270018 - AC - Computer Architecture

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 Compulsory)
BACHELOR'S DEGREE IN ENGINEERING PHYSICS (Syllabus 2011). (Teaching unit Optional)
ECTS credits: 6
Teaching languages: Catalan, Spanish

Teaching staff
Coordinator: Josep Llosa Espuny (josepll@ac.upc.edu)
Others: Angel Olivé Duran (angel@ac.upc.edu)
Antonio Juan Hormigo (antonioj@ac.upc.edu)
Carles Hernandez Luz (carles.hernandez@bsc.es)
Carlos Alvarez Martinez (calvarez@ac.upc.edu)
Fermin Sánchez Carracedo (fermin@ac.upc.edu)
Miquel Moretó Planas (mmoreto@ac.upc.edu)
Ramon Canal Corretger (rcanal@ac.upc.edu)
Teresa Monreal Arnal (teresa@ac.upc.edu)

Prior skills
Students are expected to have an understanding of statistics and probability, operating systems, digital circuits and computer organization.

Requirements
- Prerequisite EC
- Pre-Corequisite PE
- Prerequisite SO

Degree competences to which the subject contributes

Specific:
CT2.3. To design, develop, select and evaluate computer applications, systems and services and, at the same time, ensure its reliability, security and quality in function of ethical principles and the current legislation and normative.
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.
CT3.6. To demonstrate knowledge about the ethical dimension of the company: in general, the social and corporate responsibility and, concretely, the civil and professional responsibilities of the informatics engineer.
CT5.2. To know, design and use efficiently the most adequate data types and data structures to solve a problem.
CT6.2. To demonstrate knowledge, comprehension and capacity to evaluate the structure and architecture of computers, and the basic components that compound 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.
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.

General:

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.

Teaching methodology

Theory lectures interleaved with small problems. In the theory classes homework will be assigned to students for the next practice class.

Problem-solving classes are based on group activities. Using problems solved individually at home, students will work together in small groups to resolve the doubts that may have emerged. Because the methodology used in practice classes it is recommended that students do not enroll to courses that overlap with this one.

The laboratory classes support the theory. Students have the documentation available before each practice session. It is mandatory that students prepare the session beforehand (read the documentation, study the concepts used, etc.). It is also recommended, once the session ends, to review the concepts seen. Students have to prepare a preliminary work that will be delivered at the beginning of each session. The lab sessions are performed on-site and used to produce the lab grade, so it is essential that there is no overlap of the laboratory with any other course.

Learning objectives of the subject

1. Students should be able to translate routines and high-level code fragments to assembly of a real machine (IA32) and link routines in assembler with a high-level language (C) using the Linux Application Binary Interface
2. Students should be able to describe the internal structure and operation of the main components of the memory hierarchy and the techniques to improve their performance.
3. Students should be able to describe the operation and the main mechanisms for error detection and correction.
4. Students should be able to describe the structure and operation of data storage systems and evaluate their reliability.
5. Students should be able to describe the taxonomy of instruction sets (ISA) and the characteristics of the different paradigms (such as RISC-CISC).
6. Students should be able to describe the techniques used in computer design based on parallelism (such as pipelining, superscalar processors, VLIW processors, vector SIMD extensions, multithreading processors, multiprocessors and multicomputers) and their principles of operation.
7. Students should be able to evaluate the performance of code fragments and/or applications (both in assembler and high level) taking into account components such as: memory hierarchy, storage systems, instruction set architecture (ISA) and the main processor design techniques based on parallelism.
8. Students should be able to assess the impact on power and energy consumption of code fragments and/or applications (in both assembler and high level) taking into account components such as: memory hierarchy, storage systems, the design of the instruction set architecture (ISA) and the main processor design techniques based on parallelism.
9. Students should be able to apply simple optimizations to code fragments to improve their performance and/or power consumption taking into account: the memory hierarchy, storage systems, the design of the instruction set architecture (ISA) and the main processor design techniques based on parallelism.
270018 - AC - Computer Architecture

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: 15h 10.00%</td>
</tr>
<tr>
<td></td>
<td>Laboratory classes: 15h 10.00%</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>

Content

**Fundamentals of computer design and evaluation**

Degree competences to which the content contributes:

**High-level / assembler language interface**

Degree competences to which the content contributes:

**Memory Hierarchy**

Degree competences to which the content contributes:

**Storage Systems**

Degree competences to which the content contributes:

**Instruction Set Architecture Design**

Degree competences to which the content contributes:

**Pipelining and parallelism in computer design**

Degree competences to which the content contributes:
### Planning of activities

<table>
<thead>
<tr>
<th>Course</th>
<th>Hours</th>
<th>Specific objectives:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>C1</strong></td>
<td><strong>4h 30m</strong></td>
<td>1, 7, 8, 9</td>
</tr>
<tr>
<td><strong>Fundamentals of computer design and evaluation</strong></td>
<td><strong>10h</strong></td>
<td>7, 8</td>
</tr>
<tr>
<td><strong>C2</strong></td>
<td><strong>5h</strong></td>
<td>1, 2, 7, 8, 9</td>
</tr>
<tr>
<td><strong>High-level/ assembly language interface</strong></td>
<td><strong>32h</strong></td>
<td>1, 7, 8, 9</td>
</tr>
<tr>
<td><strong>C3</strong></td>
<td><strong>7h</strong></td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9</td>
</tr>
</tbody>
</table>

Guided activities: 1h 30m
Self study: 3h

Guided activities: 2h
Self study: 3h

Theory classes: 2h
Practical classes: 1h
Laboratory classes: 1h

Guided activities: 0h
Self study: 6h

Guided activities: 2h
Self study: 3h

Theory classes: 4h
Practical classes: 3h
Laboratory classes: 5h
Guided activities: 0h
Self study: 20h

Guided activities: 3h
Self study: 4h

Hours: 4h 30m
Hours: 10h
Hours: 5h
Hours: 32h
Hours: 7h

Hours: 1h 30m
Self study: 3h
Guided activities: 0h
Self study: 6h
Guided activities: 0h
Self study: 20h
Guided activities: 3h
Self study: 4h

Guided activities: 1h 30m
Self study: 3h
Guided activities: 0h
Self study: 6h
Guided activities: 2h
Self study: 3h
Guided activities: 0h
Self study: 20h
Guided activities: 3h
Self study: 4h
### Memory Hierarchy

**Specific objectives:**
2, 3, 7, 8, 9

**Hours:** 47h  
Theory classes: 10h  
Practical classes: 5h  
Laboratory classes: 4h  
Guided activities: 0h  
Self study: 28h

### Storage Systems

**Specific objectives:**
3, 4, 7, 8, 9

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

### Instruction set design

**Specific objectives:**
5, 7, 8, 9

**Hours:** 10h  
Theory classes: 2h  
Practical classes: 1h  
Laboratory classes: 1h  
Guided activities: 0h  
Self study: 6h

### Pipelining and parallelism in computer design

**Specific objectives:**
6, 7, 8, 9

**Hours:** 14h  
Theory classes: 4h  
Practical classes: 1h  
Laboratory classes: 1h  
Guided activities: 0h  
Self study: 8h
**Visita Supercomputador Marenostrum**

**Specific objectives:**
1. 2
2. 3
3. 4
4. 5
5. 6

**Hours:** 2h
- Theory classes: 2h
- Practical classes: 0h
- Laboratory classes: 0h
- Guided activities: 0h
- Self study: 0h

---

**Qualification system**

The course grade is based on the three tests (C1, C2 and C3) and the laboratory grade (L).

The final grade (NF) is calculated (with a single decimal and rounded to nearest even) as: 

\[ NF = 0.15 \times C1 + 0.25 \times C2 + 0.4 \times C3 + 0.2 \times L \]

Students may opt for up to 10% additional grade based on their participation and activity in the practice class.

**Bibliography**

**Basic:**


**Complementary:**
