Real time systems

Academic Year: (2020 / 2021) **Review date: 08-07-2020**

Department assigned to the subject: Computer Science and Engineering Department

Coordinating teacher: FERNANDEZ MUÑOZ, JAVIER

Type: Electives ECTS Credits : 6.0

Year : 4 Semester :

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

Operating Systems Design
Computer Architecture

OBJECTIVES

The goal of this course is introducing the student the main concepts related to the design and implementation of systems with time requirements, that is to say real-time systems.
In order to achieve this goal, the student has to acquire a set of general competences, knowledge, capacities, and attitudes,

Basic Competences:
CB1.- Students have demonstrated knowledge and understanding in a study area of the base of general secondary education, and is typically at a level that, whilst supported by advanced textbooks, includes some aspects that implies knowledge of the state of the art of their field of study.

Specific Competences for Computing Engineering Mention
CEIC2.- Ability to develop specific processors, embedded systems and software development/optimization of such systems.

Along with the following program outcomes: a, c, d, e, g, i, j, k

General/transversal competences:
- Analysis and synthesis capacities. (PO a)
- Abilities to organize and plan. (PO a)
- Problem solving abilities. (PO c)
- Teamwork. (PO d)
- Capacity to apply theoretical concepts. (PO a, c)

Specific competences:
- Cognitive (Knowledge) (PO a)
1. To understand the specific concepts and problems related to real¿time systems and the differentiated aspects with other computational systems.
2. To know the most important methods used to implement real-time systems, to know the way their software is organized, and to understand the principles and the deployment process.
3. To know scheduling aspects in real-time systems.
4. To know the functionality in real-time systems, their architecture and inner behavior.
5. To know some important tools (programming languages, and operating systems) suitable for developing real-time systems.
6. To know the existing alternatives for building microprocessor-based systems for embedded environments.
7. To know the techniques for developing against specific processors and for embedded systems.
8. To know the methods to analyze, evaluate and select the embedded and real-time hardware/software platforms.
- Procedimental /Instrumental (Know how) (PO e, g, j, k)
1. To design and to evaluate a real-time systems.
2. Capacity to analyze Cyclic real-time schedulers and rate monotonic priority-based real-time schedulers and to check the correctness of a real-time application.
3. To design and to implement real-time applications by using real-time operating systems.
4. Capacity to design and to build microprocessor-based systems for embedded environments and real-time systems.

5. Capacity to design and to implement systems using specific processors and embedded systems.
6. Capacity to analyze, to evaluate and to select the embedded and real-time hardware/software platforms.
7. To use tools (programming languages and operating systems) suitable for developing distributed real-time systems, checking their temporal correctness
- Attitudinal (PO c, i)
1. Capacity to generate new ideas (creativity)
2. Critical attitude towards the internal architecture of current real-time systems.
3. Concern for the quality of the real-time system.
4. Motivation to investigate for solutions to new problems related to the design of operating systems.
5. Learning capacity and achivement motivation.

DESCRIPTION OF CONTENTS: PROGRAMME

The descriptors (keywords) associated with this course are:
Real time, Real time Operating Systems, Cyclic scheduling and priority-based scheduling, concurrent multitasking, time-sharing access to resources, real-time applications, synchronization, fault-tolerance, deterministic behavior, design with general and specific microprocessors, hardware/software integration, embedded systems.

Course syllabus:
1. Introduction to real-time and embedded systems.
    - Definition of control systems, real-time systems and embedded systems.
    - Classification of real-time systems.
    - Formalization of temporal requirements.

2. Real-time and embedded systems.
    - Hardware architectures for embedded systems .
    - Software frameworks for real-time and embedded systems.
    - Real-time and embedded operating systems.
    - Design of embedded architectures using microprocessors.
    - Design of real-time and embedded systems.

3. Cyclic task scheduling.
    - Cyclic systems and multitasking systems.
    - Simple task model.
    - Design of cyclic task schedulers.

4. Priority-based task scheduling.
    - Design of Priority-based task schedulers.
    - Priority inheritance protocol.
    - Priority ceiling protocol.
    - POSIX extensions for Priority-based task scheduling.

5. Dynamic task scheduling and quality of service.
    - Soft real-time systems and multimedia systems.
    - Resources management and quality of service.
    - Dynamic resource scheduling.

6.- Certification and model-based development
    - Certification standards (Ej: ECSS, DO-178C)
    - Model-based development methodology.

7.- Control systems overview

LEARNING ACTIVITIES AND METHODOLOGY

Theoretical lectures: 1.5 ECTS. To achieve the specific cognitive competences of the course. They present the knowledge that students should acquire. In order to take advantage of these lectures, students will have the and the list of basic books used as bibliography. Thus, students could complete the information given in lectures and take a deeper vision those materials they are interesting in. (CB1, CEIC2 and POs a, j)
Practical lectures: 1.5 ECTS. To develop the specific instrumental competences and most of the general competences such as teamwork, capacity to apply theoretical concepts, planning and organizing, analysis and synthesis. Besides, to develop the specific attitudinal competences. During the practical lectures, a real-time application is designed and developed using cyclic task scheduling and priority task scheduling. Also some embedded environments are evaluated. They are developed by groups of students and on computer classrooms, with a teacher tutoring the classes (CB1, CEIC2 and POs a, c, d,

e ,g , i, j, k)
Guided academic activities (present teacher): 1 ECTS (CB1, CEIC2 and POs a, c, d, e, g, i, j, k)
- By solving exercises and case of studies in a participatory way.
It can include the study of examples of real-time operating systems, collaborative resolution of exercises, collaborative answers corrections, presentation of works, etc.
Student¿s work: 1.5 ECTS. (CB1, CEIC2 and POs a, i, j)
- Exercises and complementary text reading proposed by the teacher.
- Personal study.
Exercises and examination: 0.5 ECTS. It goal is to influence and complement the development of the cognitive specific capacities and procedural skills. (CB1, CEIC2 and POs a, c, e, g)

ASSESSMENT SYSTEM

The evaluation is focused on knowing the degree of fulfillment on the learning goals.
Because of that, it is valued all the student's works through the continue evaluation of all activities by exercises and exams, practical works, and other tutored academic activities with the following weights:

* Final exam: 40% (CB1, CEIC2 and POs a, c, e)
* Practical works: 40% (CB1, CEIC2 and POs a, b, c, d, e, g, i, k) (the practical work is mandatory)
* Marked theory exercises: 10% (CB1, CEIC2 and POs a, c, e) (the marked theory exercises are mandatory)
* Directed academic activities: 10% (CB1, CEIC2 and POs j, k)

Continuous evaluation is fulfilled if practical works are delivered and also those academic activities marked as mandatory.

Alternatively, for those students who decide not to join the previous system of continuous evaluation the final calification will be the 60% of the value of the final exam.

For the extrodinary evaluation the options are:
* With continuous evaluation: Final exam 40%, continuous evaluation 60%
* Without continuous evaluation: Final exam 100%
The second option will be chosen whenever it is better than the first option.

This course follows the University policy regarding the evaluation process.

If the copy between/among practices is detected, all the involved students (both copied and copiers) will lose the calification obtained by the continuous evaluation (the practical part is considered undelivered).
Even more, depending on the severity of the case, they shall open an administrative procedure.

% end-of-term-examination:	40
% of continuous assessment (assigments, laboratory, practicals…):	60

BASIC BIBLIOGRAPHY

- Alan Burns and Andy Wellings Real-Time Systems and Programming Languages: Ada, Real-Time Java and C/Real-Time POSIX (4th Edition) , Pearson Education , 2009