TEACHING GUIDE 2016/17

Centre 226 - Faculty of Informatics **Cycle** Indiferente

Plan GINFOR20 - Bachelor¿s Degree in Computer Engineering **Year** Second year

SUBJECT

26022 - Introduction to Operating Systems **ECTS Credits:** 6

DESCRIPTION & CONTEXTUALISATION OF THE SUBJECT

Introduction to Operating Systems is a compulsory subject in the degree of Informatics Engineering that is taught in the second year. Other subjects that complete the formation in Operating Systems are the following: Operating Systems, Systems and Networks Administration, and Operating System Design and Real Time. These subjects are offered in the Computer Engineering minor inside the Informatics Engineering degree.

Each of the previous courses raises learning operating systems from a different and complementary view. In particular, the Introduction to Operating Systems course focuses on the functional description of operating systems, via the system call interface, which is presented to the (system) programmer as a virtual machine that greatly hides the complexity of the underlying hardware. For its part, the course on Operating Systems focuses on the management policies of resources and in assessing performance; the course on System and Network Administration focuses on aspects of management, and finally the course on Design of Operating Systems and Real Time deals with the implementation of the system.

COMPETENCIES/LEARNING RESULTS FOR THE SUBJECT

After completing the course, the student should be able to identify the interfaces of an operating system, and in particular to develop utilities for the Linux operating system using the system call interface it provides. The course focuses on different types of operating systems (time-sharing, real time, etc.), on their fundamental concepts (files, access protection, processes, and communication mechanisms) and the functionality of their components (subsystems for process, memory, and input/output management).

Other important aspects that will be also considered during the course are the ability to find the information and tools needed to solve the proposed problems, as well as the ability to accurately describe the functionality of the utilities developed, so that they can be used by other users or developers.

Added to the subject specific competences, the students will also acquire the general competences C4, C5, C8 and C9 and the specific competences of the computing branch RI5, RI10 and RI14 as they appear in the following document: http://www.informatika.ehu.es/p248-content/eu/contenidos/informacion/indice_finformatica_titulacion/eu_titulaci/adjuntos/General%20competences%20of%20the%20degree.pdf

THEORETICAL/PRACTICAL CONTENT

1. Introduction
Definition of operating system as an interface for applications and as a manager of the resources of the computer. Historical perspective of operating systems. Types of systems according to their functionality (monoprogrammed, multiprogrammed, time-sharing, real-time, embedded and distributed). Operating system interfaces (for regular users, for system administrators, for applications, for developers). Current market of operating systems (proprietary systems, open systems, open source systems) and perspective. Practices: the Shell as user and administration interface.

2. System Call mechanism
On the basis of the elements provided by the hardware interface (address spaces, devices interfaces, interrupt mechanism), determination of the common basic services to set as resident code. Concept of operating system call as a way of accessing a resident service. Implementation of system calls via the interrupt mechanism and the CPU execution modes. Practices: specification in the C programming language of example routines for accessing services.

3. Input/Output and files
Concept of device independence and input/output redirection. System calls related to input/output. Operation modes on specific devices and on files. File system concept and its mechanisms of representation. Practices: redirections from the Shell, file system management from the Shell, programming of input/output example utilities.

4. User management and security
Definition of multiuser system. Types of users, access modes and protection domains. Basic mechanisms of access protection management in centralized systems. System calls related to access protection and security. Practices: management of access rights from the Shell.

5. Memory management
Program loading and placement in systems with one or several programs in memory. Physical and virtual addressing. Static and dynamic relocation. Reentrant code. Support for virtual memory systems. Dynamic-link libraries. System calls related to program loading and memory management. Practices: programming a basic program launcher (i.e., a mini-shell), monitoring of memory consumption (with reentrant programs).

6. Process management
Concept of execution flow and context. Multiprogrammed and multithreaded systems. Concept of process (Unix model), process states and transition graph. Context switching. Scheduling of processes and basic scheduling policies. System calls related to process management. Practices: launch of background processes from the Shell, process monitoring, and modification of the basic launcher to make it multiprogrammed.

7. Interprocess communication and synchronization
Concepts of concurrency, shared resource, race condition and exclusive access. Critical sections of code. Basic mechanisms of exclusive access to critical sections. Communication by message passing using mailboxes. System calls related to interprocess communication. Client-server model for resource management. Examples of resource managers (drivers). Practices: interprocess communication (using pipes) from the Shell and programmed in C, simple example of a server.

METHODS

The first half of the course will be taught according to the methodology of Problem Based Learning (PBL), which is based on the definition of a set of problems whose resolution allows the acquisition of the desired competences. Therefore, the formal division in class hours and laboratory sessions will not be necessarily followed in this part.

TYPES OF TEACHING

Type of teaching	M	S	GA	GL	GO	GCL	TA	TI	GCA
Classroom hours	40			20					
Hours of study outside the classroom	50			40					

Legend: M: Lecture S: Seminario GA: Pract.Class.Work GL: Pract.Lab work GO: Pract.computer wo
GCL: Clinical Practice TA: Workshop TI: Ind. workshop GCA: Field workshop

ASSESSMENT SYSTEMS

- Continuous assessment system
- Final assessment system

TOOLS USED & GRADING PERCENTAGES

- Extended written exam 60%
- Practical work (exercises, case studies & problems set) 40%

ORDINARY EXAM CALL: GUIDELINES & DECLINING TO SIT

The continuous assessment is chosen at the beginning of the semester, and the student can make the definitive decision (at the 60%-80% of the semester) after the teacher has supervised the students' performance. The student has to fill in a form where the percentage of the assessment and the mark obtained by the student at that time are stated. In case there is no confirmation of final registration for the continuous assessment it is assumed that the student gives up to it.

For the part of the course that is based on PBL it is envisaged a continuous assessment based on the following evaluation methods:
- Self-assessment and individual assessment questionnaires (60%)
- Notebook of the student with the code developed, specification and verification results (35%)
- Other: results of laboratory tests, crossed verification of programs, etc. (5%)

For the non PBL part and for those who do not follow the continuous assessment, there will be an exam that involves both an analysis and the development of code (70%). Besides, reports on work in the laboratories must be delivered (30%).

EXTRAORDINARY EXAM CALL: GUIDELINES & DECLINING TO SIT

There will be an exam that involves both an analysis and the development of code (70%). Besides, reports on work in the laboratories must be delivered (30%).

COMPULSORY MATERIALS

Linux operating system, manuals, tools and C programs that will be provided.

BIBLIOGRAPHY

Basic bibliography

C. Rodríguez, I. Alegria, J. González, A. Lafuente: Descripción Funcional de los Sistemas Operativos. Síntesis, 1994.

F.M. Márquez: UNIX. Programación Avanzada 3ª Edición. Rama, 2004.

Afzal: Introducción a UNIX. Un enfoque práctico. Prentice-Hall, 1997.

B.W. Kernighan, R. Pike: The Unix Programming Environment, Prentice-Hall, 1984.

A.S. Tanenbaum: Modern Operating Systems (3rd Edition), Prentice-Hall, 2008.

In-depth bibliography

M. Rochkind: Advanced Unix Programming, Addison-Wesley, 2004.

Silberschatz, P.B. Galvin, G. Gagne: Operating System Concepts (Eight edition), John Wiley & Sons, 2008.

W. Stallings: Sistemas Operativos (Quinta edición). Prentice-Hall, 2005.

Journals

Useful websites

Moodle page of the course

www.linux.org

www.gnu.org

REMARKS