CE 705: Real Time Systems – Fall 2017/18
Course Syllabus

| Contact Information | Course Goal | Course Description | Learning Outcomes | Course Format | Prerequisites | Textbook | Assessment | Exams and Assignments | Late Policy | Schedule |

Contact Information

Instructor: Dr. Salem Al-Agtash**,** alagtash@gju.edu.jo , Office C413

Course Hours: Sunday: 5:00 – 8:00 PM

Office Hours: Sunday: 3:00 – 5:00 PM or by appointment

Course Website: GJU Elearning

TOP

Course Goal

To provide understanding of the principles of real time systems, including: real-time kernel configuration and extension, main loop designs, multi-tasking, inter-task communication, cooperative and priority pre-emptive designs, hard real-time scheduling theory and design, latency, response, and performance.

TOP

Course Description

Introduction to real-time systems, real-time scheduling including: multiprocessor scheduling, real-time operating systems (kernels), real-time communication, real-time programming languages, reliability and fault-tolerance, and real-time system requirements and design methods; Design, analysis, and implementation of real-time kernel mechanisms and real-time applications using kernels such as Linux and programming languages such as C.

TOP

Learning Outcomes

Upon successful completion of this course, you will be able to:

- Understand the basic requirements of real-time systems and how these requirements can influence the design of real-time programming languages and real-time operating systems.
- Understand the implementation and analysis techniques which enable the requirements to be realized

TOP

Course Format

- Class meetings will be on Sunday – 17:00 – 20:00 Lecture hours.
- Every week you will be asked to spend some time to read on your own. You may supplement it with reading corresponding material from the textbook.
- You will be given practical assignment, covering basic topics of real time Linux operating system and would consist of programming tasks.
- It is strongly recommended that you spend enough time and effort to understand the lectures and do practical assignments on time.

TOP

Prerequisites and Technical Requirements

By topics: Computer architecture, Operating systems, Systems programming

TOP

Textbook

Required textbook:

- Liu, Jane W. S, Real-Time systems, Prentice Hall, 2000
- Phillop Laplante, Seppo Ovaska, Real Time Systems Design and Analysis, 4th Addition, John Wiley & Sons, 2012

Main reference:

- RTLinux Reference Manuals.

TOP

Assessment

Participation/ Project:	**10%**
Mid-term exam:	**25%**
Practical assignments:	**25%**
Final exam:	**40%**
Total	***100%***

TOP

Exams and Assignments

Exams: All exams (including the final exam) will be open book exams. The final exam will be comprehensive, covering material from the entire course. The final exam will be held during final exam week.

Missed Exams: If you have an excusable absence from an exam, your exam grade may be counted as a missed exam grade, until it is counted after a make up exam. Barring exceptional circumstances, you must contact me (by email) to explain your absence within 24 hours of a missed exam. Otherwise, the absence will be considered unexcused, and your grade for that exam will be 0.

Graded Exams: All exam papers will be graded and returned to students by the end of the week following the exam date.

Grading Correction: Please do not hesitate to contact me for any grading correction requests within the class week of receiving the grade. After that, your grade will not be adjusted. If you find any mistake in grading, please let me know. Your grade will not be lowered

TOP

Late Policy

All assignments must be turned in the class on the due dates for full credit. No assignments will be accepted after class on the due date

TOP

Schedule

Week	Topics	
Week 1	**Introduction**	
	- Real time systems and applications	
Week 2	**Overview**	
	- Advanced topics in C - Kernel, Device driver interface, System call interface - System calls (Fork, Thread, IPC) Practical assignment I: Installing Linux commands and administration Develop and run C programs with Fork, Threads, IPC, Semaphores Use Makefile	
Week 3	**Fundamentals of real time systems**	
	- Hardware and software development - Hard versus soft real time - Design challenges - Coding: Modules and module development and configuration	
Week 4	**Real time system elements**	
	- Hardware and software - Processor, memory, and peripheral interfacing - Sensors and actuator devices - A/D and D/A I/O interface - Microprocessor versus microcontroller Practical assignment II: Installing RT Linux Develop and run modules using insmod	

Week 5	**Real time operating systems**	
	- System services and applications - Commercial operating systems - VX Works, Windows CE, Linux	
Week 6	**Hard real-time scheduling theory and design**	
	- Tasks, Periodic, Non-periodic, - Event driven. Splitting application into non-real time and real-time components. - User-Interface, Graphics, Database, Networking needs that influence Application Design	
Week 7, 8	**Real-time kernel configuration and extension**	
	- Kernel configuration parameters, stacks, priority, Priority levels Practical assignment III Basics of RT Linux Task creation and deletion	
Week 9, 10	**Inter-task communication**	
	- Message Queues, - Mail-Box, - Shared Memory, - Semaphore, Mutex, - Real-time FIFOs Practical assignment IV RT Linux Timer RT Linux FIFO	
Week 11, 12	**Real Time Device Drivers**	
	- Hardware resources: I/O, Memory and IRQ, Timer, Serial port driver, Network - Real-time development tools; - Real time Languages; - Coding for real time software Practical assignment V RT Linux Shared Memory Semaphores and Mutex	
Week 13, 14	**Development and testing techniques**	
	- Case study – applications Requirements and specifications System design, including controllers Frequency response, State space representation Characteristic equation, poles, zeros, and root locus design Mobile Robot - Robotino Course project – 4 Weeks duration	
Week 15	**Review**	
Week 16	**Final Exam**	

TOP