270063 - SO2 - Operating Systems II

Coordinating unit: 270 - FIB - Barcelona School of Informatics
Teaching unit: 701 - AC - Department of Computer Architecture
Academic year: 2018
Degree: BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2010). (Teaching unit Optional)
ECTS credits: 6  Teaching languages: Catalan, Spanish

Teaching staff

Coordinator: - Juan Jose Costa Prats (jcosta@ac.upc.edu)
Others: - Alex Pajuelo Gonzalez (mpajuelo@ac.upc.edu)
- Yolanda Becerra Fontal (yolandab@ac.upc.edu)

Prior skills

The student must have the technical capabilities that will confer the subjects studied previously together with a medium level of technical English to read and understand documentation.

The technical capabilities could be summarized as follow:

- Operating systems: Understanding the basics of an operating system along with the creation of applications using the generic system call interface explained during the Operating System course.
- In terms of computer architecture: Knowledge of the main elements of a computer, how these elements relate to each other, internal representation of data and knowledge of the assembler language.
- In terms of programming: Ability to code complex programs from scratch composed of several modules. Definition of data types, pointers and references, and assembler code. Compilation and linkage of executables.

Requirements

- Prerequisite SO

Degree competences to which the subject contributes

Specific:

CEC2.1. To analyse, evaluate, select and configure hardware platforms for the development and execution of computer applications and services.
CEC2.2. To program taking into account the hardware architecture, using assembly language as well as high-level programming languages.
CEC2.3. To develop and analyse software for systems based on microprocessors and its interfaces with users and other devices.
CEC2.4. To design and implement system and communications software.
CEC2.5. To design and implement operating systems.
CEC3.1. To analyse, evaluate and select the most adequate hardware and software platform to support embedded and real-time applications.
CEC3.2. To develop specific processors and embedded systems; to develop and optimize the software of these systems.
CEC4.1. To design, deploy, administrate and manage computer networks.
CEC4.2. To demonstrate comprehension, to apply and manage the guarantee and security of computer systems.
Learning objectives of the subject

1. Knowing how a real OS works, since the computer boots and initializes the operating system, through the dynamic management of resources, till the computer is shutdown.
2. Details of the implementation of some of the basic components of a real OS: initialization code, memory management code, input/output management code, processes management code and upload of executable files code.
3. Understand the mechanism of dynamic insertion of code in a real OS.
4. Be able to implement a device driver and insert it into a real OS.
5. Know the multithreaded programming, the problem of sharing memory and the mechanisms of mutual exclusion.
6. Know some risk factors in the security of an operating system.
7. Understand the implementation of various security and access policies.
8. Be able to implement some of the basic components of a real OS: initialization code, memory management code, input/output management code, processes management code and load executable files code.
9. Developing system software for a specific architecture such as Intel x86. Using C and assembler.
10. Select the most appropriate settings of an operating system depending on the platform to use.
11. Find, organize and summarize relevant and quality information on an unknown topic. Write documents with a critical thinking on the main ideas of this topic.
12. Describe the basic concepts of OS view of computer networks.
13. To have a proactive attitude to quality and continuous improvement
14. Be able to adapt to situations of time constraints and/or resources and/or lack of information.
# 270063 - SO2 - Operating Systems II

## Study load

<table>
<thead>
<tr>
<th>Total learning time: 150h</th>
<th>Theory classes: 30h 20.00%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Practical classes: 0h 0.00%</td>
</tr>
<tr>
<td></td>
<td>Laboratory classes: 30h 20.00%</td>
</tr>
<tr>
<td></td>
<td>Guided activities: 6h 4.00%</td>
</tr>
<tr>
<td></td>
<td>Self study: 84h 56.00%</td>
</tr>
</tbody>
</table>
## System boot

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

**Description:**
This chapter will explain all the actions carried out to start all the operating system services. We also describe and discuss possible implementations of this process in the current operating system.

## Mechanisms to enter the system

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

**Description:**
Describes and discusses the possible implementation of the mechanisms to enter the system along with its hardware support. Important aspects of these mechanisms and dependencies between the operating system and hardware will be detailed. We discuss the benefits of implementing virtualization techniques today. Presents and discusses the various implementations of operating systems today.

## Memory Management

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

**Description:**
This chapter covers the following topics: The logical address space of the process. Page-based memory systems and their hardware support. Design and implementation of virtual memory. Replacement algorithms for virtual memory. Implementation of shared memory. Different memory systems design and implementation in modern operating systems and their hardware dependencies.

## Process Management

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

**Description:**
This chapter covers the following topics: Detailed Implementation of the process control block. Implementation details for the process of creating processes. Implementation details for the process of finishing processes. Implementation details for the process of loading executables. Implementation details of context switching between processes. Detailed Description of the structures and algorithms for process scheduling. Implementation details of the processes scheduling routines. Description and discussion of various implementations of process scheduling in modern operating systems.

## Extending the system kernel

**Degree competences to which the content contributes:**
Description:
This chapter covers the following topics: Concept, design and implementation of kernel modules of the OS. Loading kernel modules at system initialization time and run time. Description of access to features implemented in kernel modules. Understanding the relationship between kernel modules and physical and logical devices.

Input / output Management and file systems

Degree competences to which the content contributes:
Description:
This chapter covers the following topics: Description and implementation of the OS part independent and dependent of the device. Implementation of a device descriptor. Description, features and implementation of the I/O structures and file systems. Description and implementation of the logical structure of file systems on disk. Communication mechanisms between processes via file systems. Communication mechanisms between processes across the network. Description and implementation of various file systems today.

Shared Memory

Degree competences to which the content contributes:
Description:
This chapter covers the following topics: Problems with shared memory between multiple threads of execution within a process. Race conditions. Areas of mutual exclusion. Description and implementation of mechanisms for mutual exclusion in accessing shared memory, in particular, test and set, mutex and semaphores. Description, discussion and implementation of mechanisms for mutual exclusion in modern operating systems.
### Planning of activities

<table>
<thead>
<tr>
<th>Activity</th>
<th>Hours</th>
<th>Description</th>
<th>Specific objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Become familiar with the workplace</strong></td>
<td>4h</td>
<td>Acquiring the knowledge necessary to develop the laboratory.</td>
<td>1, 2, 8, 9</td>
</tr>
<tr>
<td><strong>Interrupt Service Routine</strong></td>
<td>8h</td>
<td>Develop an ISR for the clock interrupt</td>
<td>1, 2, 8, 9, 13, 14</td>
</tr>
<tr>
<td><strong>Creating an entry point into the system</strong></td>
<td>7h</td>
<td>Creating a new entry point into the system to use system services</td>
<td>1, 2, 6, 8, 9, 13, 14</td>
</tr>
<tr>
<td><strong>Memory management</strong></td>
<td>13h</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Description:
It describes the main algorithms and implementation of virtual memory.

### Specific objectives:
1, 2, 6, 7, 8, 9

<table>
<thead>
<tr>
<th>Process Management</th>
<th>Hours: 20h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td></td>
</tr>
<tr>
<td>Develop the basic structures of process management, as well as features to identify, create and destroy processes</td>
<td></td>
</tr>
<tr>
<td>Specific objectives:</td>
<td>1, 2, 6, 8, 9, 10, 13, 14</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>First Control</th>
<th>Hours: 8h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td></td>
</tr>
<tr>
<td>First part of the course</td>
<td></td>
</tr>
<tr>
<td>Specific objectives:</td>
<td>1, 2, 6, 8, 9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>First Lab Exam</th>
<th>Hours: 8h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td></td>
</tr>
<tr>
<td>First Lab Exam</td>
<td></td>
</tr>
<tr>
<td>Specific objectives:</td>
<td>1, 2, 6, 7, 8, 9, 12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Process scheduling</th>
<th>Hours: 12h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td></td>
</tr>
<tr>
<td>Specific objectives:</td>
<td></td>
</tr>
<tr>
<td>Theory classes: 0h</td>
<td></td>
</tr>
<tr>
<td>Practical classes: 0h</td>
<td></td>
</tr>
<tr>
<td>Laboratory classes: 6h</td>
<td></td>
</tr>
<tr>
<td>Guided activities: 0h</td>
<td></td>
</tr>
<tr>
<td>Self study: 6h</td>
<td></td>
</tr>
</tbody>
</table>
## Description:
Implementing the change of context and a simple planner based on quantum

## Specific objectives:
2, 6, 8, 9, 13, 14

### Kernel extension

<table>
<thead>
<tr>
<th>Hours: 6h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory classes: 2h</td>
</tr>
<tr>
<td>Practical classes: 0h</td>
</tr>
<tr>
<td>Laboratory classes: 0h</td>
</tr>
<tr>
<td>Guided activities: 0h</td>
</tr>
<tr>
<td>Self study: 4h</td>
</tr>
</tbody>
</table>

**Description:**
We describe the mechanisms used to extend the kernel of an operating system

**Specific objectives:**
1, 2, 3, 4, 6, 9, 10

### Management I / O and file systems

<table>
<thead>
<tr>
<th>Hours: 15h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory classes: 5h</td>
</tr>
<tr>
<td>Practical classes: 0h</td>
</tr>
<tr>
<td>Laboratory classes: 4h</td>
</tr>
<tr>
<td>Guided activities: 0h</td>
</tr>
<tr>
<td>Self study: 6h</td>
</tr>
</tbody>
</table>

**Description:**
Develop some of the mechanisms of input / output file system of an operating system

**Specific objectives:**
1, 2, 6, 7, 8, 9, 10, 12, 13, 14

### Interprocess communication

<table>
<thead>
<tr>
<th>Hours: 10h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory classes: 2h</td>
</tr>
<tr>
<td>Practical classes: 0h</td>
</tr>
<tr>
<td>Laboratory classes: 2h</td>
</tr>
<tr>
<td>Guided activities: 0h</td>
</tr>
<tr>
<td>Self study: 6h</td>
</tr>
</tbody>
</table>

**Description:**
Implementation of semaphores

**Specific objectives:**
1, 2, 5, 6, 8, 9, 13, 14
## Network

**Hours:** 5h  
Theory classes: 1h  
Practical classes: 0h  
Laboratory classes: 0h  
Guided activities: 0h  
Self study: 4h

**Description:**  
Mount a small network with a specialized service

**Specific objectives:**  
6, 7, 10, 11, 12

### Second Control

**Hours:** 8h  
Guided activities: 2h  
Self study: 6h

**Description:**  
Second control

**Specific objectives:**  
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12

### Second lab exam

**Hours:** 8h  
Guided activities: 2h  
Self study: 6h

**Description:**  
Second test lab

**Specific objectives:**  
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12

### Final Theory Exam

**Hours:** 9h  
Guided activities: 3h  
Self study: 6h

**Description:**  
Final exam for the course

**Specific objectives:**  
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12

### Final lab exam

**Hours:** 9h  
Guided activities: 3h  
Self study: 6h


The final grade for the course consists of the grade of the core technical competences (CT), and the grade of generic competence (CTr) by the formula:

$$\text{Final Grade} = (\text{CT} + \text{CTr}) \times (\frac{10}{11})$$

Where the maximum score of CTr is 1.

The grade of the CT can be obtained by continuous assessment (CTc) or exceptionally by a final exam (CTf). Is calculated as:

$$\text{CT} = \max(\text{CTc}, \text{CTf})$$

Where grade CTc is composed of several evaluative acts: theory exams (T) and laboratory exams (L). The formula for calculating this grade is as follows:

$$\text{CTc} = 50\% \, T + 50\% \, L$$

To calculate T two assessments are used with the same weights:

$$T = 50\% \, T1 + 50\% \, T2$$

To calculate L, two assessments with the same weights are also used and a follow up grade (S):

$$L = 35\% \, L1 + 35\% \, L2 + 30\% \, S$$

The follow up grade (S) is assigned by the teacher as an evaluation of the right progress throughout the laboratory.

The CTf grade is calculated using a theory exam (T) and a laboratory exam (L). To accomplish this grade is mandatory to do both exams. This option is only available for those students that fail the continuous assessment. The formula is as follows:

$$\text{CTf} = 50\% \, T + 50\% \, L$$

The grade of generic competence (CTr) is obtained during the semester through various activities. Rating of this competence will have values A, B, C, D or NA: A corresponds to an excellent level, B to a desired level, C to a sufficient level, D to a failed level, and NA to not assessed.
Bibliography

Basic:


Complementary:

