PEC-CS601D: Parallel & Distributed Systems

[Course Outcomes]

1. Remember the basics of formal and real distributed systems
2. Understand the the design issues and how to design a new distributed system
3. Apply knowledge to solve the design issues related to the distributed systems
4. Analyze specific distributed systems e.g. distributed database system (DDBS), Kerberos
5. Evaluate the specific challenges of DDBS while getting a formal notion of other types of distributed systems
6. Create new distributed systems and protocols

[Module 1] [6L]

1. Introduction to Distributed Systems in general [2L] [CO 1,2]

   System Models, Nodes, Events, Messages, Channels, Local and Global States. Algorithm writing techniques for distributed systems, Basics of complexity in distributed systems.
2. Clocks and Event Ordering [4L] [CO 2,3,5]

   Global and Logical Clocks; Berkeley Algorithm, Network Time Protocol (NTP); Lamport's Clock, Vector Clock.

[Module 2] [11L]

3. Leader Election [6L] [CO 2,3,5,6]

   Definition and Requirement; Hirschberg-Sinclair Algorithm; Spanning Tree Construction; DFS tree construction
4. Global Snapshots [3L] [CO 2,3,5]

   Definition and Requirement; Chandy-Lamport's Algorithm

[Module 3] [13L]

5. Distributed Database Systems [8L] [CO 1,2,3,4,5]

   Overview; Distributed Database Design,Semantics Data Control, Processing Issues;
6. Distributed Query Optimization, Transaction Management, Concurrency Control; [7L] [CO 2,3,4,5,6]

[Module 4] [12L]

7. Mutual Exclusion [6L] [CO 2,3,5]

   Definition and Requirement; Permission based algorithm: Lamport's Algorithm, Ricart-Agrawala Algorithm, Maekawa's Algorithm; Token based algorithm:Suzuki Kasami Algorithm
8. Distributed File Systems [6L] [CO 2,3,4,5,6]

   Definition and Requirement; DFS Architecture, Design Issues; Caching; Network File System (NFS)
9. Basic concepts of parallelism, Complexity calculation, Prefix sum problem [2L] [CO 1,2, 3,4,5,6]

[Advanced Topics] [6L]

1. Replication [2L] [CO 1,2,3]

   Definition and Requirement; Design Issues in Replication; Consistency models: Sequential, Eventual, Casual;
2. Case study: Kerberos [2L] [CO 3,4,5]
3. Basics of Fault Tolerance [2L] [CO 1,2,3]

   Definition and Requirement; Types of faults, Types of tolerance; Agreement Problems, Atomic Commits: 2-phase, 3-phase commits; Checkpointing and Recovery

[Books/ References]

1. Advanced Operating Systems by Mukesh Singhal and Nirajan Shivaratri
2. Distributed Systems: Principles and Paradigms by Andrew Tanenbaum and Maarten van Steen
3. Distributed Systems - An Algorithmic approach by Sukumar Ghosh
4. Design and Analysis of Distributed Algorithms by Nicola Santoro
5. Distributed Database Systems by D.Bell and J. Grimson, Addison