Course guide
270063 - SO2 - Operating Systems II

Last modified: 11/07/2025

Unit in charge: Barcelona School of Informatics
Teaching unit: 701 - DAC - Department of Computer Architecture.

Degree: BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2010). (Optional subject).

Academic year: 2025 **ECTS Credits:** 6.0 **Languages:** Catalan, Spanish

LECTURER

Coordinating lecturer: JUAN JOSÉ COSTA PRATS

Others: Primer quadrimestre:
JUAN JOSÉ COSTA PRATS - 10

PRIOR SKILLS

The student must have the technical capabilities that will confer the subjects studied previously together with a medium level of technical English to read and understand documentation.

The technical capabilities could be summarized as follow:

-Operating systems: Understanding the basics of an operating system along with the creation of applications using the generic system call interface explained during the Operating System course.
-In terms of computer architecture: Knowledge of the main elements of a computer, how these elements relate to each other, internal representation of data and knowledge of the assembler language.
-In terms of programming: Ability to code complex programs from scratch composed of several moduls. Definition of data types, pointers and references, and assembler code. Compilation and linkage of executables.

REQUIREMENTS

- Prerequisite SO

DEGREE COMPETENCES TO WHICH THE SUBJECT CONTRIBUTES

Specific:
CEC2.1. To analyse, evaluate, select and configure hardware platforms for the development and execution of computer applications and services.
CEC2.2. To program taking into account the hardware architecture, using assembly language as well as high-level programming languages.
CEC2.3. To develop and analyse software for systems based on microprocessors and its interfaces with users and other devices.
CEC2.4. To design and implement system and communications software.
CEC2.5. To design and implement operating systems.
CEC3.1. To analyse, evaluate and select the most adequate hardware and software platform to support embedded and real-time applications.
CEC3.2. To develop specific processors and embedded systems; to develop and optimize the software of these systems.
CEC4.1. To design, deploy, administrate and manage computer networks.
CEC4.2. To demonstrate comprehension, to apply and manage the guarantee and security of computer systems.
CT6.1. To demonstrate knowledge and capacity to manage and maintain computer systems, services and applications.
CT6.3. To demonstrate knowledge about the characteristics, functionalities and structure of the Operating Systems allowing an adequate use, management and design, as well as the implementation of applications based on its services.
CT7.3. To determine the factors that affect negatively the security and reliability of a hardware/software system, and minimize its effects.

Generical:
G6. SOLVENT USE OF THE INFORMATION RESOURCES: To manage the acquisition, structuring, analysis and visualization of data and information of the field of the informatics engineering, and value in a critical way the results of this management.
G8. APPROPIATE ATTITUDE TOWARDS WORK: to have motivation to be professional and to face new challenges, have a width vision of the possibilities of the career in the field of informatics engineering. To feel motivated for the quality and the continuous improvement, and behave rigorously in the professional development. Capacity to adapt oneself to organizational or technological changes. Capacity to work in situations with information shortage and/or time and/or resources restrictions.

TEACHING METHODOLOGY

The course will have mainly two types of class: theory and laboratories. The theory classes explain the concepts, designs and implementing various components of current operating systems. These classes will be supported by slides.

Laboratory classes will be weekly. During the lab you will design and implement an operating system called Zeos. This project will be done in groups of two people.

LEARNING OBJECTIVES OF THE SUBJECT

1.Knowing how a real OS works, since the computer boots and initializes the operating system, through the dynamic management of resources, till the the computer is shutdown.
2.Details of the implementation of some of the basic components of a real OS: initialization code, memory management code, input/output management code, processes management code and upload of executable files code.
3.Understand the mechanism of dynamic insertion of code in a real OS.
4.Be able to implement a device driver and insert it into a real OS.
5.Know the multithreaded programming, the problem of sharing memory and the mechanisms of mutual exclusion
6.Know some risk factors in the security of an operating system.
7.Understand the implementation of various security and access policies.
8.Be able to implement some of the basic components of a real OS: initialization code, memory management code, input / output management code , process management code and load executable files code.
9.Developing system software for a specific architecture such as Intel x86. Using C and assembler.
10.Select the most appropriate settings of an operating system depending on the platform to use
11.Find, organize and summarize relevant and quality information on an unknown topic. Write documents with a critical thinking on the main ideas of this topic.
12.Describe the basic concepts of OS view of computer networks.
13.To have a proactive attitude to quality and continuous improvement
14.Be able to adapt to situations of time constraints and / or resources and / or lack of information

STUDY LOAD

Type	Hours	Percentage
Hours large group	30,0	20.00
Guided activities	6,0	4.00
Hours small group	30,0	20.00
Self study	84,0	56.00

Total learning time: 150 h

CONTENTS

System boot

Description:
This chapter will explain all the actions carried out to start all the operating system services. We also describe and discuss possible implementations of this process in the current operating system.

Mechanisms to enter the system

Description:
Describes and discusses the possible implementation of the mechanisms to enter the system along with its hardware support. Important aspects of these mechanisms and dependencies between the operating system and hardware will be detailed. We discuss the benefits of implementing virtualization techniques today. Presents and discusses the various implementations of operating systems today.

Memory Management

Description:
This chapter covers the following topics: The logical address space of the process. Page-based memory systems and their hardware support. Design and implementation of virtual memory. Replacement algorithms for virtual memory. Implementation of shared memory. Different memory systems design and implementation in modern operating systems and their hardware dependencies.

Process Management

Description:
This chapter covers the following topics: Detailed Implementation of the process control block. Implementation details for the process of creating processes. Implementation details for the process of finishing processes. Implementation details for the process of loading executables. Implementation details of context switching between processes. Detailed Description of the structures and algorithms for process scheduling. Implementation details of the processes scheduling routines. Description and discussion of various implementations of process scheduling in modern operating systems.

Extending the system kernel

Description:
This chapter covers the following topics: Concept, design and implementation of kernel modules of the OS. Loading kernel modules at system initialization time and run time. Description of access to features implemented in kernel modules. Understanding the relationship between kernel modules and physical and logical devices.

Input / output Management and file systems

Description:
This chapter covers the following topics: Description and implementation of the OS part independent and dependent of the device. Implementation of a device descriptor. Description, features and implementation of the I/O structures and file systems. Description and implementation of the logical structure of file systems on disk. Communication mechanisms between processes via file systems. Communication mechanisms between processes across the network. Description and implementation of various file systems today.

Shared Memory

Description:
This chapter covers the following topics: Problems with shared memory between multiple threads of execution within a process. Race conditions. Areas of mutual exclusion. Description and implementation of mechanisms for mutual exclusion in accessing shared memory, in particular, test and set, mutex and semaphores. Description, discussion and implementation of mechanisms for mutual exclusion in modern operating systems.

ACTIVITIES

Become familiar with the workplace

Description:
Acquiring the knowledge necessary to develop the laboratory.

Specific objectives:
1, 2, 8, 9

Full-or-part-time: 4h
Self study: 2h
Laboratory classes: 2h

Interrupt Service Routine

Description:
Develop an ISR for the clock interrupt

Specific objectives:
1, 2, 8, 9, 13, 14

Related competencies :
G8. APPROPIATE ATTITUDE TOWARDS WORK: to have motivation to be professional and to face new challenges, have a width vision of the possibilities of the career in the field of informatics engineering. To feel motivated for the quality and the continuous improvement, and behave rigorously in the professional development. Capacity to adapt oneself to organizational or technological changes. Capacity to work in situations with information shortage and/or time and/or resources restrictions.

Full-or-part-time: 8h
Self study: 4h
Theory classes: 2h
Laboratory classes: 2h

Creating an entry point into the system

Description:
Creating a new entry point into the system to use system services

Specific objectives:
1, 2, 6, 8, 9, 13, 14

Related competencies :
G8. APPROPIATE ATTITUDE TOWARDS WORK: to have motivation to be professional and to face new challenges, have a width vision of the possibilities of the career in the field of informatics engineering. To feel motivated for the quality and the continuous improvement, and behave rigorously in the professional development. Capacity to adapt oneself to organizational or technological changes. Capacity to work in situations with information shortage and/or time and/or resources restrictions.

Full-or-part-time: 7h
Self study: 4h
Theory classes: 2h
Laboratory classes: 1h

Memory management

Description:
It describes the main algorithms and implementation of virtual memory.

Specific objectives:
1, 2, 6, 7, 8, 9

Full-or-part-time: 15h
Self study: 8h
Theory classes: 4h
Laboratory classes: 3h

Process Management

Description:
Develop the basic structures of process management, as well as features to identify, create and destroy processes

Specific objectives:
1, 2, 6, 8, 9, 10, 13, 14

Related competencies :
G8. APPROPIATE ATTITUDE TOWARDS WORK: to have motivation to be professional and to face new challenges, have a width vision of the possibilities of the career in the field of informatics engineering. To feel motivated for the quality and the continuous improvement, and behave rigorously in the professional development. Capacity to adapt oneself to organizational or technological changes. Capacity to work in situations with information shortage and/or time and/or resources restrictions.

Full-or-part-time: 20h
Self study: 8h
Theory classes: 6h
Laboratory classes: 6h

First Control

Description:
First part of the course

Specific objectives:
1, 2, 6, 8, 9

Full-or-part-time: 8h
Self study: 6h
Guided activities: 2h

First Lab Exam

Description:
First Lab Exam

Specific objectives:
1, 2, 6, 7, 8, 9, 12

Full-or-part-time: 8h
Self study: 6h
Guided activities: 2h

Process scheduling

Description:
Implementing the change of context and a simple planner based on quantum

Specific objectives:
2, 6, 8, 9, 13, 14

Related competencies :
G8. APPROPIATE ATTITUDE TOWARDS WORK: to have motivation to be professional and to face new challenges, have a width vision of the possibilities of the career in the field of informatics engineering. To feel motivated for the quality and the continuous improvement, and behave rigorously in the professional development. Capacity to adapt oneself to organizational or technological changes. Capacity to work in situations with information shortage and/or time and/or resources restrictions.

Full-or-part-time: 12h
Self study: 6h
Laboratory classes: 6h

Kernel extension

Description:
We describe the mechanisms used to extend the kernel of an operating system

Specific objectives:
1, 2, 3, 4, 6, 9, 10

Full-or-part-time: 5h
Self study: 4h
Theory classes: 1h

Management I / O and file systems

Description:
Develop some of the mechanisms of input / output file system of an operating system

Specific objectives:
1, 2, 6, 7, 8, 9, 10, 12, 13, 14

Related competencies :
G8. APPROPIATE ATTITUDE TOWARDS WORK: to have motivation to be professional and to face new challenges, have a width vision of the possibilities of the career in the field of informatics engineering. To feel motivated for the quality and the continuous improvement, and behave rigorously in the professional development. Capacity to adapt oneself to organizational or technological changes. Capacity to work in situations with information shortage and/or time and/or resources restrictions.

Full-or-part-time: 16h
Self study: 7h
Theory classes: 5h
Laboratory classes: 4h

Interprocess communication

Description:
Implementation of semaphores

Specific objectives:
1, 2, 5, 6, 8, 9, 13, 14

Related competencies :
G8. APPROPIATE ATTITUDE TOWARDS WORK: to have motivation to be professional and to face new challenges, have a width vision of the possibilities of the career in the field of informatics engineering. To feel motivated for the quality and the continuous improvement, and behave rigorously in the professional development. Capacity to adapt oneself to organizational or technological changes. Capacity to work in situations with information shortage and/or time and/or resources restrictions.

Full-or-part-time: 10h
Self study: 7h
Theory classes: 2h
Laboratory classes: 1h

Network

Description:
Mount a small network with a specialized service

Specific objectives:
6, 7, 10, 11, 12

Related competencies :
G6. SOLVENT USE OF THE INFORMATION RESOURCES: To manage the acquisition, structuring, analysis and visualization of data and information of the field of the informatics engineering, and value in a critical way the results of this management.

Full-or-part-time: 5h
Self study: 4h
Theory classes: 1h

Second Control

Description:
Second control

Specific objectives:
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12

Full-or-part-time: 8h
Self study: 6h
Guided activities: 2h

Project

Description:
Delivery of the programming project code and documentation.

Specific objectives:
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12

Full-or-part-time: 6h
Self study: 6h

Final Theory Exam

Description:
Final exam for the course

Specific objectives:
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12

Full-or-part-time: 9h
Self study: 6h
Guided activities: 3h

Final lab exam

Specific objectives:
2, 3, 4, 5, 6, 7, 8, 9, 10

Full-or-part-time: 9h
Self study: 6h
Guided activities: 3h

GRADING SYSTEM

The final grade for the course consists of the grade of the core technical competences (CT), and the grade of generic competence (CTr) by the formula:

Final Grade = (CT + CTr) * (10/11)

Where the maximum score of CTr is 1.

The grade of the CT can be obtained by continuous assessment (CTc) or excepcionally by a final exam (CTf). Is calculated as:

CT = max (CTc, CTf)

Where CTc grade is evaluates the theory (T) and laboratory (L) components. The formula for calculating this grade is as follows:

CTc = 50% T + 50% L

To calculate T two assessments are used with the same weights:

T = 50% T1 + 50% T2

To calculate L, an exam (L1), a follow up grade (S) and a project (P) is used, with the following weights:

L = 40% L1+ 10% S + 50% P

The follow up grade (S) is assigned by the teacher as an evaluation of the right progress throughout the laboratory.

The project grade (P) corresponds to the design, follow up and final implementation of the project.

The CTf grade is calculated using a theory exam (T) and a laboratory exam (L). To accomplish this grade is mandatory to do both exams. This option is only available for those students that fail the continuous assessment. The formula is as follows:

CTf = 50% T + 50% L

The grade of generic competence (CTr) is obtained during the semester through various activities. Rating of this competence will have values A, B, C, D or NA: A corresponds to an excellent level, B to a desired level, C to a sufficient level, D to a failed level, and NA to not assessed.

BIBLIOGRAPHY

Basic:

- Silberschatz, A.; Galvin, P.B.; Gagne, G. Operating system concepts. 10th ed. John Wiley & Sons, 2019. ISBN 9781119454083.

Complementary:

- Bovet, D.P.; Cesati, M. Understanding the Linux kernel. 3rd ed. O'Reilly, 2005. ISBN 9780596005658.
- Kernighan, B.W.; Ritchie, D.M. C: programming language. 2nd ed. Prentice Hall, 1988. ISBN 0131103628.
- Russinovic, M.; Salomon, D.A.; Ionescu, A. Windows internals. 6th ed. Microsoft Press, 2012. ISBN 9780735648739.