
Ano Letivo 2016-17

Unidade Curricular SISTEMAS DE TEMPO REAL

Cursos ENGENHARIA ELETRÓNICA E TELECOMUNICAÇÕES (Mestrado Integrado)

Unidade Orgânica Faculdade de Ciências e Tecnologia

Código da Unidade Curricular 14811114

Área Científica CIÊNCIA DE COMPUTADORES

Sigla

Línguas de Aprendizagem
Português

Modalidade de ensino
Presencial diurno

Docente Responsável Helder Aniceto Amadeu de Sousa Daniel

DOCENTE	TIPO DE AULA	TURMAS	TOTAL HORAS DE CONTACTO (*)
Helder Aniceto Amadeu de Sousa Daniel	PL; T	T1; PL1	30T; 30PL

* Para turmas lecionadas conjuntamente, apenas é contabilizada a carga horária de uma delas.

ANO	PERÍODO DE FUNCIONAMENTO*	HORAS DE CONTACTO	HORAS TOTAIS DE TRABALHO	ECTS
4º	S2	30T; 30PL	168	6

* A-Anual;S-Semestral;Q-Quadrimestral;T-Trimestral

Precedências

Sem precedências

Conhecimentos Prévios recomendados

Programação imperativa, programação orientada por objectos, sistemas digitais e sistemas Operativos

Objetivos de aprendizagem (conhecimentos, aptidões e competências)

Conhecer os princípios fundamentais dos sistemas de tempo real e embebidos e ambientes de desenvolvimento adequados.

Compreender o escalonamento em tempo real de tarefas periódicas e assíncronas.

Compreender os padrões de comunicação entre sistemas.

Aplicar métodos formais de verificação e desenvolvimento de sistemas tolerantes a falhas.

Projeto e desenvolvimento de sistemas embebidos e de tempo real.

Entender como é implementado em Linux o suporte para aplicações de tempo real.

Conteúdos programáticos

1. Introdução aos sistemas de tempo real e embebidos: definições, características e paradigmas
2. Linguagens para tempo real
3. Programação concorrente (genérica e sobre Xenomai)
4. Interrupções e Tempo
5. Escalonamento para tempo real
6. Comunicação para tempo real
7. Sistemas Embebidos
8. Métodos formais de verificação
9. Tolerância a faltas

Metodologias de ensino (avaliação incluída)

Nas aulas teóricas os conteúdos são apresentados e discutidos. Na componente prática são aplicados esses conhecimentos no desenvolvimento de mini-projectos, que serão objecto de avaliação.

A avaliação, em qualquer época (normal, recurso, especial para finalistas, melhoria de classificação e trabalhadores estudantes) é composta por duas componentes:

classificação final = 50% Exame + 50 % mini-projecto

(Aprovação se $\geq 9,5$ valores)

O mini-projecto é de realização obrigatória constituindo 50% da avaliação final em qualquer época, sendo os restantes 50% obtidos no exame da época correspondente

Na época de melhoria de classificação, poderá ser efectuada melhoria a uma ou a ambas as componentes da avaliação: Exame e/ou mini-projecto

Todos os alunos regularmente inscritos são admitidos ao exame de época normal e recurso.

Consulta para exame: 25 páginas A4 com qualquer conteúdo

Bibliografia principal

Laplante, P. (1997). *Real-Time Systems Design and Analysis*, 2nd Ed., IEEE Press

outras fontes de informação que poderão ser úteis:

Shaw, Alan C. (2001). *Real - Time systems and software*?, Wiley

Grehan, R., R. Moote, I. Cyliax (1998). *Real-Time Programming: A guide to 32-bit embedded development?*, Addison-Wesley

Douglass, Bruce Powel (1999). *Doing hard time: developing real-time systems with UML, objects, frameworks, and patterns?*, Addison-Wesley

Academic Year 2016-17

Course unit REAL TIME SYSTEMS

Courses ELECTRONIC ENGINEERING AND TELECOMMUNICATIONS (Integrated Masterçs)

Faculty / School Faculdade de Ciências e Tecnologia

Main Scientific Area CIÊNCIA DE COMPUTADORES

Acronym

Language of instruction Portuguese

Teaching/Learning modality *Presential*

Coordinating teacher Helder Aniceto Amadeu de Sousa Daniel

Teaching staff	Type	Classes	Hours (*)
Helder Aniceto Amadeu de Sousa Daniel	PL; T	T1; PL1	30T; 30PL

* For classes taught jointly, it is only accounted the workload of one.

Contact hours

T	TP	PL	TC	S	E	OT	O	Total
30	0	30	0	0	0	0	0	168

T - Theoretical; TP - Theoretical and practical ; PL - Practical and laboratorial; TC - Field Work; S - Seminar; E - Training; OT - Tutorial; O - Other

Pre-requisites

no pre-requisites

Prior knowledge and skills

Imperative programming, Object Oriented Programming, digital systems and operating systems

The students intended learning outcomes (knowledge, skills and competences)

Knowing the fundamental principles of real time systems and embedded systems and appropriate development environments.

Understand the scheduling in real time of periodic and asynchronous tasks.

Understand communication standards between systems.

Apply formal methods of verification and development of fault-tolerant systems.

Design and development of embedded and real time systems.

Understand real time application support in Linux.

Syllabus

1. Introduction to real time systems and embedded systems: definitions and paradigms
2. Languages for real time
3. Concurrent programming
4. Interrupts and time
5. Real time scheduling
6. Real time communication
7. Embedded systems
8. Formal methods of verification
9. Fault tolerant systems

Teaching methodologies (including evaluation)

The course contents are presented and discussed in the lectures. In labs the skills learned will be used in the development of mini-projects which will be object of assessment.

Assessment, at any season (regular, recourse, special for finalists, grade improvement, working students) has 2 components:

final grade = 50% written exam + 50% mini project

(Approval if ≥ 9.5 (rounded to 10), grade range: [0, 20] values)

Development of the mini project is mandatory in any season and has a weight of 50% in the final grade. The written exam weights the remaining 50%.

In the "grade improvement" season, one or both assessment components can be improved: Written exam and/or mini-project.

The written exams may include questions on skills needed to develop the mini-project.

All regularly enrolled students are admitted at the regular and recourse seasons.

25 written A4 pages, with any information, can be brought to the exam as consulting material.

Main Bibliography

Laplante, P. (1997). "Real-Time Systems Design and Analysis", 2nd Ed., IEEE Press

outras fontes de informação que poderão ser úteis:

Shaw, Alan C. (2001). "Real - Time systems and software?", Wiley

Grehan, R., R. Moote, I. Cyliax (1998). "Real-Time Programming: A guide to 32-bit embedded development?", Addison-Wesley

Douglass, Bruce Powel (1999). "Doing hard time: developing real-time systems with UML, objects, frameworks, and patterns?", Addison-Wesley