COURSE DETAILS

"REAL-TIME SYSTEMS AND INDUSTRIAL APPLICATIONS"

SSD ING-INF/05

DEGREE PROGRAMME: MASTER SCIENCE IN COMPUTER ENGINEERING

ACADEMIC YEAR 2022-2023.

GENERAL INFORMATION – TEACHER REFERENCES

TEACHER: MARCELLO CINQUE
PHONE: 0817683874
EMAIL: MARCELLO.CINQUE@UNINA.IT

GENERAL INFORMATION ABOUT THE COURSE

INTEGRATED COURSE (IF APPLICABLE):
MODULE (IF APPLICABLE):
CHANNEL (IF APPLICABLE):
YEAR OF THE DEGREE PROGRAMME (I, II, III): I OR II
SEMESTER (I, II): I
CFU: 6

REQUIRED PRELIMINARY COURSES (IF MENTIONED IN THE COURSE STRUCTURE "ORDINAMENTO")

No

PREREQUISITES (IF APPLICABLE)

Basic knowledge on operating systems and programming, acquired during the bachelor's degree.

LEARNING GOALS

The course provides students with advanced notions related to real-time systems and their application in different industrial areas, with particular reference to mission and safety-critical systems. It provides the necessary skills for designing and developing real-time mixed-criticality systems, using operating systems and virtualization platforms for real-time embedded systems, including hybrid high-performance hardware architectures. The course addresses both the recommendations imposed by certification standards in different industrial contexts, such as automotive, railways and avionics, and research initiatives on related themes, such as the Industrial Internet of Things and Industry 4.0.

EXPECTED LEARNING OUTCOMES (DUBLIN DESCRIPTORS)

Knowledge and understanding

The student needs to show to understand the theoretical problems at the base of real-time computing, to know the scheduling and resource management algorithms to address such problems. He/she has to be able to illustrate the virtualization principles and techniques that can be used to realize mixed-criticality systems while respecting isolation properties, to recognize the main industrial areas in which such systems are used, along with the constraints imposed by the normative.

Applying knowledge and understanding

The student needs to show to be able to apply the methodological instruments for the feasibility analysis and the dimensioning of real-time systems on single-core, multi-core and/or asymmetric/hybrid platforms; to be able to analyze the theoretical and practical problems concerning the use of operating systems and virtualization platforms in embedded and industrial environments, to be capable to design and implement complex real-time mixed-criticality systems, with reference to processes and technologies conformant to the main industrial standards.

COURSE CONTENT/SYLLABUS

Introduction to real-time systems. Fundamental properties. Hard and soft real-time. The safety concept, safety- critical systems. Scheduling of periodic tasks: cyclic executive, rate monotonic (RM), earliest deadline first (EDF). Aperiodic servers: polling server, deferrable server, sporadic server, total bandwidth server, hard constant bandwidth server. Resources management: non-preemptive protocol, highest locker priority, priority inheritance and priority ceiling. Feasibility tests based on upper bounds and on response time analysis.

Real-time kernel architectures. Real-time operating systems. Latency sources in computing platforms. Real-time executive. Monolithic kernels. Microkernels and L4 family. Dual kernels: RTAI and Xenomai examples. Preemptable kernels. The PREEMPT_RT patch for the Linux kernel.

Programming real-time tasks in Linux. The RT-POSIX standard. Programming cyclic periodic tasks. Fixed priority scheduling in Linux (SCHED_FIFO). Programming real-time applications on Xenomai. Real-time performance evaluation. Practical applications.

Real-time systems monitoring. Monitoring methodologies: embedded constraints and monitored constraints. Fault tolerance and timing failures management. Rule-based logging. Error propagation analysis.

Real-time multi-processing. Use of multiprocessing systems for real-time computing. Tasks models and theoretical limits. Symmetric and Asymmetric Multi-Processing. Partitioned scheduling of sporadic tasks with EDF and RM. Bin packing allocation. Global scheduling: the pfair algorithm, global EDF, Dhall effect, pfEDF, RM-light and RM-US. Feasibility bounds. Interference and isolation in multiprocessors due to memory hierarchies.

Mixed-criticality systems. The Vestal's model. Multi-criticality response time analysis. Audsley method. Hierarchical scheduling: open system architecture, fixed-priority hierarchical systems, hierarchical constant bandwidth server. The use of hybrid asymmetric platforms and multi-processor systems on chip for the realization of mixed-criticality systems. OpenAMP: RemoteProc, RPMsg and examples of use on the Zync Ultrascale+ platform.

Real-time virtualization. Introduction to virtualization: CPU, memory and I/O virtualization. Virtualization issues in real-time systems. Types of hypervisors. Examples of real-time hypervisors: XEN with null and RTDS schedulers, Jailhouse and practical examples of use. Real-time cloud and real-time containers: architectural alternatives and examples of implementation/use in dual-kernel environments.

Industrial applications. The automotive context: the ISO 26262 standard, ASIL and development process, coding rules and MISRA-C, the concept of SEooC, the OSEK standard, the AUTOSAR standard, the CAN protocol, operating systems and hypervisors used in automotive. The avionics context: the DO 178B standard, the ARINC 653 standard, IMA architecture and APEX interface, real-time networking in avionics, operating systems and hypervisors used in avionics. The railways context: the CENELEC EN 50128 standard. The Industrial Internet of Things: terminology, architecture, notes on standardization initiatives and communication protocols.

READINGS/BIBLIOGRAPHY

- Textbooks:
    - Giorgio Buttazzo: “Hard real-time computing systems: Predictable Scheduling Algorithms and Applications”, Third Edition, Springer, 2011.
    - S. Baruah, M. Bertogna, G. Buttazzo. “Multiprocessor Scheduling for Real- Time Systems”, Springer, 2015
- Lessons’ slides, reports and scientific articles available on the official course website.

TEACHING METHODS

Teacher will use a) lectures for approx. 75% of total hours, b) practical guided exercises in classroom to deepen the understanding and skills on real-time tasks programming and the use of hybrid virtualization architectures for about 20% of total hours, and c) seminars on specific themes, such as industrial standards, for about 5% of total hours.
Lectures and seminars are explained with the help of detailed slides, which will be provided to students through the official course website.

EXAMINATION/EVALUATION CRITERIA

a) Exam type:

Exam type	
written and oral	
only written	
only oral	X
project discussion	X
other	

The exam is articulated in a single oral session, which comprehend the presentation of a project assigned during the course and three questions on problems, algorithms and theoretical/technological solutions among the ones explained at the course. The project can be developed in teams of maximum three students.