DEGREE CURRICULUM

OPERATING SYSTEMS

Coordination: MATEO FORNES, JORDI

Academic year 2022-23

OPERATING SYSTEMS 2022-23

Subject's general information

<table>
<tr><td>Subject name</td><td>OPERATING SYSTEMS</td></tr>
<tr><td>Code</td><td>102377</td></tr>
<tr><td>Semester</td><td>1st Q(SEMESTER) CONTINUED EVALUATION</td></tr>
<tr><td>Typology</td><td>
<table>
<tr><th>Degree</th><th>Course</th><th>Character</th><th>Modality</th></tr>
<tr><td>Bachelor's degree in Digital Interaction and Computing Techniques</td><td>2</td><td>COMPULSORY</td><td>Attendance-based</td></tr>
</table>
</td></tr>
<tr><td>Course number of credits (ECTS)</td><td>6</td></tr>
<tr><td>Type of activity, credits, and groups</td><td>
<table>
<tr><th>Activity type</th><td>PRALAB</td><td>TEORIA</td></tr>
<tr><th>Number of credits</th><td>3</td><td>3</td></tr>
<tr><th>Number of groups</th><td>1</td><td>1</td></tr>
</table>
</td></tr>
<tr><td>Coordination</td><td>MATEO FORNES, JORDI</td></tr>
<tr><td>Department</td><td>COMPUTER SCIENCE AND INDUSTRIAL ENGINEERING</td></tr>
<tr><td>Teaching load distribution between lectures and independent student work</td><td>Overall, the course has 150 hours of work distributed with 60 hours of class attendance and 90 hours of individual work of the student.
6 ECTS = 25 * 6 = 150 hours of work
40% -> 60 classroom hours (face-to-face)
60% -> 90 hours of autonomous work of the student</td></tr>
<tr><td>Important information on data processing</td><td>Consult this link for more information.</td></tr>
<tr><td>Language</td><td>Catalan (Spanish if any student shows difficulties with Catalan). All the supplementary material and technical documentation in English.</td></tr>
</table>

OPERATING SYSTEMS 2022-23

Teaching staff	E-mail addresses	Credits taught by teacher	Office and hour of attention
MATEO FORNES, JORDI	jordi.mateo@udl.cat	6	

Subject's extra information

To take this course it is highly recommended to have passed the *Computer Organization and Computer Structures courses*, as well as *Algorithms and Programming*.

Operating systems are the central pillar of any computer system. This course studies fundamental design and implementation ideas in the engineering of operating systems. Lectures are based on a study of UNIX and research papers. Topics include process management, communication mechanisms, task scheduling, and memory management . Individual laboratory assignments involve coding in C using system calls, and shell scripting.

The study of this subject will be completed with the subject of third-year *Administration of Systems and Virtualization*.

Learning objectives

- To determine the functional characteristics and design of the elements that make up an Operating System (OS).
- Analyze the importance of each module that make up an operating system.
- To identify the different services provided by the operating system to users and applications.
- Efficient use of services provided by the OS for the design and development of computer applications.
- Critically analyze the characteristics and functionalities of the policies that make up an operating system.
- Applying the techniques described to other problems.
- Critically compare the different mechanisms of memory management.

Competences

Basic:

- *CB3*: That students have the ability to gather and interpret relevant data (usually within their area of study) to make judgments that include a reflection on relevant social, scientific or ethical issues.

Transversals:

- *CT3*: Acquire training in the use of new technologies and information and communication technologies.
- *CT6*. Apply the gender perspective to the tasks of the professional field.

Generals:

- *CG3*: Capacity to use appropriate hardware and software platforms for the development and execution of interactive digital applications.

Specific:

- *CE3*: Basic knowledge of the use and programming of computers, operating systems and databases, and their use in the development of interactive applications.
- *CE7*: Know, manage and maintain systems, services and interactive applications.
- *CE11*: Knowledge of the characteristics, functionalities and structure of operating systems and design and implement applications based on their services.

Subject contents

Topic 1: Introduction

- Concept of an operating system
- Objectives
- History of the operating systems
- Types of operating systems

Topic 2: Operating System Structure

- Components of the operating system
- Services of the operating system
- Calls and programs of the system
- Case study: UNIX / LINUX

Topic 3: Process Management

- Concept of process
  - States of the processes
  - Process Control Block (PCB)
- Threads
- Communication between processes
- Types of communication
- Case study: Managing processes in UNIX.
- Case study: Communicating with pipes

Topic 4: Scheduling of the CPU

- Basic concepts
- Types of schedulers
- Performance metrics
- Scheduling algorithms
- Multilevel queues

Topic 5: Memory Management

- Basic principles
- Assigning contiguous Memory
  - Nude Machine
  - Resident Monitor
  - Multiple Partitions
- Assigning non-contiguous Memory
  - Pagination
  - Segmentation
- Combined systems
  - Page segmentation
  - Segmented pagination
- Virtual Memory
  - Demand paging
  - Effective Access Time
  - Frames allocation algorithms
  - Page replacement algorithms
  - Thrashing

Topic 6: *Shell Script programming*

- Introduction
- Programming with Bash
- Programming with AWK

Methodology

Theory

- In these sessions, the theoretical contents of the subject will be explained, accompanied by illustrative examples. The slides of the subject will be used as support material
- An active methodology is used where the student is the protagonist by performing **HandsOn**.

Practices

- These sessions will be used to work on the most practical parts of the subject Linux Programming with C and also Shell Scripting Programming.
- A **social** learning methodology will be used with **live-coding** sessions, where we will read and implement, and propose activities to solve in the classroom with the help of the rest of the students.

Self-employment

- A **cooperative methodology based on challenges** will be used, where students in groups will have to start from the basis of the knowledge acquired in the theoretical and practical sessions of the course and complement it with external information.

Development plan

Week	Theory	Laboratory	Homework
1	**T1**	*Lab-01*	**EV1:** Act01
2	**T2**	*Lab-02*	**EV1:** Act02
3	**T2**	*Lab-03*	**EV1:** Act03
4	**T3**	*Lab-04*	**EV1:** Act04
5	**T3**	**HOLIDAYS**	**EV2:** Act05
6	**T3**	*Lab-05*	**EV2:** Act06
7	**T4**	*Lab-06*	**EV3:** Act07
8	**T4**	*Lab-07*	**EV3:** Act08
9	First Exam (E1)		
10	**T5**	*Lab-08*	**EV4:** Act09

OPERATING SYSTEMS 2022-23

Week	Theory	Laboratory	Homework
11	**T5**	*Lab09*	**EV4:** Act10
12	**T5**	*Lab-10*	**EV5:** Act11
13	**T6**	**HOLIDAYS**	**EV5:** Act12
14	**T6**	*Lab-11*	**EV5:** Act13
15	**T6**	*Lab-12*	**EV5:** Act14
16	Second Exam (E2)		
17			
18			
19	**Recovery**		

Evaluation

Acr.	Evaluation Activities	Weight	Minimum Mark	Groups	Recoverable
E1	1st partial exam	15%	NO	NO	YES
E2	2nd partial exam	15%	NO	NO	YES
PART	Tracking and Participacion	10 %	NO	NO	NO
AV1	Evaluation activities(Kernel)	10 %	NO	YES	NO
AV2	Evaluation activities (Process)	15 %	NO	YES	NO
AV3	Evaluation activities (Scheduling)	10 %	NO	YES	NO
AV4	Evaluation activities (Memory)	10%	NO	YES	NO
AV5	Evaluation activities (Shell Scripting)	15 %	NO	YES	NO

FINAL MARK: 15% E1 + 15% E2 + 10% AV1 + 15% AV2 + 10% AV3 + 10% AV4 + 15% AV5 + 10% PART
*** To pass the course, **FINAL MARK** must be *greater than or equal to* **5**.

Considerations:

- The examns are done with a computer and the students are allowed to consult their notes. Both practical and theoretical content is evaluated.
- In case of plagiarism, the grade for that activity is 0.
- The **Tracking and Participacion** activity represents the realization and delivery of HandsOn and Labs realized in class, and the presentation of problems or complementary activities proposed in the classroom, as well as student participation in classroom discussions, *slack* channels, and keeping personal notes about their learning on *github.*
- The presentation of activities **(AV1,AV2,AV3,AV4,AV5)** with retard represents a weighting of 75% on the weighting of that activity, for example, if the activity has a weighting of 10% in the final grade, it will have a weighting of 7.5%. Therefore, these activities are not recoverable, since they can be delivered throughout the course but with a 25% penalty.
- Partials **(E1, E2)** are **retrievable** through a make-up exam.

Bibliography

Basic Bibliografy:

- Francesc Solsona. “Sistemes Operatius. Teoria aplicada”. Edicions de la Universitat de Lleida (Col·leció eines 78). ISBN: 978-84-8409-747-1. 2015.

Additional Bibliografy:

- Linux Kernel in a Nutshell, by Greg Kroah-Hartman, published by O'Reilly.
- Evi Nemeth, Garth Snyder, Trent R. Hein, Ben Whaley, and Dan Mackin. 2017. *UNIX and Linux System Administration Handbook (5th Edition)* (5th. ed.). Addison-Wesley Professional.
- Silberschatz A., Peterson J. Y Galvin P.: “Sistemas Operativos. Conceptos Fundamentales”; Addison-Wesley, 1999.
- Carretero Pérez, Jesús, y otros: “Sistemas Operativos. Una Visión Aplicada”. McGraw-Hill, 2001.
- F.M. Marquez García: “Unix. Programación Avanzada”, Edt. Rama 3aedició, 2004.
- E.Quigley: “UNIX Shells by Example”, Edt. Prentice-Hall, 3ra edició, 2002
- Tanenbaum,Andrew S. “Sistemas Operatius, Diseño e Implementación”, 2aedició, Edt. Prentice- Hall, 1998.
- TackettJ. y Gunter D., “Utilizando Linux”, Prentice Hall, 1996
- KayA. Robbins, Steven Robbins, "UNIX Programación Práctica. Guía para la Concurrencia, la Comunicación y los Multihilos", Edt.Prentice-Hall, 1997.
- Afzal, A.: Introduccióna Unix. Un enfoque práctico. Ed. Prentice Hall, 1997. [Tac96]Tackett J. y Gunter D.: Utilizando Linux 2a. Prentice Hall, 1996