Universidad de Alcalá

TEACHING GUIDE

Advanced Operating Systems

Bachelor's Degree in
Computer Science
Computer Engineering

Universidad de Alcalá

Course 2019/2020

3rd Course – 1st Term

TEACHING GUIDE

Subject name:	Advanced Operating Systems
Code:	780012
Degree:	Graduate on Computer Science Graduate on Computer Engineering
Department and area:	Computer Engineering dpt. *(Dpto. Automática)* Computer Architecture and Technology
Type:	Obligatory
ECTS credits:	6
Course and term:	3rd course, 1st term
Teaching staff:	Óscar García Población
Tutorship hours:	Available in Aula Virtual
Language:	English

1. Presentation

Advanced Operating Systems is a compulsory subject imparted in the first term of the second course of the Graduate on Computer Engineering (third term of the whole degree), with a total of 6 ECTS credits. It continues the matter introduced in Operating Systems.

The goal of this subject is making the student acquire the required knowledge related to the structure and design of Operating Systems. The student will be able to understand, on one side, which are the basic components that form an Operating System as well as their interconnection and, on the other side, the fundamentals of the current Operating Systems. This will help the student to develop his/her skills in the professional environment.

This subject is based on Programming Fundamentals, Operating Systems and Computer Structure and Organization and establishes the fundamentals of Operating Systems Administration; thus, it is recommended to continue with this later subject for specialization.

Prerequisites and recommendations:

It is recommended to have completed Computer Structure and Organization and Operating Systems.

2. Competences

General skills:

CG4 Ability to define, evaluate and select hardware and software platforms for the development and implementation of systems, services and applications, according to the knowledge acquired as provided in paragraph 5 of resolution BOE-A-2009-12977.

CG6 Ability to design and develop systems or centralized or distributed architectures integrating hardware, software and networks according to the knowledge acquired as provided in paragraph 5 of resolution BOE-A-2009-12977.

CG8 Knowledge of basic materials and technologies that enable learning and development of new methods and technologies, as well as to equip them with great versatility to adapt to new situations.

CG9 Ability to solve problems with initiative, decision making, autonomy and creativity. Ability to communicate and transmit knowledge and skills of the profession of Technical Engineer.

Specific skills:

CI5 Knowledge, management and maintenance systems, services and applications.

CI6 Knowledge and application of basic algorithmic procedures of computer to design solutions to problems technologies, analyzing the suitability and complexity of the proposed algorithms.

CI7 Knowledge, design and use efficiently the types and structures more suited to solving a problem data.

CI9 Ability to know, understand and evaluate the structure and architecture of computers as well as the basic components that make them up.

CI10 Knowledge of features, functionality and structure of operating and design and implement applications based services systems.

Learning Competences:

RA1: To develop programs using the operating system API [OS2]

RA2: To develop physical and virtual management systems [CE-OPS4]

RA3: To understand and apply memory management mechanisms.

RA4: To understand and apply memory management algorithms.

RA5: To understand and apply input/output algorithms.

RA6: To analyse the internal organization of a file system.

RA7: To understand and apply the ethics of the engineering profession and their implications with the responsibility to society, to clients and to the profession itself.

3. Contents

Content blocks	Total hours

Theme I: Memory management
 	
1. Principles of memory management:
  1. Memory hierarchy on a computer.
  2. Space-locality and time-locality.
  3. Fragmentation.
2. The memory management of a process.
3. Memory management mechanisms.
  1. Segmentation.
  2. Paging.
  3. Paged segmentation.
4. Case studies.
	16 hours

Theme II: Virtual memory management
 	
5. Introduction to virtual memory.
6. Concepts:
  1. Dynamic load.
  2. Pagers.
  3. Working set of a process.
7. Virtual memory management algorithms:
  1. Assignment policies.
  2. Allocation policies.
  3. Fetching policies.
  4. Page replacement policies.
8. Case studies.
	16 hours

Theme III: Input-output management 9. I/O software layers. 10. I/O elements. 11. Memory mapped I/O. 12. I/O execution techniques. 13. Device example: disc management.	6 hours

Theme IV: File system
 	
14. File system functionalities.
15. Disc partitioning and formatting.
16. File system reliability.
17. File system interface.
18. Memory mapped files.
19. File system structure.
    1. Files.
    2. Directories.
    3. Usage from processes.
20. Services of the file system.
21. Case studies.
	18 hours

Week / Session	Contents
01ª-04ª	• PART 1: Theory (6h) + Practices and tests (10h)
05ª-08ª	• PART 2: Theory (6h) + Practices and tests (10h)
09ª-10ª	• PART 3: Theory (6h)
11ª-14ª	• PART 4: Theory (6h) + Practice (12h)
-	• Final evaluation (2h-4h).

4. Teaching-Learning methodologies. Formative activities.

4.1. Credit distribution in hours

Number of presence hours:	Theory 28 hours + Lab 28 hours + 4 hours of assessment
Number of hours of student's independent work:	90 hours
Total hours:	150 hours

4.2. Methodological strategies, materials and didactic resources

Methodology strategies	• **Exposition or lecture with participation.** This strategy consists in the transfer of knowledge by the lecturer, encouraging critical reasoning and helping to establish links between related concepts. • **Case studies.** The lecturer will analyse in detail real world systems, designs, algorithms, etc., to explain how they work, to compare them with other solutions and to learn how to face real implementation problems and trade-offs. • **Problem based learning.** The student will gain competences applying the acquired knowledge to simulated problems, exercising algorithms, manipulating information and interpreting results. • **Project oriented learning.** The student will complete a project to solve a given problem, analysing it and providing a design that fulfils the given requisites.

Didactic material and resources	1. Bibliographic references. 2. Web resources. 3. Equipment. • Classroom. • Laboratory. • Website of the University. • Computer rooms of the School. • Library.

5. EVALUATION: Procedures, evaluation and grading criteria

5.1. Procedures

Evaluation can be done following either continuous evaluation or with a final exam. For each alternative there are two examination sessions: the ordinary and the extraordinary one.

- **Continuous evaluation**

The continuous evaluation considers the development of the competencies during all the learning procedure by using a set of formative tests distributed along the semester, so the student can learn the subject progressively.

This methodology guarantees the early feedback in the learning procedure of the student, and allows the teachers, coordinators and stakeholders in the Quality Assurance System to perform inspections and to react to certain indicators or situations.

The evaluation of the practical contents will be done at the end of its corresponding block.

- **Extraordinary evaluation**.

It has to be explicitly requested following the current regulations (*Normativa de evaluación de los aprendizajes*)

5.2. Evaluation criteria

In order to evaluate the skills acquired by the student, the following evaluation criteria will be considered:

CE1: The student takes the initiative to design algorithms and develop software using the operating system API in workgroups.

CE2: The student arguments his ideas and reflects about study cases.

CE3: The student fulfils the requested tasks, applies its own work clearly and formally.

CE4: The student applies different algorithms of virtual memory.

CE5: The student has acquired the knowledge about input-output devices in a computer system.

CE6: The student is able to design an UNIX-like file system.

CE7: The student can develop original work, properly citing sources when this works is based upon another's work, according to professional ethics in engineering.

5.3. Grading instruments

Student's performance will be evaluated according to his work, knowledge and skills acquired and his improvements on the learning procedure.

The evaluation instruments to use are:

PEI: Intermediate evaluation tests.

PL: Laboratory evaluation tests (T) and development of programs (P) related with certain operating system modules.

PEF: Final evaluation exam. This is comprehensive written exam problem solving oriented, that will be done at the end of the teaching period. Books will be allowed to be used during the exam.

5.4. Grading criteria

The relationship between learning outcomes, criteria and grading instruments is the following:

Ordinary call – continuous evaluation

Competencies	Learning outcomes	Grading criteria	Grading instruments	Weight
CG4-CG9, CI6, CI7, CI9, CI10	RA1 – RA4, RA7	CE1 – CE4, CE7	PEI1	15%
CG4-CG9, CI5, CI7, CI10	RA1, RA5, RA6, RA7	CE1 – CE3, CE5, CE6, CE7	PEI2	15%
CG4-CG9, CI6, CI7, CI9, CI10	RA1 - RA4, RA7	CE1 – CE4, CE7	PL1 (T)	10%
CG4-CG9, CI6, CI7, CI9, CI10	RA1 - RA4, RA7	CE1 – CE4, CE7	PL2 (P)	10%
CG4-CG9, CI5, CI7, CI10	RA5, RA6, RA7	CE1 - CE3, CE6, CE7	PL3 (P)	10%
CG4-CG9, CI5-CI7, CI9, CI10	RA1 – RA7	CE1 – CE7	PEF	40%

Ordinary and extraordinary call – single exam evaluation

Competencies	Learning outcomes	Grading criteria	Grading instruments	Weight
CG4-CG9, CI5-CI7, CI9, CI10	RA1 – RA7	CE1 – CE7	PEF	70%
CG4-CG9, CI5-CI7, CI9, CI10	RA1 – RA7	CE1 – CE7	PL1-3 / PEF	30%

* The student may choose between keeping the grades obtained in the continuous evaluation tests (PL1, PL2 and PL3) or taking the PEF exam.

From the article 34.3 of the Learning Grading Regulations (Normativa de evaluación de los aprendizajes), approved in the Consejo de Gobierno (2016/05/05), about the originality of works and tests:

1. *Plagiarism is defined as the copy of texts without citing it source, passing them off as one's own, and it implies a grade of zero in the work where plagiarism was detected. Also this fact will be notified to the headmaster so he can proceed with the corresponding disciplinary measurements.*

6. BIBLIOGRAPHY

Basic bibliography:

- Sistemas Operativos. S. Sánchez Prieto. Segunda edición. Servicio de Publicaciones de la Universidad de Alcalá. 2005.

Complementary bibliography:

- Sistemas Operativos. William Stallings. Quinta edición. Prentice-Hall International.
- Unix. Programación avanzada. Francisco M. Márquez García. Tercera edición. Ra-Ma 2004.
- Fundamentos de Sistemas Operativos. A. Silberschatz, P. B. Galván y G. Gagne. Séptima edición. McGraw Hill. 2006.
- Sistemas Operativos Modernos. A. S. Tanenbaum. Tercera edición. Prentice Hall, 2009
- Linux: guía práctica. Sebastián Sánchez Prieto, Óscar García Población. Ed. RA-MA.