- Show basic abstractions provided by the operating system and what operations can be done with them, emphasizing the role of the operating system as a platform for developing and running applications.
- Show the correspondence between these basic abstractions and the physical components of a computer, illustrating how the operating system requires hardware support to provide these abstractions. And how operating systems manage the physical resources available, with special emphasis on the efficiency and cost of the different solutions.
- Show the differences between the conventional operating systems and those related with the execution of multimedia applications
- Knowledge of the main peripherals use in multimedia applications
- Enable the student as a user and as a programmer in the multimedia operating system environment.

Contents

- Introduction
- Processes and threads
- Processor scheduling
- Memory Management
- Process communication and synchronization
- File systems
- Input/output management and multimedia peripherals

PREVIOUS KNOWLEDGE

Relationship to other subjects of the same degree

There are no specified enrollment restrictions with other subjects of the curriculum.

Other requirements

It is recommended to have completed the following courses: Computer Science, Computer Structure and Programming.

COMPETENCES (RD 1393/2007) // LEARNING OUTCOMES (RD 822/2021)

1405 - Degree in Multimedia Engineering

- G3 - Take into account the economic and social context in engineering solutions, be aware of diversity and multiculturalism and ensure sustainability and respect for human rights and equality between men and women.
- G5 - Be able to lead working groups properly, respect and appreciate the work of others, take into account the needs of the group and be available and accessible.
- B4 - Have basic skills in the use and programming of computers, operating systems, databases and computer software for use in engineering.
- I2 - Know, design and make an efficient use of the data types and data structures that are most suited to solving a problem.
- I3 - Be able to analyse, design, build and maintain applications in a robust, secure and efficient manner by choosing the most suitable paradigm and programming languages.
- I4 - Be able to identify, understand and evaluate the structure and architecture of computers, as well as the basic components that comprise them.
- I5 - Know the features, functionalities and structure of operating systems and be able to implement applications based on their services.
- MM1 - Have knowledge and ability to understand essential facts, concepts, principles and theories related to multimedia systems including all the disciplines covered by these systems.
- MM2 - Be able to understand and manage the different technologies involved in multimedia systems, both from the point of view of hardware and electronics and of software.
- MM3 - Be able to implement methodologies, technologies, processes and tools for the professional development of multimedia products in a real context of use by applying the appropriate solutions for each environment.
- MM5 - Know how to apply the theoretical and practical resources to deal with a multimedia application as a whole.
- MM22 - Have knowledge and ability to understand essential facts, concepts, principles and theories related to multimedia and to the spectrum of reference disciplines.

LEARNING OUTCOMES (RD 1393/2007) // NO CONTENT (RD 822/2021)

This course allows for the following learning outcomes:

1. Describe what an operating system (OS) is and what its role is, being able to compare major operating systems.
2. Explain what are processes and threads and how they are managed by the operating system, and write simple programs which use the services to manage processes and threads.
3. Explain the advantages and disadvantages of several scheduling algorithms and evaluate their suitability based on certain objectives.

4. Explain the advantages and disadvantages of the different mechanisms of memory management including virtual memory.
5. Describe the different communication and synchronization mechanisms and select which one to use in a particular case, being able to design and implement concurrent algorithms that use them.
6. Explain the differences among different I/O devices based on how they are managed by the operating system and what is the structure of the I/O.
7. Explain the basic abstractions provided by file systems and the services related to them, and compare different file systems.

To complement the above results, this subject also allows to acquire the following dexterities and social skills:

- Understanding what an operating system (OS) is, being able to compare among major operating systems.
- Using operating system services for sequential and concurrent application development.
- Understanding the relationship between OS services and hardware, as well as the relationship among the abstractions, getting an overall knowledge of how an operating system works.
- Compare and select the most appropriate algorithms for the management of processes and threads, memory, I/O and file systems.
- Solve problems that span different concepts of the subject.
- Analyze the reasons for low performance or malfunctioning of operating systems.
- Knowing the multimedia application requirements and the main methods the operating system employs to guarantee them.

Social skills:

- Being able to justify in writing the work done, including the analysis of different options and why one of them was selected.
- Being able to discuss issues orally on the subject.

Being able to collaborate with others in problem solving and implementation of programs, participating in the organization and review of group work.

DESCRIPTION OF CONTENTS

1. Introduction

Theory and problems (3T)
- Definition and purpose of operating systems
- Milestones in the development of operating systems
- Operating system performance

Laboratory
- Creating a virtual machine and installing a Linux operating system (2.5 hours)
- Shell scripting(2.5 hours)
- C Language (2.5 hours)

2. Processes and threads

Theory and problems (3T +1P)
- Concept of process Concept
- Creating and destroying
- Context change
- Multithreaded processes
- Creating and destroying threads
- Advantages and disadvantages of using multiple threads

3. Processor scheduling

Theory and problems (5T+2P)
- Short, medium and long-term scheduling
- Scheduling algorithms for single processsor systems
- Multiprocessor and real-time scheduling

Laboratory
- Process and thread creation (2,5 hours)

4. IPC and synchronization

Theory and problems (5T +3P)
- Concept of concurrency
- Communication and synchronization models
- Mutex and condition variables
- Message Passing
- Other mechanisms of communication and synchronization
- Deadlocks

Laboratory
- Concurrent Programming (2.5 hours)

5. Memory

Theory and problems (4T +2 P)
- contiguous allocation
- segmented model
- paged model
- virtual memory

6. File systems

Theory and problems (4T +1 P)
- Filesystem concept
- Logical description: files, folders, aliases, indirect files
- Physical Description: filesystem structure, free space management, space allocation
- Example cases

Laboratory
- Filesystems (2.5 hours)

7. Input/output and multimedia devices

Theory and problems (5T)
- Requirements and general structure
- Device Drivers
- Device-independent I/O software, access control, synchronous and asynchronous I/O
- User-mode code, system and I/O libraries, queue management
- Multimedia devices

Laboratory
- Multimedia Operating Systems (5 hours)

WORKLOAD

ACTIVITY	Hours	% To be attended
Theory classes	30,00	100
Laboratory practices	20,00	100
Classroom practices	10,00	100
Development of individual work	8,00	0
Preparation of evaluation activities	12,00	0
Preparing lectures	20,00	0
Preparation of practical classes and problem	30,00	0
Resolution of case studies	20,00	0
TOTAL	**150,00**	

TEACHING METHODOLOGY

Theorical classroom activities will be used to introduce the main points of the subject, providing a global and inclusive vision, analyzing in detail key issues, encouraging student participation. These activities are complemented by practical activities in order to apply the basics and expand the knowledge and experience. They include the following types of classroom activities:

- Problem-solving sessions. (Individually and in groups).
- Labs. (In couples).
- Evaluation quizzes.

In addition to classroom activities, students must perform personal homework (directed bibliographic research, questions, problems, preparation of classroom activities, study). These tasks will be primarily on an individual basis, in order to promote autonomous work, but they will also include work requiring the participation of small groups of students (2-4) to build team work skills.

The e-learning platform of the University of Valencia will be used to support communication with students. This platform will provide access to course materials.

EVALUATION

The evaluation of the subject will be carried out on the first call by:

Assessment of theory and problems (TP).

This part will have a weight of 75% of the final grade and it will be necessary to reach a 4 out of 10 to average. In turn, the weights of this part are divided into:

- 30% for continuous assessment (CE), based on participation and degree of involvement in the teaching-learning process, taking into account regular attendance at planned face-to-face activities and resolution of proposed questions and problems.
    - 70% of this part corresponds to academic work and problems done at home
    - 30% corresponds to class attendance (either face-to-face or by videoconference)
- 70% for the individual objective tests that will consist of both theoretical-practical questions and problems.
    - 33% of this part corresponds to the first partial T1 (it is done at the end of the first half of the semester) and contains the first 3 topics. It will be necessary to take a minimum of 4.5 to remove material from the final test.
    - 33% of this part corresponds to the second partial T2 (it is carried out during the second half of the semester) and contains topics 4 and 5. It will be necessary to take a minimum of 4.5 to remove material from the final test.
    - 33% - 66% - 100% (depending on the subject eliminated) of this part corresponds to the Final Test of the final topics of the course and of the partial T1 and T2 where the 4.5 has not been passed. Additionally, a student may request a week in advance the repetition of any of the T1 and T2 tests, thus cancelling the previous grade of said test.
    -

Evaluation of practical laboratory activities (L) based on the achievement of objectives in laboratory sessions.

These activities will be carried out in pairs, their weight will be 25% of the final grade and it will be necessary to reach a 4 out of 10 to average (both in the first and second calls). All laboratory sessions will have the same weight on the final grade.

In case of not being able to attend a session, the student may submit the corresponding work to their laboratory teacher. The delivery must be in person, during tutoring hours and the student must be prepared to answer questions about the practice and to carry out parts of it at the moment (with small changes). This type of delivery must be made before any laboratory group has carried out the practice and will have a 20% penalty.

In the **second call** the subject will be evaluated in the same way as in the first call, with the following exceptions:

- The mark of the continuous evaluation will be calculated without taking class attendance into account, so the weight of the academic work will be 100%. A deadline for submitting new works will be opened if you want to improve the grade, but it will be taken into account that there will be a 40% penalty. The delivery limit will be the day before the examination of the second call.
- The exam of the second call will follow the same rules as in the first call, so it will only be evaluated for the parts of the subject not previously approved. In the same way, you can also request the repetition of some part to improve the grade.
- An internship delivery period will be opened with the same conditions as in the 1st call (logically they will not be carried out in the laboratory), except that the penalty will be 30%. The delivery limit will be the day before the examination of the second call.

REFERENCES

Basic

- Sistemas Operativos. William Stallings. Prentice Hall.
- Fundamentos de Sistemas Operativos. Abraham Silberschatz, Peter Baer Galvin y Greg Gagne. John Wiley & Sons.
- Sistemas Operativos. Una visión aplicada. Jesús Carretero, Félix García, Pedro de Miguel y F. Pérez. McGraw-Hill.

Additional

- Programación estructurada en C. James L. Antonakos, Kenneth C. Mansfield. Prentice Hall.
- Unix and Linux System Administration Handbook, Evi Nemeth, Garth Snyder, Trent R. Hein, Ben Whaley. Prentice Hall. (Libro electrónico).
- Administración de sistemas Linux, Evi Nemeth, Garth Snyder y Trent R. Hein. Anaya.