Universidad de Alcalá

TEACHING GUIDE

Advanced Operating Systems

Degree in
Computer Engineering (GIC)
Computer Science Engineering (GII)

Universidad de Alcalá

Academic Year 2026/2027

3rd Year - 1st Semester (GIC+GII)

Approved by the EPS School Board on June 15th, 2026

TEACHING GUIDE

Course Name:	**Advanced Operating Systems**
Code:	**780012 (GIC+GII)**
Degree in:	**Computer Engineering (GIC)** **Computer Science Engineering (GII)**
Department and area:	**Automática** **Computer Architecture and Technology**
Type:	**Compulsory (GIC+GII)**
ECTS Credits:	**6.0**
Year and semester:	**3rd Year - 1st Semester (GIC+GII)**
Teachers:	Por definir
Tutoring schedule:	Consultar al comienzo de la asignatura
Language:	English

1. COURSE SUMMARY

Advanced Operating Systems is a compulsory subject of 6 ECTS that is taught during the first four-month period of the third year in the Computer Engineering and Computer Science Engineering Degrees. This course represents a natural continuation of the knowledge acquired in “Operating Systems”, delving into key aspects of the inner workings of modern operating systems and laying the foundations for various professional profiles, such as system administrator, real-time systems specialist, embedded systems developer and architect of cloud-based solutions.

To successfully complete this course, it is essential to have previous knowledge acquired in subjects such as “Programming Fundamentals”, “Operating Systems” and “Computer Structure and Organization”. At the same time, the concepts developed in this Course will be the basis for other third and fourth year subjects, thus forming a coherent formative itinerary within the degree.

The course is structured in four fundamental blocks that deal with the most advanced components of an operating system:

Block 1: Memory Management - In this block we will explore the techniques and algorithms used by operating systems to efficiently manage the critical resource of memory. We will study the different allocation schemes, allocation and replacement policies, paging and segmentation techniques, as well as memory protection mechanisms that guarantee the integrity and security of running processes.

Block 2: Virtual Memory - We will delve into one of the most important concepts of modern operating systems: virtual memory. We will analyze how this technology allows the execution of programs that require more memory than is physically available. We will study page replacement algorithms, page table management and the impact of thrashing on system performance.

Block 3: Input and Output - This block focuses on the architecture and management of input/output subsystems. We will examine the principles of I/O hardware, device drivers, I/O programming techniques (polling, interrupts, DMA) and buffering mechanisms that optimize data transfer between devices and main memory.

Block 4: File Systems - In the last block we will study the organization, structure and operation of file systems. We will analyze the different implementations, from traditional to distributed file systems. Special emphasis will be placed on the study of classic systems such as FAT and the Unix inode system. We will cover aspects such as space allocation, metadata management, caching techniques, and failover mechanisms.

Through these four blocks, students will gain a thorough understanding of the internal components of a modern operating system, developing the skills necessary to manage, optimize, and troubleshoot complex computing environments.

Prerequisites and recommendations

To successfully take this subject, it is recommended to have previously taken the courses “Programming Fundamentals”, “Computer Structure and Organization” and “Operating Systems”. These subjects provide essential knowledge: “Programming Fundamentals” provides the necessary basis to understand the algorithmic and implementation aspects; “Computer Structure and Organization” allows understanding the hardware architecture on which the operating system operates; and “Operating Systems” introduces the basic concepts that will be expanded and deepened in this subject. In turn, the concepts developed in this Course will be the basis for other subjects of the third and fourth year, thus forming a coherent formative itinerary within the degree.

2. SKILLS

Basic, Generic and Cross Curricular Skills.

This course contributes to acquire the following basic, generic and cross curricular skills:

en_CG4 - Ability to define, evaluate and select hardware and software platforms for the development and execution of computer systems, services and applications, in accordance with the knowledge acquired as set out in section 5, annex 2, of resolution BOE-A-2009 -12977.

en_CG6 - Ability to conceive and develop centralized or distributed computer systems or architectures integrating hardware, software and networks in accordance with the knowledge acquired as set out in section 5, annex 2, of resolution BOEA-2009-12977.

en_CG8 - Knowledge of the basic subjects and technologies, which enable them to learn and develop new methods and technologies, as well as those that provide them with great versatility to adapt to new situations.

en_CG9 - Ability to solve problems with initiative, decision making, autonomy and creativity. Ability to know how to communicate and transmit the knowledge, skills and abilities of the profession of Computer Engineering Engineer.

en_CB1 - That students have demonstrated to possess and understand knowledge in an area of study that is based on general secondary education, and is usually found at a level that, although supported by advanced textbooks, also includes some aspects that involve knowledge from the forefront of their field of study.

en_CB2 - That the students know how to apply their knowledge to their work or vocation in a professional manner and possess the competencies that are usually demonstrated through the elaboration and defense of arguments and the resolution of problems within their area of study.

en_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.

en_CB4 - That students can transmit information, ideas, problems and solutions to both a specialized and non-specialized public.

en_CB5 - That the students have developed those learning skills necessary to undertake further studies with a high degree of autonomy.

en_TRU1 - Capacity of analysis and synthesis.

en_TRU2 - Oral and written competencies.

en_TRU3 - Ability to manage information.

en_TRU4 - Autonomous learning skills.

en_TRU5 - Team work.

Specific Skills

This course contributes to acquire the following specific skills:

en_CI5 - Knowledge, administration and maintenance of computer systems, services and applications.

en_CI6 - Knowledge and application of the basic algorithmic procedures of computer technologies to design solutions to problems, analyzing the suitability and complexity of the proposed algorithms.

en_CI7 - Knowledge, design and efficient use of the types and structures of data most appropriate to the resolution of a problem.

en_CI9 - Ability to know, understand and evaluate the structure and architecture of computers, as well as the basic components that make them up.

en_CI10 - Knowledge of the characteristics, functionalities and structure of the Operating Systems and to design and implement applications based on their services.

Learning Outcomes

After succeeding in this subject the students will be able to:

RA1. Develop programs using the API of an operating system.

RA2. Develop physical and virtual memory management systems.

RA3. Understand and apply memory management mechanisms.

RA4. Understand and apply memory management algorithms.

RA5. Apply input-output algorithms.

RA6. Analyze the organization of a file system.

RA7. Demonstrate awareness of engineering practice responsibility, social and environmental impact, and commitment to professional ethics, accountability, and standards of engineering practice.

3. CONTENTS

Contents Blocks	Total number of hours

Block I. Memory system management
• Principles of memory management.
  ◦ Computer memory hierarchy.
  ◦ Spatial and temporal locality.
  ◦ Fragmentation.
• Memory management of a process.
• Memory management mechanisms.
  ◦ Segmentation.
  ◦ Paging.
  ◦ Paged segmentation.
• Case studies.
	16 hours

Block II. Management of the memory system
• Introduction to virtual memory.
• Concepts
  ◦ Dynamic load.
  ◦ Pagers.
  ◦ Working sets of a process.
• Virtual memory management algorithms.
  ◦ Allocation policies.
  ◦ Location policies.
  ◦ Search policies.
  ◦ Replacement policies.
• Case studies.
	16h hours

Block III. Input-Output (I/O) Management • I/O software layers. • I/O elements • I/O projection in memory • I/O implementation techniques • Device example: disk management	6 hours

Block IV. File system
• File system functions.
• Disk partitioning and formatting.
• File system reliability.
• File system interface.
• Projection of files into memory.
• File system structure.
  ◦ Files.
  ◦ Directories.
  ◦ Usage from processes
• Case studies.
	18 hours

4. TEACHING - LEARNING METHODOLOGIES. FORMATIVE ACTIVITIES.

4.1. Credits Distribution

Number of on-site hours:	60 hours (56 hours on-site + 4 exams hours)
Number of hours of student work:	90
Total hours	150

4.2. Methodological strategies, teaching materials and resources

Distribución de créditos

Number of classroom hours: 58 hours	**Lecture**. Classes will be given to large groups, with the presentation in the classroom of the concepts of operating systems, making use of expository and participatory methodology, through which students are provided with essential and organized information from various sources with specific predefined objectives, with the use of audiovisual and other teaching resources. Its purpose is the transmission of the contents of operating systems, motivating students to reflect, facilitating the discovery of the relationships between various concepts and forming a critical mindset. Evaluation through online tests of the acquired skills. **Resolution of practical cases**. This teaching method is used as a complement to the theory class and is carried out in small groups. It is based on the development of solutions, exercise of routines and application of algorithms of the operating systems for the realization of software development projects; in addition to the application of transformation procedures of the available information and the interpretation of the results. The main intention is to apply what has already been learned in order to favor the understanding of a new subject, strengthen knowledge and strategies and their application in the practical situations that arise. Evaluation of solutions to the problems posed and functional tests through the development of projects.

Number of hours of the student's student's own work: 92 hours	**Self-study activities**. These are activities that the student will carry out individually, under the supervision or not of the professor, through face-to-face or virtual tutorials; so that the student can advance in the acquisition of knowledge and procedures of the subject. The main purpose is to develop the ability to self-regulate their learning by planning, designing and adapting it to their particular conditions. **Team work activities.** These are activities, guided and not guided by the teacher, in which a group of students solve a specific task or project. The main purpose is to favor in students the generation and exchange of ideas, the identification and analysis of different points of view, the generalization or transfer of knowledge and its critical evaluation.

Methodological strategies, materials and didactic resources

Methodological strategies	**Exposition**. It consists of the transmission of knowledge, offering a critical approach to the subject, leading students to reflect and discover the relationships between the various concepts in order to form a critical mentality in the way of facing problems and applying a methodology, involving the student in the teaching process. **Case studies.** It consists of the intensive analysis of one or several operative systems, or subsystem of these; with the purpose of knowing them, interpreting them, contrasting them, reflecting, completing knowledge and training in the possible methods of their implementation. **Problem solving**. The student develops competencies by proposing adequate solutions and applying the acquired skills and knowledge; through the exercise of routines, application of algorithms, transformation procedures of the available information and interpretation of results. **Project-based**. It consists of carrying out a project to solve a problem through the analysis, design and implementation of a subsystem.

Materials and didactic resources	**Bibliographical references**. For each activity a series of bibliographical references will be provided that can be consulted in the School's library or on the Internet. **Web resources**. In the web of the subject will be found the didactic resources for the follow-up of the face-to-face sessions, as well as the activities and projects to be carried out by the student in an autonomous way or in work teams. **Equipment.** • Classroom. • Laboratory. • Web portal of the University of Alcalá. • Computer room of the Polytechnic Building. • Library.

5. ASSESSMENT: procedures, evaluation and grading criteria

Preferably, students will be offered a continuous assessment model that has characteristics of formative assessment in a way that serves as feedback in the teaching-learning process.

5.1. PROCEDURES

The evaluation must be inspired by the criteria of continuous evaluation (Learning Assesment Guidelines, LAG, art 3). However, in compliance with the regulations of the University of Alcalá, an alternative process of final evaluation is made available to the student in accordance with the Learning Assesment Guidelines as indicated in Article 10, students will have a period of fifteen days from the start of the course to request in writing to the Director of the Polytechnic School their intention to take the non-continuous evaluation model adducing the reasons that they deem convenient. The evaluation of the learning process of all students who do not apply for it or are denied it will be done, by default, according to the continuous assessment model. The student has two calls to pass the subject, one ordinary and one extraordinary.

The evaluation is inspired by the criteria of continuous evaluation (Learning Evaluation Regulations, NEA, art 3). However, respecting the regulations of the University of Alcalá, an alternative process of final evaluation is made available to the student, according to the Regulations of Evaluation of Learning as indicated in Article 10, students will have a period of fifteen days from the beginning of the course to apply in writing to the Director of the Polytechnic School their intention to take the model of non-continuous evaluation giving the reasons they deem appropriate. The evaluation of the learning process of all students who do not apply or whose application is denied will be carried out, by default, according to the continuous evaluation model.

The student has two calls to pass the course, one ordinary and one extraordinary.

Ordinary call

Continuous assessment:

The continuous assessment evaluates the development of competencies throughout the learning

process of the subject by means of a series of formative tests distributed throughout the course, which allow the student to approach the subject progressively.

This form of evaluation guarantees early feedback in the student's learning process and allows professors, coordinators and other elements of the Quality Assurance System to make a global follow-up, with the possibility of acting in case indicators or specific situations make it advisable.

The evaluation of each part will be carried out at the end of its corresponding block.

Final assessment:

It will be requested in writing following the Regulations of Evaluation of Learning.

Extraordinary call

The procedure will be the same as the one described for the evaluation by means of final exam in the ordinary call.

5.2. EVALUATION

EVALUATION CRITERIA

The skills acquired by the student will be assessed considering the following evaluation criteria:

CE1. The student shows initiative when designing algorithms and developing software, using the API of the operating system in work teams.

CE2. The student demonstrates argumentation in the ideas and reflects on case studies.

CE3. The student fulfills the assigned tasks, making his/her own contributions with clarity and rigor.

SC4. The student applies different algorithms for managing the virtual memory of the processes.

SC5. The student has acquired knowledge about the management of the input-output devices of a computer.

SC6. The student demonstrates that he/she is able to design a UNIX file system.

SC7. The student is able to develop own production works, properly citing the sources when such works are based on third party material, according to the criteria of correct professional ethics in the practice of engineering.

GRADING TOOLS

The students' performance will be assessed by their work, knowledge and skills acquired and the improvement of their learning process. The continuous assessment instruments to be used will consist of the continuous assessment activities proposed by the professor for each of the topics.

The total of the proposed activities, their contents and timing, will be communicated to the student during the presentation class or at the end of the corresponding block of content. These activities include:

- **PEI**: Intermediate Evaluation Tests of test type.
- **PL**: Laboratory tests, of test type (T) and/or program development (P) that illustrate modules of the operating system and tests on these programs.
- **PEF**: Final Evaluation Test, consisting of a written problem-solving exam at the end of the teaching period, in which the use of books will be allowed.

GRADING CRITERIA

Ordinary call, continuous assessment

In the ordinary call, continuous assessment the relation between the competences, learning results, criteria and grading tools is the following.

Competence	Learning outcome	Grading criteria	Grading tool	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 call-final assessment and extraordinary call

In the ordinary call, final assessment and extraordinary call, the relationship between competencies, learning outcomes, evaluation criteria and instruments is as follows.

Competence	Learning outcome	Grading criteria	Grading tool	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 will be able to choose between keeping the grades of the evaluation tests taken during the continuous assessment (PL1, PL2 and PL3) or taking the PEF test.

Notifications regarding the contents and evaluation tests of the course will be made through the web page of the course.

As quoted in Article 34.3 of the Learning Assessment Regulations, regarding the originality of papers and tests:

"Plagiarism is understood as the copying of texts without citing their source and giving them as their own elaboration and will automatically lead to the grade of failure (0) in the work or tests in which it had been detected. The professor who notices signs of plagiarism in the work or evaluation tests submitted to them will report this fact to the dean or director of the center within a maximum period of two days, to proceed, if necessary, to bring it to the attention of the Rector in case it could constitute a disciplinary infraction or a crime.

A student who submits any of the parts or evaluation tests of the Course will be considered as

presented and the rest will be considered as not presented (NP).

The teaching-learning methodology and the evaluation process will be adjusted when necessary, with the guidance of the Unit for Attention to Diversity, to apply curricular adaptations for students with specific needs.

6. BIBLIOGRAPHY

6.1. Basic Bibliography

- Sistemas Operativos. S. Sánchez Prieto. Ed.: Servicio de Publicaciones de la Universidad de Alcalá

6.2. Additional Bibliography

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

Disclosure Note

During the evaluation tests, the guidelines set out in the Regulations establishing the Rules of Coexistence of the University of Alcalá must be followed, as well as the possible implications of the irregularities committed during said tests, including the consequences for committing academic fraud according to the Regulation of Disciplinary Regime of the Students of the University of Alcalá.