Universitat de Lleida

DEGREE CURRICULUM

OPERATING SYSTEMS

Coordination: MATEO FORNÉS, JORDI

Academic year 2020-21

OPERATING SYSTEMS 2020-21

Subject's general information

Subject name	OPERATING SYSTEMS
Code	102377
Semester	1st Q(SEMESTER) CONTINUED EVALUATION
Typology	

Degree	Course	Character	Modality
Bachelor's degree in Digital Interaction and Computing Techniques	2	COMPULSORY	Attendance-based

Course number of credits (ECTS)	6
Type of activity, credits, and groups	

Activity type	PRALAB	TEORIA
Number of credits	3	3
Number of groups	1	1

Coordination	MATEO FORNÉS, JORDI
Department	COMPUTER SCIENCE AND INDUSTRIAL ENGINEERING
Teaching load distribution between lectures and independent student work	Overall, the course has 150 hours of work distributed with 60 hours of class attendance (30 hours at class, 15 hours virtual, and 15 hours virtual asynchronous) and 90 hours of individual work of the student. 6 ECTS = 25 * 6 = 150 hours of work 20% -> 30 classroom hours (face-to-face) 10% -> 15 virtual classroom hours (face-to-face) 10% -> 15 virtual asynchronous hours (face-to-face) 60% -> 90 hours of autonomous work of the student
Important information on data processing	Consult this link for more information.
Language	Catalan (Spanish if any student shows difficulties with Catalan). All the supplementary material and technical documentation in English.

OPERATING SYSTEMS 2020-21

Teaching staff	E-mail addresses	Credits taught by teacher	Office and hour of attention
MATEO FORNÉS, 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. In this subject, the concept of an operating system is introduced. Moreover, process management, communication mechanisms, task scheduling, and memory management used by the operating system are described and discussed.

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.

Generals:

- *CG3*: Use adequate hardware and software platforms to develop and execute 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 structures of the operating systems and design and implement applications based in their services.

OPERATING SYSTEMS 2020-21

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

A **Blended Learning** methodology is used that combines face-to-face and virtual sessions synchronous with asynchronous virtual sessions.

Virtual 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.
- Attendance at these sessions will not be mandatory but recommended. All virtual sessions will be performed using the virtual campus videoconferencing tool and will be recorded for consultation at any moment.

Asynchronous Virtual Theory

- An active methodology is used where the student is the protagonist by performing **HandsOn**.
- These sessions will be recorded on video and will have additional material (slides or documents) and will consist of performing different **HandsOn** or solving related problems related to the theoretical contents.

Practices (Face-to-face Sessions)

- 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. The course will propose 4 major challenges related to the theoretical contents of the subject.
- At the end of each challenge there will be a collective reflection using the methodology of reflection metacognitive to assess learning:
    - What have I learned from doing the challenge?
    - How did I learn that?
    - What has allowed me to improve?
    - Why did it help me?
    - Why can it serve me?
- At the end of the course, there will be a validation and oral presentation of all the challenges together. Where students will discuss their solutions with the other groups and make a self-evaluation.

OPERATING SYSTEMS 2020-21

Development plan

Week	Theory (Virtual)	Async Theory (Virtual)	Laboratory (Face-to-face)	Homework
1	**Topic 1**: Presentation / Introduction	***HandsOn-01:*** *Introducció a Linux*	Unix Programming with C	**R1**:Unix Kernel
2	**Topic 2:**Structure of the Operating System	***HandsOn-02:*** *Spying and modifing the Kernel*	Unix Programming with C	
3	**Topic 3**: Process Management	Process Management Problems	**Topic 6:** *Shell Script Programming*	
4	**Topic 3**: Process Management (Pipes)	***HandsOn-03:*** *Gestión de Procesos*	**Topic 6:** *Shell Script Programming*	**R2**: Unix Shell
5	**Topic 3**: Process Management (Signals)	***HandsOn-04:*** *Room Scape*	Process Management Problems	
6	**Topic 3**: Process Management (Threads)	***HandsOn-05:*** *Threads*	**Topic 6:** *Shell Script Programming*	
7	**Topic 4**: Scheduling of the CPU	Scheduling of the CPU Problems	**Topic 6:** *Shell Script Programming*	
8	**Topic 4**: Scheduling of the CPU	Scheduling of the CPU Problems	**Topic 6:** *Shell Script Programming*	**R3:** Daemon
9	**1st Exam**			
10	**Topic 5:** Memory Management	Memory Management Problems	**Topic 6:** *Shell Script Programming*	
11	**Topic 5:** Memory Management	Memory Management Problems	**Topic 6:** *Shell Script Programming*	**R4**: CPU scheduler simulator
12	HOLIDAY	***HandsOn-06:*** Introduction to RUST	**Topic 6:** *Shell Script Programming*	
13	**Topic 5:** Memory Management(Virtual Memory)	***HandsOn-07:*** Memory Management with RUST	**Topic 6:** *Shell Script Programming*	
14	Virtual Memory Problems	HOLIDAY	HOLIDAY	
15	HOLIDAY	HOLIDAY	Validation and Presentations	
16	**2nd Exam**			
17				
18				
19	**Recovery**			

NOTES: In the 20/21 academic year:

- The theory is virtual using the campus video conferencing tool. Attendance is not mandatory. Sessions will be recorded.
- The asynchronous virtual activities are carried out during the week when the student is available, using the material that will be available in the Virtual Campus.
- Laboratory activities (PRALAB) are face-to-face and compulsory attendance.

- Partials and recoveries are face-to-face. ***Except for a new situation of exceptionality***.

Evaluation

Evaluation Activities	Weight	Minimum Mark	Groups	Mandatory	Recoverable
First Exam (P1)	15%	YES >= 5	NO	YES	YES
Second Exam (P2)	15%	YES >= 5	NO	YES	YES
Challenge *1 (R1)*	10 %	YES >= 5	YES	YES	YES
Challenge *2 (R2)*	15%	YES >= 5	YES	YES	YES
Challenge *3 (R3)*	10 %	YES >= 5	YES	YES	YES
Challenge *4 (R4)*	15 %	YES >= 5	YES	YES	YES
Tracking	20%	NO	NO	NO	NO

Final mark: 15% *P1* + 15% *P2* + 10% *R1* + 15% *R2* + 10% *R3* + 15% *R4* + 20% *Tracking*

*** To pass the course, the final grade 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** activity represents the realization and delivery of the asynchronous **HandsOn**, and the presentation of problems or complementary activities proposed in the classroom, as well as the participation in the face-to-face sessions.
- The presentation of activities 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%.
- An additional **0.25** points will be awarded to the final grade to the best group; selected by the rest of the classmates in the validation session and presentation of the challenges.

Recoveries:

- **Challenges**: Challenges can be recovered by submitting an improved version with a maximum grade of 7.5. Each 1 of the 4 challenges is recoverable separately. They cannot be recovered jointly.
- **Examns**. They can be recovered by taking a similar exam that allows them to obtain a minimum grade of 5 or a maximum of 7.5.
  Tracking: Not recoverable. It is an activity that rewards the work and effort of students during the course.

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