30205 - Computer architecture and organisation I

Información del Plan Docente

Academic Year	2018/19
Subject	30205 - Computer architecture and organisation I
Faculty / School	110 - Escuela de Ingeniería y Arquitectura 326 - Escuela Universitaria Politécnica de Teruel
Degree	443 - Bachelor's Degree in Informatics Engineering 439 - Bachelor's Degree in Informatics Engineering
ECTS	6.0
Year	1
Semester	Second semester
Subject Type	Basic Education
Module	

1.General information

1.1.Aims of the course

1.2.Context and importance of this course in the degree

1.3.Recommendations to take this course

2.Learning goals

2.1.Competences

2.2.Learning goals

2.3.Importance of learning goals

3.Assessment (1st and 2nd call)

3.1.Assessment tasks (description of tasks, marking system and assessment criteria)

4.Methodology, learning tasks, syllabus and resources

4.1.Methodological overview

The course learning process is based on:

- Lectures
- Problem-based learning in small group sessions.
- Practical sessions in the laboratory.
- Tutored work at the end of the semester (evaluated independently).
- Voluntary seminars and nonacademic tutorials to complement training activities.
- Effective study of the concepts

30205 - Computer architecture and organisation I

4.2.Learning tasks

The learning activities that will help the student to achieve the objectives of the subject are the following ones:

- Development of the subject program in the lectures.
- In the lessons of problems, the concepts discussed in the lectures will be worked on detail.
- In the computer lab sessions small groups of students will work supervised by a teacher.
- Students can also attend personalized tutorial classes.

4.3.Syllabus

The program to be developed in this course is:

- **Architecture Processor:** Interpretation and translation, machine and assembly language, development environment, representation and coding of information, basic operations, registers and memory, addressing modes, instruction set, translation of data structures and control of high-level languages.
- **Subroutines:** Procedure calls, activation record. Case study: Integration of high-level language code with assembly code and library routines.
- **I/O subsystem:** Generic model device driver records. Basic methods of synchronization and transfer. Exceptions. Integration of peripherals in microcontrollers.

4.4.Course planning and calendar

Schedule sessions and presentation of works

The educational organization of the course is as follows.

- Lectures (2 hours per week)
- Problem classes (1 hour weekly)
- Computer lab sessions:
  - Escuela de Ingeniería y Arquitectura de Zaragoza: 2 hours every 2 weeks
  - Escuela Universitaria Politécnica de Teruel: 1 hour a week
- Tutorial and evaluation activities

The timetable of classes and practice sessions will be defined when the academic calendar of the University of Zaragoza is approved.

Student Work

To achieve the targets of this subject, students have to spend about 150 hours distributed as follows:

- 56 hours approximately, during the learning activities (lectures, problems and practical lab sessions)
- 51 hours of personal study (study of notes and texts, problems solving problems, preparation for classes and practices, and learning of the software development process)
- 40 hours of software development work
- 3 hours for the written final exam.

4.5.Bibliography and recommended resources

1.ARM Assembly Language: Fundamentals and Techniques, William Hohl. CRC press. 2009
2.ARM System-on-Chip Architecture (2nd ed.), Steve Furber. Addison-Wesley. 2000

30205 - Computer architecture and organisation I

3.ARM Architecture Reference Manual (2nd ed.), David Seal. Addison-Wesley. 2000
4.ARM System Developer's Guide: Designing and Optimizing System Software, Andrew N. Sloss, Dominic Symes, Chris Wright. Elsevier. 2004
5.Organización y Arquitectura de computadores(7ª ed.). W. Stallings. Prentice Hall. 2006