

Real Time Systems

Course Title: Real Time Systems

Course No: CSC471

Nature of the Course: Theory + Lab

Semester: VIII

Full Marks: 60+20+20

Pass Marks: 24+8 + 8

Credit Hrs: 3

Course Description:

This course familiarizes students with different concepts of real time systems mainly focusing on scheduling, access control, memory management, optimization, and real time communications.

Course Objective:

The main objective of this course is to provide core knowledge of different concepts of real time system which will enhance the student capacity in building real time systems.

Course Contents:

Unit 1: Introduction (4 Hrs.)

Definition, Hard, Soft and Firm Real Time System, Real Time Vs. Embedded System, Timing Constraints, Application of Real Time System, Brief Survey of Real Time Programming: Ada 95, C, C++, C#, Fortran, Java, Occam 2, Special Real-Time Languages

Unit 2: Reference Model of Real Time System (4 Hrs.)

Processor and Resources, Temporal Parameters of Real-Time Workload, Periodic and Aperiodic Task Model, Precedence Constraints and Data Dependency, Other Dependencies, Functional Parameters, Resource Parameters of Jobs and Parameters of Resources

Unit 3: Periodic Task Scheduling (7 Hrs.)

Clock Driven Scheduling – Definition, Notations and Assumption, Scheduler Concepts, General Scheduling Structure, Cyclic Executives. Priority Driven Scheduling - Notations and Assumption, Fixed Priority Verses Dynamic Priority, Fixed Priority Scheduling Algorithms (RM and DM) and their Schedulability Analysis, Concept of Schedulability Tests – Inexact and Exact Schedulability Tests for RM and DM, Optimality of the RM and DM Algorithms, Practical Factors.

Unit 4: Aperiodic Task Scheduling (7 Hrs.)

Aperiodic Task Scheduling: Assumption and Approaches, Server Based and Non-Server Based Fixed Priority Scheduling Algorithms: Polling Server, Deferrable Server, Simple Sporadic Server, Priority Exchange, Extended Priority Exchange, Slack Stealing. Introduction to Scheduling of Flexible Computations: Flexible Applications, Imprecise Computation Model and Firm Deadline Model. Introduction to Scheduling of Flexible Computations –Flexible Applications, Imprecise Computation Model and Firm Deadline Model.

Unit 5: Real-Time Memory Management (5 Hrs.)

Process Stack Management, Multiple-Stack Arrangements, Memory Management in the Task-Control-Block Model, Swapping, Overlays, Block or Page Management, Memory Locking, Working Sets, Real-Time Garbage Collection, Contiguous File Systems

Unit 6: Resources and Resource Access Control (5 Hrs.)

Assumptions on Resources and their Usage, Effects of Resources Contention and Resource Access Control, Non Preemptive Critical Sections, Basic Priority-Inheritance Protocol, Basic Priority-Ceiling Protocol, Stack-Based, Priority-Ceiling (Ceiling-Priority) Protocol, Use of Priority-Ceiling Protocol In Dynamic-Priority System, Preemption-Ceiling Protocol, Controlling Accesses to Multiple-Unit Resources, Controlling Concurrent Accesses to Data Objects

Unit 7: Performance Analysis and Optimization of Real-Time Systems (6 Hrs.)

Challenges in Analyzing Real-Time Systems, Performance Analysis: Analysis of Round-Robin Systems, Response-Time Analysis for Fixed-Period Systems, Response-Time Analysis: RMA Example, Analysis of Sporadic and Aperiodic Interrupt Systems, Performance Optimization: Compute at Slowest Cycle, Scaled Numbers, Binary Angular Measure, Optimizing Memory Usage; Analysis of Memory Requirements; Reducing Memory Utilization: Variable Selection, Memory Fragmentation

Unit 8: Real Time Communication (7 Hrs.)

Introduction, Model of Real-Time Communication, Real Time Traffic Model, Real Time Connections and Service Disciplines, Priority – Based Service Disciplines for Switched Network, Weighted Round-Robin Service Disciplines, Medium Access-Control Protocols of Broadcast Networks, Internet and Resource Reservation Protocols, Real-Time Protocol

Laboratory Work / Case Study:

The laboratory work should focus on implementation of concepts related to scheduling, memory management, synchronization and optimization using suitable simulators and programming languages. There should also be a case study in group with at most 4 students focusing on any real time system implemented system.

Text Books:

1. Real-Time Systems, Jane W. S. Liu, Pearson Education Asia, Latest Edition
2. Real-Time Systems, Design Principles for Distributed Embedded Applications Kopetz, Hermann, Springer Latest Edition