270091 - SOAD - Operating Systems for Distributed Applications

Coordinating unit: 270 - FIB - Barcelona School of Informatics
Teaching unit: 701 - AC - Department of Computer Architecture
Academic year: 2017
Degree: BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2010). (Teaching unit Optional)
ECTS credits: 6
Teaching languages: Catalan, Spanish

Teaching staff

Coordinator: - Marisa Gil Gomez (marisa@ac.upc.edu)
Others: - Beatriz Otero Calviño (botero@ac.upc.edu)

Prior skills

The student must have the technical capabilities that give the subjects studied previously over a fluid level of English to read and understand technical documentation, and to communicate orally.
As for the technical skills could be summarized as:
- In terms of operating systems: Knowledge of the basics of an operating system along with application programming interface using the generic system calls given to all Operating Systems.
- In terms of Computer Structure: Knowledge of the main elements of a computer, connection of these elements together, internal representation of data and knowledge of machine language.
- In terms of programming: Ability to build complex programs and modular from scratch including definition of data types, pointers and references to code in assembler. Compilation and installation of executable files.

Requirements

- Prerequisite SO

Degree competences to which the subject contributes

Specific:

CES1.4. To develop, maintain and evaluate distributed services and applications with network support.

CES1.8. To develop, maintain and evaluate control and real-time systems.

CES2.2. To design adequate solutions in one or more application domains, using software engineering methods which integrate ethical, social, legal and economical aspects.

CES3.1. To develop multimedia services and applications.

CT2.4. To demonstrate knowledge and capacity to apply the needed tools for storage, processing and access to the information system, even if they are web-based systems.

CT6.3. To demonstrate knowledge about the characteristics, functionalities and structure of the Operating Systems allowing an adequate use, management and design, as well as the implementation of applications based on its services.

CT6.4. To demonstrate knowledge and capacity to apply the characteristics, functionalities and structure of the Distributed Systems and Computer and Internet Networks guaranteeing its use and management, as well as the design and implementation of application based on them.

CT7.1. To demonstrate knowledge about metrics of quality and be able to use them.

CT7.2. To evaluate hardware/software systems in function of a determined criteria of quality.
Learning objectives of the subject

1. Knowing how a real OS works, from the boot, through the dynamic management of resources, to shutdown the computer.

2. To know the implementation details of some of the basic components of a real OS: initialization code, memory management, I/O management, process management and network management.

3. Understanding the multithreaded programming, the requirements to correctly use shared memory access and the mutual exclusion mechanisms.

4. Be able to understand the impact on power and energy consumption of the execution of functions and/or applications considering the components used such as: data structures, algorithms, policies and mechanisms as well as communication and synchronization techniques based on blocking and waiting.

5. Knowing the existing basic legislation regarding data protection and intellectual property.

6. Understanding the impact in terms of efficiency and performance can be caused by different levels of management resources.

7. Understand different scheduling policies for general purpose systems such as Linux, and understand how the use of a specific scheduling policy can affect the performance application.

8. Understanding the technological aspects that impact on economic phenomena, social and environmental.

9. Being able to split the resources management into layers, when working with middlewares, libraries or virtual machines.

10. Understanding the basic allocation of resources that makes the OS for a program or application can be run and the impact regarding performance and energy savings.

Teaching methodology

The course is designed for practical work that students should develop in the lab sessions, resting on a theoretical basis. The theoretical basis is developed in classes organized and directed by the teacher and the active participation of students. On the basis of references and documentation that the teacher will provide the students will gain experience on resource management of distributed applications and/or based on data distribution. The practical aspect of the course will develop the teaching laboratories, by performing a series of supervised practice. The practical experiments will be limited initially to facilitate achievement of objectives and then apply all the knowledge necessary to optimize a particular situation or assess the impact of different solutions, implementations and policies. At the end of the course and in groups of 3-4 students will present a case study.

CT7.3. To determine the factors that affect negatively the security and reliability of a hardware/software system, and minimize its effects.

CT8.1. To identify current and emerging technologies and evaluate if they are applicable, to satisfy the users needs.

CT8.4. To elaborate the list of technical conditions for a computers installation fulfilling all the current standards and normative.

Generical:

G2. SUSTAINABILITY AND SOCIAL COMPROMISE: to know and understand the complexity of the economic and social phenomena typical of the welfare society. To be capable of analyse and evaluate the social and environmental impact.
**Study load**

| Total learning time: 150h | Hours large group: 15h 10.00% | Hours medium group: 30h 20.00% | Hours small group: 15h 10.00% | Guided activities: 6h 4.00% | Self study: 84h 56.00% |
# Content

## Introduction

**Degree competences to which the content contributes:**

**Description:**

OS and basic concepts definition. OS structure. User interface and resource management. Basic features in current OS. Other software for resource management: servers and middlewares. Distributed applications resource management: remote services, data and services concurrency. Heterogeneous and hybrid systems.

## System boot and communication interfaces

**Degree competences to which the content contributes:**

**Description:**


## Processes

**Degree competences to which the content contributes:**

**Description:**


## Memory

**Degree competences to which the content contributes:**

**Description:**


## I/O

**Degree competences to which the content contributes:**

**Description:**


## File Systems
### Degree competences to which the content contributes:

#### Description:

### Protection and security

#### Degree competences to which the content contributes:

#### Description:
Protection definition. Protection mechanisms. Authoring, access control and system integrity. Capabilities, virus, worms and malware.
## Planning of activities

| Introduction to OS code: data structures and services | **Hours:** 16h  
Theory classes: 2h  
Practical classes: 4h  
Laboratory classes: 0h  
Guided activities: 0h  
Self study: 10h |
|------------------------------------------------------|--------------------------------------------------|
| **Description:**  
While introducing the subject, there will be an interaction among the students to remember concepts already known from previous courses and to complete the basic nomenclature.  
Includes search for information and proposal guidelines to assess sustainability skill from the standard skill definition provided and the course syllabus  
**Specific objectives:**  
1, 2, 4, 5, 8 |

| Lab 1: OS communication mechanisms and first-class objects: processes | **Hours:** 14h  
Theory classes: 0h  
Practical classes: 0h  
Laboratory classes: 4h  
Guided activities: 0h  
Self study: 10h |
|---------------------------------------------------------------------|--------------------------------------------------|
| **Description:**  
From a tiny OS (ZeOS), students will work and complete the code needed to manage exceptions and interrupts, he/she will program a simple system call, completing the basic information that the OS needs from the programs running.  
**Specific objectives:**  
2, 4, 9 |

| Execution-unit information and resource management (processes, threads,..) | **Hours:** 16h  
Theory classes: 3h  
Practical classes: 6h  
Laboratory classes: 0h  
Guided activities: 0h  
Self study: 7h |
|------------------------------------------------------------------------|--------------------------------------------------|
| **Description:**  
See how the execution unit (process, thread, or others) that performs the work as well as the resources and policies assigned by the system can affect the data and resource allocation structures.  
**Specific objectives:**  
3, 4, 6, 7, 9 |
## Lab 2: Process scheduling policies and mechanisms

**Hours:** 14h  
Theory classes: 0h  
Practical classes: 0h  
Laboratory classes: 4h  
Guided activities: 0h  
Self study: 10h

**Description:**  
From a queue mechanism and an RR scheduling policy, different priority-based and preemption-based policies will be implemented. The student should evaluate which are the most appropriate policies based on the work to be done (interactive, calculation, synchronization,...)  

**Specific objectives:**  
2, 3, 6, 9

## Different layers memory management interaction: middlewares and virtual machines.

**Hours:** 15h  
Theory classes: 2h  
Practical classes: 4h  
Laboratory classes: 0h  
Guided activities: 0h  
Self study: 9h

**Description:**  
From the understanding of the memory use and allocation the system does to the running programs, learn the possible strategies on allocation, reserve and both static and dynamic mapping, also implicitly and explicitly through system calls. The student will work on the performance evaluation issues and the implications that using different resource management policies have. In addition, he/she will work on the implications of the middleware's own resources management.  

**Specific objectives:**  
1, 2, 5, 6, 9

## Workshop on scheduling policies

**Hours:** 8h  
Guided activities: 3h  
Self study: 5h

**Specific objectives:**  
1, 2, 4, 6, 8, 10

## Lab 3: Using tools to extract information about the system resources

**Hours:** 6h  
Theory classes: 0h  
Practical classes: 0h  
Laboratory classes: 2h  
Guided activities: 0h  
Self study: 4h
**Description:**
Know and use resource-related basic system tools and data structures to improve resource system use by the running applications.
There are also energy saving tools such as PowerTOP.

**Specific objectives:**
2, 6, 10

<table>
<thead>
<tr>
<th><strong>I/O</strong></th>
<th><strong>Hours:</strong> 13h</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Theory classes: 2h</td>
</tr>
<tr>
<td></td>
<td>Practical classes: 4h</td>
</tr>
<tr>
<td></td>
<td>Laboratory classes: 0h</td>
</tr>
<tr>
<td></td>
<td>Guided activities: 0h</td>
</tr>
<tr>
<td></td>
<td>Self study: 7h</td>
</tr>
</tbody>
</table>

**Description:**
From the I/O operation independence and uniformity, be able to delve into device virtualization (channel and port), abstraction level, different characteristics management such as sharing and distribution, ...

**Specific objectives:**
1, 2, 4, 9

<table>
<thead>
<tr>
<th><strong>File system</strong></th>
<th><strong>Hours:</strong> 18h</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Theory classes: 4h</td>
</tr>
<tr>
<td></td>
<td>Practical classes: 6h</td>
</tr>
<tr>
<td></td>
<td>Laboratory classes: 0h</td>
</tr>
<tr>
<td></td>
<td>Guided activities: 0h</td>
</tr>
<tr>
<td></td>
<td>Self study: 8h</td>
</tr>
</tbody>
</table>

**Description:**
Based on the knowledge already acquired in other subjects and depth in the theory classes, explore the basic file operations, as well as the sharing and replication of data in current situations, such as working with databases and/or distributed searches.

**Specific objectives:**
2, 4, 9, 10

<table>
<thead>
<tr>
<th><strong>Lab 4: I/O management in distributed applications: Message passing</strong></th>
<th><strong>Hours:</strong> 6h</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Theory classes: 0h</td>
</tr>
<tr>
<td></td>
<td>Practical classes: 2h</td>
</tr>
<tr>
<td></td>
<td>Laboratory classes: 4h</td>
</tr>
<tr>
<td></td>
<td>Guided activities: 0h</td>
</tr>
<tr>
<td></td>
<td>Self study: 0h</td>
</tr>
</tbody>
</table>
Description:
Characterize different data access according to the type of application. Know basic tools to assess and characterize the behavior and performance of programs.

From small benchmarks on different sizes of data and results, see the different behavior of distributed applications (Web-based searches, databases, ...).

Specific objectives:
2, 4, 5, 6, 9

Protection and security

| Hours: 17h |
|---|---|
| Theory classes: 2h |
| Practical classes: 4h |
| Laboratory classes: 1h |
| Guided activities: 0h |
| Self study: 10h |

Description:
From the information provided, privacy situations will be solved. Tools and data structures will be proposed in order to guarantee system security and the privacy of user objects. Conclusions will be extracted and, if necessary, the guidelines and advice on sustainability in this area will be reconsidered.

Specific objectives:
5, 8, 10

Final presentation

| Hours: 10h |
|---|---|
| Guided activities: 3h |
| Self study: 7h |

Specific objectives:
3, 4, 5, 9, 10

Qualification system

To evaluate the course takes into account four grades:
a. Theory, which can be obtained by continuous assessment (Cont. Ass.) or the final presentation (Final).
b. Note Lab (NL), which is the continuous assessment of the work of the lab.
c. Presentation (Final). Teams of 3-4 students will present at the end of course work a poster, paper or oral presentation to show the expertise acquired during the course.
b. Evaluation of generic skills: it will be 10% of the final mark: 5% assessed on the theory and 5% in the laboratory. Students will only be awarded a "Not Presented" final grade if they fail to attend all of the tests.
The final grade is calculated using the following formulae:
NT = MAX (Cont Ass., Final)
NL = 1/num.pract * [(NotaPr 1) + ... + (NotaPr n)]
NActa = (0.40 * NL + 0.40* NT + 0.20* Pres)
Bibliography

Basic:

