

SEMB / SETR - 2019 / 2020

Lecture	Dates	Planning	weeks
1	10-fev	Course introduction. Objectives, organization, assessment. Introduction to embedded systems. Examples. Main requirements. Typical interfaces	1
2	11-fev	Hardware and software platforms for embedded systems. Cross platform development.	
3	17-fev	Memory models. Compiling, linking and debugging. Real-Time requirements, origin and characterization.	2
4	18-fev	Models of computation. Task and execution time abstraction. Temporal task models. Models for tracking the environment.	
		Carnival Break	
5	2-mar	Introduction to task scheduling. Offline scheduling techniques.	3
6	3-mar	Online task scheduling with fixed priorities. Rate-Monotonic Scheduling (RMS), Deadline-Monotonic Scheduling (DMS), Generalized Fixed Priorities Scheduling (FPS). Utilization-based schedulability tests. Response-time schedulability tests.	
7	9-mar	Online task scheduling with dynamic priorities. Earliest Deadline First scheduling (EDF). Differences to RMS. Utilization-based schedulability tests. Response-time and processor-demand tests.	4
8	10-mar	Sharing resources across tasks. Critical sections. Basic synchronization techniques: interrupt and preemption disabling. Priority inversion and blocking time. Synchronization techniques based on semaphores. Priority inheritance. Priority ceiling. Stack resource policy.	
9	16-mar	Handling aperiodic tasks. Concept of server. Examples of servers for FPS and EDF.	5
10	17-mar	Fault-tolerance. Conflict real-time versus reliability. Temporal redundancy. Redundancy management.	
11	23-mar	POSIX profiles for embedded real-time systems. Common interfaces of Real-Time Operating Systems.	6
12	24-mar	Project assignment. Guidelines for structuring the project execution	
17	30-mar	Seminar with students presentations (separate per lab class)	7
16	31-mar	exam	
	6-13 abr	Easter Break	
13	14-abr	The Arduino platform. Simple multi-tasking programming patterns, the cyclic executive pattern. Interrupt service routines as tasks. The cycle executive with periodic trigger using a timer interrupt.	8
14	20-abr	Executing independent periodic tasks. Non-preemptive and preemptive (single shot) kernels. Exercizing simple multi-tasking applications	9
18	21-abr	Project	
19	27-abr	Project	10
20	28-abr	Project	
	4-8 maio	Academic Week	
21	11-mai	Project	11
22	12-mai	Project	
23	18-mai	Project	12
24	19-mai	Project	
25	25-mai	Project	13
26	26-mai	Project	