UNIVERSIDAD AUTÓNOMA DE MADRID

17830 - MICROPROCESSOR-BASED SYSTEMS

Information of the subject

Code - Course title: 17830 - MICROPROCESSOR-BASED SYSTEMS

Degree: 473 - Graduado/a en Ingeniería Informática
474 - Graduado/a en Ingeniería Informática y Matemáticas
722 - Graduado/a en Ingeniería Informática
734 - Graduado/a en Ingeniería Informática y Matemáticas (2019)

Faculty: 350 - Escuela Politécnica Superior

Academic year: 2020/21

1. Course details

1.1. Content area

Computer Science

1.2. Course nature

Compulsory

1.3. Course level

Grado (EQF/MECU 6)

1.4. Year of study

734 - Graduado/a en Ingeniería Informática y Matemáticas (2019): 3
474 - Graduado/a en Ingeniería Informática y Matemáticas: 3
722 - Graduado/a en Ingeniería Informática: 2
473 - Graduado/a en Ingeniería Informática: 2

1.5. Semester

Second semester

1.6. ECTS Credit allotment

6.0

1.7. Language of instruction

Español, English

1.8. Prerequisites

No prerequisites are needed to attend this course.

1.9. Recommendations

It is *highly recommended to have completed the course Computers Structure of the first year and second semester.*

1.10. Minimum attendance requirement

There are two itineraries: one with compulsory attendance and other without it. Students must choose one of them at the beginning of the course and meet the different evaluation requirements involved in every model published in this syllabus.

ITINERARY WITH COMPULSORY ATTENDANCE
Attendance is required at least 85%.

ITINERARY WITHOUT COMPULSORY ATTENDANCE
Attendance is highly recommended but not mandatory.

-

VERY IMPORTANT

By default, it supposed all the students choose an ITINERARY WITH COMPULSORY ATTENDANCE.

Before doing the ordinary final exam on May, every student who chooses an ITINERARY WITHOUT COMPULSORY ATTENDANCE must send an e-mail to his theory professor to communicate his decision. Evaluation will not be continuous for these students. Otherwise, the student will be considered like an ITINERARY WITH COMPULSORY ATTENDANCE student. The exchange between both itineraries will be not penalized. The final exam for both itineraries could be different.

1.11. Subject coordinator/s

Miguel Angel Garcia Garcia

https://autoservicio.uam.es/paginas-blancas/

1.12. Competences and learning outcomes

1.12.1. Competences

The competencies to be acquired with this subject are:

Basic:
B5: Knowledge of the structure, organization, operation and interconnection of computerized systems, the basics of programming, and its application for solving problems of engineering.

Common:
C9: Ability to learn, understand and evaluate the structure and architecture of computers, as well as the basic components that make them up.

Specific:
IC1: Ability to design and build digital systems, including computers, microprocessor-based systems and communications systems.

1.12.2. Learning outcomes

In this course the student will learn the low-level programming model of microprocessor based on digital systems. In particular, they will study and analyze the basic concepts related to the low-level programming resources and integration of programs developed as a mixture of mid-level languages (C language) and low-level (assembly). The different programming strategies of input/output subsystem resources will be analyzed, focusing on the programming of major device drivers. This is complemented with the study and analysis of the buses and input/ output computer interfaces. The Intel 80x86

microprocessor family will be used to get the particularization of all the theoretical concepts.

The course involves a practical part where the student becomes familiar with the development tools and debugging of programs written in low-level languages (assembly language) and middle level (C language), and settle the knowledge through the design and implementation of small programs in assembly language and C.

1.12.3. Course objectives

The objectives to be achieved with this subject are:

GENERAL OBJECTIVES	
G1	Design and write programs in 80x86 assembly language.
G2	Design and write programs using 80x86 interruptions.
G3	Design and write programs combining language assembly language and C (middle level).
G4	Using software resources provided by the BIOS and the OS.
G5	Design and write memory-resident programs (drivers).
G6	Programming basic PC hardware I/O resources.
G7	Use a low-level development and debugging environment.

UNIT BY UNIT SPECIFIC OBJECTIVES	
UNIT 1.- Microprocessor-based Digital Systems.	
1.1.	Describe the basic components of a microprocessor-based digital system.
1.2.	Describe the basic components of a microprocessor.
1.3.	Describe the basic operation of a microprocessor-based digital system.
UNIT 2.- Intel 80x86 Programming Model.	
2.1.	Describe the basic components of a 80x86 microprocessor.
2.2.	Describe the logical organization of memory in 80x86-based systems.
2.3.	Describe the addressing modes of 80x86 microprocessors.
2.4.	Design and write programs using 80x86 assembly language (instructions, directives, addressing modes).
2.5.	Describe the Memory Map of a 80x86-based system.
2.6.	Design and write programs using 80x86 interruptions.
2.7.	Use the low-level development and debugging environment.
UNIT 3.- Assembly and C language interface.	
3.1.	Describe the different memory models used by a classic C compiler.
3.2.	Describe the conventions, functions parameters process, results return and so on.
3.3.	Design and write programs combining assembly language (low-level) and C language (medium-level).
3.4.	Use the low-level development and debugging environment to write programs combining assembly and C languages.
UNIT 4.- Software resources in the 80x86 architecture.	
4.1.	Use the BIOS interruptions.
4.2.	Use the DOS interruptions.
4.3.	Describe and use the Prefix Segment Program structure (PSP).
4.4.	Design and write .EXE and .COM programs.
4.5.	Design and write memory-resident programs (*drivers*).
UNIT 5.- Input / Output in the 80x86 architecture.	
5.1.	Describe the programming basic techniques for using input/output

	resources.
5.2.	Describe the operation of the programmable hardware interruptions controller PIC (8259A).
5.3.	Describe and use the basic operation commands (OCW) of the 8259A controller.
5.4.	Describe the interconnections among I/O basic hardware resources and the interruptions controller 8259A.
UNIT 6.- Programming the PC basic hardware resources in the 80x86 architecture.	
6.1.	Describe and use the keyboard controller (8042).
6.2.	Describe and use the Timer (8253/54).
6.3.	Describe and use the Real-Time Clock RTC (MC146818).
6.4.	Describe and use the Video controller and the display.
6.5.	Describe and use the Parallel Port and Printer Port.

1.13. Course contents

Brief Course Contents

UNIT 1. Microprocessor-based Digital Systems.
UNIT 2. Intel 80x86 Programming Model.
UNIT 3. C and Assembler language Interface.
UNIT 4. Software resources.
UNIT 5. Input/Output.
UNIT 6. PC basic hardware resources programming.

-

Detailed Course Contents

1. **Microprocessor-based Digital Systems.**
   1.1. Microprocessor-based digital system basic architecture.
   1.2. Microprocessor basic architecture.
2. **How works a microprocessor-based system**.
3. **Intel 8x86 Programming Model.**
   3.1. 80x86 family as a particular case.
   3.2. Internal registers and 80x86 architecture.
   3.3. Memory access and organization.
   3.4. Addressing modes.
   3.5. Directives and operators of the 80x86 assembler.
   3.6. Assembly program structure.
   3.7. Assembly Instructions: data transfer, arithmetic and logic operations, control, interruptions, etc.
   3.8. PC Memory Map.
4. **Interruptions: mechanism and interruption vectors.**
5. **C and Assembler languages interface.**
   5.1. General characteristics.
   5.2. The C language as an example.
   5.3. Memory Models of C language.
   5.4. Conventions.
6. **Software resources.**
   6.1. BIOS interruptions.
   6.2. DOS interruptions.

6.3. Program execution from DOS.

6.4. PSP (Prefix Segment Program).

6.5. Program types: EXE, COM, y residents (TSR).

7. **Input / Output.**

7.1. Input/output programming techniques (I/O).

7.2. Pulling.

7.3. Interruptions.

7.4. DMA.

7.5. Management and programming of 80x86 interruptions: the 8259A programmable controller.

8. **PC hardware resources programming.**

8.1. Keyboard.

8.2. Timer.

8.3. Real-Time Clock (RTC).

8.4. Video controller and display.

8.5. Parallel Port. Printer.

1.14. Course bibliography

1. El universo digital del IBM PC, AT y PS/2. Ciriaco García de Celis, documentación gratuita en la red
2. Los microprocesadores Intel. Barry B. Brey, Ed. Prentice-Hall
3. IBM PC & XT, Assembly Language. Leo. J. Scalon, Ed. Brady
4. Arquitectura, programación y diseño de sistemas basados en microprocesadores (80x86/80186/80286). Yu-Cheng Liu y Glenn A. Gibson, Ed. Anaya
5. The 80x86 Family: Design, Programming, and Interfacing by John E. Uffenbeck
6. The X86 Microprocessors: Architecture and Programming (8086 to Pentium) **By:** Lyla B Das.
7. 80x86 assembly programming by Douglas Samuel Jones

Main and secondary bibliography in relation with the course proposed contents:

UNIT 1. Sistemas digitales basados en microprocesador.
Main: Ref[6] complete, Ref[7].
Secondary: Ref[1] complete.

UNIT 2. Modelo de programación del 80x86 de Intel.
Main: Ref[5] complete, Ref[8].
Secondary: Ref[3], Ref[4], Ref[6].

UNIT 3.Interfaz del ensamblador con el lenguaje C.
Main: Ref[1] complete.
Secondary: Ref[4].

UNIT 4. Recursos de programación.
Main: Ref[1] complete, Ref[8].
Secondary: Ref[4] complete.

UNIT 5. Entrada/Salida.
Main: Ref[1] complete, Ref[8].
Secondary: Ref[4] complete.

UNIT 6. Programación de los recursos hardware básicos del PC.
Main: Ref[1] complete, Ref[8].

Secondary: Ref[4] complete.

2. Teaching-and-learning methodologies and student workload

2.1. Contact hours

	#horas
Contact hours (minimum 33%) (57%)	86
Independent study time (43%)	64

2.2. List of training activities

Activity	# hours
Lectures	37
Seminars	
Practical sessions	
Clinical sessions	
Computer lab	
Laboratory	26
Work placement	
Supervised study	
Tutorials	3
Assessment activities	8
Other	

3. Evaluation procedures and weight of components in the final grade

3.1. Regular assessment

The both parts, theory and practice, are evaluated on 10 points.

The subject final rating is estimated using the theory and practice ratings, according the following equation:

Final Rating = 0.4*Practice + 0.6*Theory

To pass the subject is mandatory to obtain a rating bigger or equal to 5 points, both theory and practice. Otherwise, the final rating will be obtained using:

Rating = (0.4*Min(5,Practice) + 0.6*Min(5,Theory))

Students who do not desire to follow the ITINERARY WITH COMPULSORY ATTENDANCE must communicate by e-mail to their theory professor his intention, before the ordinary final exam.

1. For students who decide the ITINERARY WITH COMPULSORY ATTENDANCE, their ratings will be obtained with the following equation:

Theory rating = 0.6*Ordinary Final Exam + 0.15*First Knowledge Test + 0.25*Second Knowledge Test

The ordinary final exam will be a written test about the whole subject contents. The rating of knowledge tests could be obtained from written tests, raised problems and activities, or both. The written tests could include questions and problems to solve.

To pass the theory part, the student must deliver all the proposed exercises and activities, and must attend all the classes. The student could not attend to classroom a maximum number of 6 hours.

The final practice rating is the sum weighted sum of the ratings of the practices carried out during the

course.

To pass the practical part, students must attend all practical classes. With justified reasons, a student may miss up to two practice sessions (4 hours) and must submit reports. Otherwise, a student must perform a practice exam consisting of a practice of greater complexity than those carried out in the laboratory.

The rating of the practical part will consider the quality of the designs and the level of results. Also assess the validity of the results obtained in each of the sections that have been set for implementation in practice scripts.

- In the itinerary with compulsory attendance, the minimum number of exams that the student has to submit to receive a numerical rating two-thirds of the maximum number of exams. Otherwise, the student will receive "No rating".
- Practical and theory ratings are preserved (validate) only for the extraordinary in the same academic year.
- Students who choose to leave their itinerary with compulsory attendance, if they notified before May final examination writing an e-mail to their theory professor, will be evaluated as no compulsory attendance, i.e. outside continuous assessment. Any student can change the continuous assessment approach to evaluation does not continue without penalty. The final evaluation to be presented may be different.

2. For students who choose an itinerary without compulsory attendance, ratings will be obtained as follows:

a. The rating for the theory part is:
The Ordinary Final Exam rating (100%).

The final exam will be written and the contents will cover all the objectives to be achieved by students in the full course. This exam may include theoretical issues and problems.

b. The rating for the practice parte is:
The rating obtained on a single practice exam that will evaluate the concepts developed in the labs for students of compulsory attendance itinerary.

- The theory qualification is conserved (validates) only for the same academic year. The practice note is conserved (validated) for the extraordinary final exam in the same academic year, provided that the qualification obtained is equal to or greater than 7.0 points for the two calls the following year.

3.1.1. List of evaluation activities

Evaluatory activity	%
Final exam	0
Continuous assessment	100

3.2. Resit

- The theory qualification is conserved (validates) only for the same academic year. ç
- The practice note is conserved (validated) for the extraordinary final exam in the same academic year, provided that the qualification obtained is equal to or greater than 7.0 points for the two calls the following year.

3.2.1. List of evaluation activities

Evaluatory activity	%
Final exam	100

4. Proposed workplan

Week	Content	Classroom Hours	Personal Hours
1st	◦ **Subject introduction.** ◦ **U1.** Microprocessor-based digital systems. Chapters: 1.1, 1.2, 1.3	3	3 • Studying the proposed materials on U1.
2nd	◦ **U2**. 80x86 Programming Model. Chapters: 2.1, 2.2, 2.3 ◦ **P1.** 80x86 Development Suite tutorial.	3+2	3 • Studying the proposed materials and problems solving on chapters: 2.1 to 2.3
3rd	◦ **U2.** 80x86 Programming Model. Chapters: 2.4, 2.5 (I) ◦ **P2.** 80x86 Debugging Suite tutorial.	3+2	3 • Studying the proposed material and problems solving on chapters: 2.4 to 2.5 (I).
4th	◦ **U2.** 80x86 Programming Model. Chapters: 2.5 (II), 2.6 ◦ **P3.** Addressing Modes, Directives and Operators (I)	3+2	3 • Entrega P1 y P2. • Studying the proposed material and problems solving on chapters: 2.5 (II) to 2.6.
5th	◦ **U2.** 80x86 Programming Model. Chapters: 2.7 (I) ◦ **P4.** Directives/Operators (II) and software structure in assembler.	1+2	3 • Studying the proposed material and problems solving on chapters: 2.7 (I).
5th	• Tutorial (Chapters: 1.1 to 2.6)	1	
5th	• Knowledge test 1 (Chapters: 1.1 to 2.6)	1	
6th	◦ **U2.** 80x86 Programming Model. Chapters: 2.7 (II), 2.8, 2.9 ◦ **P5.** Simple software design using instructions from 2.7 (I)	3+2	3 • Deliver P3 y P4. • Studying the proposed material and problems solving on chapters: 2.7 (II) to 2.9.
7th	◦ **U3.** Assembler and C interface. Chapters: 3.1, 3.2, 3.3, 3.4 ◦ **P6.** Simple software design using instructions from 2.7 (II) and interruption vectors installation.	3+2	3 • Studying the proposed material and problems solving on chapters: 3.1 to 3.4.
8th	◦ **U4.** Software resources (I). Chapters: 4.1, 4.2, 4.3, 4.4 ◦ **P7.** Software design using C and 80x86 Assembler languages.	3+2	3 • Deliver P5 y P6. • Studying the proposed material and problems solving on chapters: 4.1 to 4.4.
9th	◦ **U4.** Software resources (II). Chapters: 4.5 ◦ **U5.** Input / Ouput (I). Chapters: 5.1, 5.2, 5.3, 5.4 ◦ **P8.** Software design using BIOS and DOS interruptions. PSP parameters.	1+2	3 • Studying the proposed material and problems solving on chapters: 4.5 to 5.4.
9th	• Tutorial (Chapters: 2.7 to 5.4)	1	

Week	Content	Classroom Hours	Personal Hours
9th	• Knowledge test 2 (Chapters: 2.7 to 5.4)	1	
10th	◦ **U5.** Input/Output (II). Chapter: 5.5 ◦ **P9.** Resident software design (*drivers*).	3+2	3 • Deliver P7 and P8. • Studying the proposed material and problems solving on chapters: 5.5.
11th	◦ **U6.** PC hardware resources programming (I). Chapters: 6.1, 6.2 ◦ **P10.** Software design examples using hardware interruptions through the 8259A controller (PIC).	3+2	3 • Studying the proposed material and problems solving on chapters: 6.1 to 6.2.
12th	◦ **U6.** PC hardware resources programming (II). Chapters: 6.3, 6.4 ◦ **P11.** Software design examples using the Keyboard controller and the Timer.	3+2	3 • Deliver P9 and P10. • Studying the proposed material and problems solving on chapters: 6.3 to 6.4.
13th	◦ **U6.** PC hardware resources programming (III). Chapters: 6.5 (I) ◦ **P12.** Software design examples using the RTC (Real-Time Clock) and Video Controller.	3+2	3 • Studying the proposed material and problems solving on chapters: 6.5 (I).
14th	◦ **U6.** PC hardware resources programming (IV). Chapter: 6.5 (II) ◦ **P13.** Small project using the developed programs in previous practices.	2+2	3 • Deliver P11 and P12. • Studying the proposed material and problems solving on chapters: 6.5. • Deliver P13 (optional to increase the rate).
14th	• Tutorial (Chapters: 5.5 to 6.5)	1	
	• Ordinary Final Exam	3	13 • Ordinary Final Exam preparation.
	• Extraordinary Final Exam	3	9 • Extraordinary Final Exam preparation.