

Subject Code: 1ET2030106	Subject Title: REAL TIME OPERATING SYSTEM (Elective – I)
Pre-requisite	Fundamentals of Operating Systems

Course Objective:

The course aims to make the student understand various types of scheduling in real-time environment along with designing fault tolerant systems. It also helps learning dynamic behavior of various real-time databases. The course also intends to make the student understand software and system design process for microcontroller based real time systems and applies their programming skills to develop real time systems

Teaching Scheme (Hours per week)				Evaluation Scheme (Marks)				
Lecture	Tutorial	Practical	Credit	Theory		Practical		Total
				University Assessment	Continuous Assessment	University Assessment	Continuous Assessment	
2	-	2	4	60	40	30	20	150

Subject Contents				
Sr. No	Topic	Total Hours	Weight (%)	
1	Introduction: Structures of Real-Time System, Issues in Real-Time Computing, Task Classes, Performance Measures for Real-Time Systems, Estimating Program Run Times	5	21	
2	Task Assignment and Scheduling: Classical Uniprocessor Scheduling algorithm, Cooperative scheduling, Preemptive scheduling, Uniprocessor Scheduling of IRIS Tasks, Task Assignment, Mode Changes, Fault Tolerant Scheduling	5	20	
3	Real-Time Databases: Basic Definitions, Real-Time Vs General-Purpose Databases, Main Memory Databases, Transaction Priorities, Transaction Aborts, Concurrency Control Issues, Disk Scheduling algorithm, A Two Phase Approach To Improve Predictability.	6	25	
4	Fault Tolerant Techniques: Fault Types, Fault Detection, Fault and error Containment, Redundancy, Data Diversity, Reversal Checks, Integrated Failure Handling, Obtaining Parameter Values, Reliability Models for Hardware Redundancy, Software Error models	5	21	
5	Real-Time Communication: Network Topologies, Protocols, Clocks Synchronization.	3	13	

Review Presentation: The student is expected to spend minimum 2 hours per week (as mentioned in the teaching and evaluation scheme) to refer at least two peer reviewed journal papers related to this domain/subject. The student is expected to identify issues/challenges and emerging trends in the domain/subject. Student is supposed to explore various video lectures (E.g. NPTEL) available in the domain/subject. Student is required to make a review-presentation on the work carried out for the same.

Recommended sites for journal papers are (1) dl.acm.org (2) springer.com (3) sciencedirect.com (4) elsevier.com (5) ieeexplore.ieee.org (6) scholar.google.co.in (7) scopus.com or others of similar repute.

Course Outcome:

After learning the course, the students would be able to

- Understand the structure of various types of Scheduling Real-Time Tasks
- Assess and design fault tolerant systems
- Understand the dynamic behavior of various Real-Time Databases.
- Understand software and system design process for microcontroller based real time systems, comprehend intricacies of real time databases, apply programming skills to develop real time systems
- Understand the Real-Time Communication.

List of References:

1. Krishna and Shin, "Real-Time Systems," Tata McGraw Hill. 1999.
2. Jane W. Liu, "Real-Time Systems" Pearson Education, 2001.
3. Rajib Mall, "Real-Time Systems: Theory and Practice," Pearson, 2008.
4. Doug Abbott, "Linux for Embedded and Real-Time Applications", Third Edition, Elsevier, 2013
5. Ivan Cibrario, Bertolotti and Gabriele Manduchi, "Real Time Embedded Systems", CRC press, 2012

E-Resources / Web Links:

1. www.nptel.ac.in/courses/106105036/
2. faculty.cs.tamu.edu/bettati/Courses/663/2013C/course_homepage.html

List of Experiments:

Note: The experiment list provided beneath is for reference only. The course teacher may change/formulate it as per his/her methodology and requirement.

1. Implement Cooperative scheduling.
Implement following Preemptive scheduling;
 - 2.1 Rate-monotonic scheduling
 - 2.2 Round-robin scheduling
 - 2.3 Fixed priority pre-emptive scheduling, an implementation of preemptive time slicing
2.
 - 2.4 Fixed-Priority Scheduling with Deferred Preemption
 - 2.5 Fixed-Priority Non-Preemptive Scheduling
 - 2.6 Critical section preemptive scheduling
 - 2.7 Static time scheduling
3. Design a Programs to simulate scheduling algorithms – EDF (Earliest Deadline First)
4. Design a Programs to simulate scheduling algorithms – LRT (Least Release Time)
5. Design a Programs to simulate scheduling algorithms – LST (Least Slack Time)