**Course Objective:**  
The objective of the course is to be familiar with different aspect of the distributed system, middleware, system level support and different issues in designing distributed algorithms.

Distributed System is often described as a collection of logically related data that is distributed over different processing nodes of computer network. It’s collection of independent computers that appear to the users of the system as a single computer. A distributed system is one in which hardware or software components located at networked computers communicate and coordinate their actions only by message passing. Design Challenges of Distributed Systems:

* Heterogeneity
* Openness
* Security
* Scalability
* Failure handling
* Transparency

**1. Introduction (4 hours)**  
1.1 Introduction to Distributed Systems  
1.2 Examples of Distributed Systems  
1.3 Main Characteristics  
1.4 Advantages and Disadvantages of Distributed System  
1.5 Design Goals  
1.6 Main Problems  
1.7 Models of Distributed System  
1.8 Resource Sharing and the Web Challenges  
1.9 Types of Distributed System: Grid, Cluster, Cloud

**2. Distributed Objects and File System (7 hours)**  
2.1 Introduction  
2.2 Communication between distributed objects  
2.3 Remote Procedure Call  
2.4 Events And Notifications  
2.5 Java RMI Case Study  
2.6 Introduction to DFS  
2.7 File Service Architecture  
2.8 Sun Network File System  
2.9 Introduction to Name Services  
2.10 Name Services and DNS  
2.11 Directory and Discovery Services  
2.12 Comparison of Different Distributed File Systems

A Distributed File System ( DFS ) is a classical model of a file system distributed across multiple machines. The purpose is to promote sharing of dispersed files. The resources on a particular machine are local to itself. Resources on other machines are remote. A file system provides a service for clients. The server interface is the normal set of file operations: create, read, etc. on files.

Distributed file systems support the sharing of information in the form of files throughout the intranet.

**3. Operating System Support (3 hours)**  
3.1 The operating system layer  
3.2 Protection  
3.3 Process and threads  
3.4 Communication and invocation  
3.5 Operating system architecture

**4. Distributed Heterogeneous Applications and CORBA (3 hours)**  
4.1 Heterogeneity in Distributed Systems  
4.2 Middleware  
4.3 Objects in Distributed Systems  
4.4 The CORBA approach  
4.5 CORBA services

**5. Time and State in Distributed Systems (5 hours)**  
5.1 Time in Distributed Systems,  
5.1.1 Physical Clocks  
5.1.2 Logical Clocks  
5.1.3 Vector Clocks  
5.1.4 Clock Synchronization  
5.2 Causal Ordering of Messages  
5.3 Global State and State Recording  
5.4 Distributed debugging

**6. Coordination and Agreement (4 hours)**  
6.1 Mutual Exclusion in Distributed Systems  
6.2 Algorithms for Mutual Exclusion  
6.3 Distributed Elections  
6.4 Multicast communication  
6.5 Consensus

**7. Replication (4 hours)**  
7.1 Reasons for Replication  
7.2 Object Replication  
7.3 Replication as Scaling Technique  
7.4 Fault Tolerant Services  
7.5 High Available Services  
7.6 Transaction with Replicated Data

**8. Transaction and Concurrency Control (6 hours)**  
8.1 Transactions  
8.2 Nested Transaction  
8.3 Locks  
8.4 Optimistic Concurrency Control  
8.5 Timestamp Ordering  
8.6 Comparison of Methods For Concurrency Control  
8.7 Introduction to Distributed Transactions  
8.8 Flat and Nested Distributed Transactions  
8.9 Atomic Commit Protocols  
8.10 Concurrency Control in Distributed Transactions  
8.11 Distributed Deadlocks  
8.12 Transaction Recovery

**9. Fault Tolerance (4 hours)**  
9.1 Introduction to Fault Tolerance  
9.2 Process Resilience  
9.3 Reliable Client Server Communication  
9.4 Distributed Commit  
9.5 Recovery

**10. Case Studies (5 hours)**  
10.1 CORBA  
10.2 Mach  
10.3 JINI  
10.4 TIB/Rendezvous

**Practical:**  
1. Implementation of Election Algorithm.  
2. Simulation for Clock Synchronization in Distributed System using Lamport’s Algorithm.  
3. Implementation of Banker’s Algorithm for avoiding Deadlock  
4. Experiment on DFS  
5. Case Study – CORBA,JINI, Mach, TIB/Rendezvous

**Reference:**  
1. George Coulouris, Jean Dollimore, Tim Kindberg, “Distributed Systems Concepts and Design”, Third Edition, Pearson Education.  
2. A.S. Tanenbaum, M. VanSteen, “Distributed Systems”, Pearson Education.  
3. Mukesh Singhal, “Advanced Concepts in Operating Systems”, McGraw-Hill Series in Computer Science.

**Evaluation Scheme:**  
The questions will cover all the chapters of the syllabus. The evaluation scheme will be as indicated in the table below.

|  |  |  |
| --- | --- | --- |
| **Chapters** | **Hours** | **Marks Distribution\*** |
| 1 | 4 | 8 |
| 2 | 7 | 16 |
| 5 | 5 | 8 |
| 6 | 4 | 8 |
| 7 | 4 | 8 |
| 8 | 6 | 8 |
| 9 | 4 | 8 |
| 3,4,10 | 11 | 16 |
| **Total** | **45** | **80** |

**\*Note: There may be minor deviation in marks distribution**