**ROHINI COLLEGE OF ENGINEERING & TECHNOLOGY**

**Department of Computer Science and Engineering**

**COURSE DELIVERY PLAN**

Academic year : 2019-2020

Year/Sem/ Sec : III/VI

Course Code : CS8603

Course Name : DISTRIBUTED SYSTEMS

Regulation : 2017

Course Category Code : Core L: 3 T: 0 P: 0

Course Credit : 3

Course Faculty : MEENAKSHIAMMAL.R

Course Coordinator : MEENAKSHIAMMAL.R

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **PERIOD** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
|  |
| **DAY** |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **Day I** |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | CS8603 |
| **Day II** |  |  |  |  |  |  |  |
|  | CS8603 |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **Day III** |  |  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  | CS8603 |  |  |  |  |  |
| **Day IV** |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **Day V** |  |  |  |  |  | CS8603 |  |
|  |  |  |  |  |  |
|  |  |  | CS8603 |  |  |  |  |
| **Day VI** |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**COURSE TIME TABLE**

**Tentative dates for Internal Assessment**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date**  **IAT** | **Internal Assessment**  **Test I (IAT I)** | **Internal Assessment**  **Test II (IAT II)** | **Internal Assessment**  **Test III (IAT III)** |
|  |  |  |

**COURSE OBJECTIVES**

1. Understand foundations of Distributed Systems.
2. To understand in detail network virtualization and remote invocations required for distributed system.
3. Introduce the idea of peer to peer services and file system.
4. To understand clock synchronization techniques, transactions and concurrency control mechanisms.
5. Understand in detail the system level and support required for distributed system.
6. Understand the issues involved in studying process and resource management.

**PRE – REQUISITECHART**

DISTRIBUTED SYSTEMS (SEM: VI)

OPERATING SYSTEMS (SEM: IV)

DATABASE MANAGEMENTSYSTEMS (SEM: III)

**MODE OF DELIVERY and ASSESSMENT COMPONENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| MD 1. | Oral presentation | MD4 | Hands on/Demonstration |
| MD 2. | Tutorial | MD5 | Seminar/Guest lecture |
| MD 3. | Videos/PPT |  |  |

**COURSE OUTCOMES:**

**[Cognitive Level K1- remember, K2- Understand, K3- Apply, K4 - Analyze, K5-Evaluate, K6- Synthesize]**

**After successful completion of the course, the students should be able to**

|  |  |  |
| --- | --- | --- |
| **CO No.** | **Course Outcomes** | **Highest Cognitive Level** |
| 17C314.1 | Identify the core concepts of distributed systems, design issues and challenges. | K3 |
| 17C314.2 | Analyse the issues related to clock synchronization and the need for global state in distributed system and compare different algorithms | K4 |
| 17C314.3 | Compare various distributed mutual exclusion algorithms and deadlock detection algorithms. | K4 |
| 17C314.4 | Analyse the agreement protocols and fault tolerance mechanisms in distributed system. | K4 |
| 17C314.5 | Identify the features of P2P, shared memory systems. | K3 |

**MAPPING OF COURSE OUTCOMES WITH PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOME**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course Out Comes** | **Level of CO** | **Program Outcomes** | | | | | | | | | | | | | **Program Specific Outcomes** | | |
| PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | | PSO2 | PSO3 |
| K3 | K4 | K5 | K5 | K6 | K3 | K2 | K3 | K3 | K2 | K3 | K3 |  | |  |  |
| **17C314.1** | K3 | 3 | 2 |  |  |  |  |  |  |  |  |  | 3 | 1 | |  |  |
| **17C314.2** | K4 | 3 | 3 | 2 |  |  |  |  |  |  |  |  | 2 | 1 | | 1 |  |
| **17C314.3** | K4 | 3 | 3 | 2 |  |  |  |  |  |  |  |  | 2 | 1 | | 1 |  |
| **17C314.4** | K4 | 3 | 3 | 2 |  |  |  |  |  |  |  |  | 2 | 1 | | 1 | 1 |
| **17C314.5** | K3 | 3 | 2 | 1 |  |  |  |  |  |  |  |  | 3 | 1 | | 1 |  |
| **Average Values** |  | 3 | 2.6 | 1.75 |  |  |  |  |  |  |  |  | 2.4 | 1 | | 1 | 1 |

# JUSTIFICATION FOR MAPPING COs WITH POs

|  |  |  |  |
| --- | --- | --- | --- |
| **COURSE OUT COMES** | **MAPPED WITH**  **POs & PSOs** | **LEVEL**  **OF MAPPING** | **JUSTIFICATION** |
| 17C314.1 | PO-1 | 3 | Students identify major issues in distributed systems by applying engineering knowledge on operating systems, networks.. |
| PO-2 | 2 | Students will be able to identify the core concepts in distributed systems by analyzing major issues like fault tolerance, replication etc.. |
| PO12 | 3 | Students will have the preparation to engage in life long learning in new concepts emerging from distributed systems. |
| 17C314.2 | PO-1 | **3** | Students will be able to know about synchronization of clocks by applying engineering knowledge. |
| PO-2 | **3** | Students will be able to analyze the issues related to clock synchronization like message complexity, delay etc. |
| PO-3 | 2 | Students will be able to design solutions for clock synchronization |
| PO12 | **2** | Students will be able to recognize the need for new solutions in synchronization & global state |
| 17C314.3 | PO1 | **3** | Students will be able to know about algorithms for mutual exclusion and deadlock detection by applying knowledge of operating system. |
| PO2 | **3** | Students will be able to compare various distributed mutual exclusion algorithms and deadlock detection algorithms. |
| PO3 | **2** | Students will be able to design solutions for mutual exclusion and deadlock detection. |
| PO12 | **2** | Students will be able recognize the need of optimizing deadlock detection algorithms |
| 17C314.4 | PO1 | **3** | Students will be able to know about fault tolerance mechanisms by applying engineering knowledge on operating systems |
| PO2 | **3** | Students will be able to analyse the agreement protocols and fault tolerance mechanisms in distributed system in terms of messages and no of phases. |
| PO3 | **2** | Students will be able to develop solutions for fault tolerance |
| PO12 | **2** | Students will be able recognize the need of optimizing fault tolerant mechanisms |
| 17C314.5 | PO1 | **3** | Students will be able to identify the features of P2P, shared memory systems |
| PO2 | **2** | Students will be able to analyse distributed routing algorithms in terms of routing complexity |
| PO3 | **1** | Students will be able to develop programs for shared memory |
| PO12 | **3** | Students will be able to learn distributed shared memory algorithms for efficient sharing |

ASSESSMENT WEIGHTAGE COMPONENTS

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Direct Assessment** | **TARGET PERCENTAGE** | | | | **CO WEIGHTAGE** | | | | |
| **TARGET LEVEL %** | **% OF STUDENTS** | | | **CO1** | **CO2** | **CO3** | **CO4** | **CO5** |
| **LEVEL 1** | **LEVEL 2** | **LEVEL 3** |
| **Assignment – I** | 70 | 60 | 70 | 80 | 20 |  |  |  |  |
| **Assignment – II** | 70 | 60 | 70 | 80 |  | 20 |  |  |  |
| **IAT – I** | 50 | 50 | 60 | 70 | 80 | 80 |  |  |  |
| **Assignment – III** | 70 | 60 | 70 | 80 |  |  | 20 |  |  |
| **Assignment – IV** | 70 | 60 | 70 | 80 |  |  |  | 20 |  |
| **IAT – II** | 50 | 50 | 60 | 70 |  |  | 80 | 80 |  |
| **Assignment – V** | 70 | 60 | 70 | 80 |  |  |  |  | 20 |
| **IAT – III** | 50 | 50 | 60 | 70 |  |  |  |  | 80 |
| **University Exam** | B | 60 | 70 | 80 |  |  |  |  |  |
| **CO WEIGHTAGE TOTAL** | | | | | 100 | 100 | 100 | 100 | 100 |

**CS8603 DISTRIBUTED SYSTEMS**

**UNIT I INTRODUCTION**

Introduction: Definition –Relation to computer system components –Motivation –Relation to parallel systems – Message-passing systems versus shared memory systems –Primitives for distributed communication –Synchronous versus asynchronous executions –Design issues and challenges. A model of distributed computations: A distributed program –A model of distributed executions –Models of communication networks –Global state – Cuts –Past and future cones of an event –Models of process communications. Logical Time: A framework for a system of logicalclocks –Scalar time –Vector time – Physical clock synchronization: NTP.

**UNIT II MESSAGE ORDERING & SNAPSHOTS**

Message ordering and group communication: Message ordering paradigms –Asynchronous execution with synchronous communication –Synchronous program order on an asynchronous system –Group communication – Causal order (CO) – Total order. Global state and snapshot recording algorithms: Introduction –System model and definitions –Snapshot algorithms for FIFO channels

**UNIT III DISTRIBUTED MUTEX & DEADLOCK**

Distributed mutual exclusion algorithms: Introduction – Preliminaries – Lamport‘s algorithm –Ricart-Agrawala algorithm – Maekawa‘s algorithm – Suzuki–Kasami‘s broadcast algorithm. Deadlock detection in distributed systems: Introduction – System model – Preliminaries –Models of deadlocks – Knapp‘s classification –Algorithms for the single resource model, the AND model and the OR model.

**UNIT IV RECOVERY & CONSENSUS**  
Checkpointing and rollback recovery: Introduction – Background and definitions – Issues in failure recovery – Checkpoint-based recovery – Log-based rollback recovery – Coordinated checkpointing algorithm – Algorithm for asynchronous checkpointing and recovery. Consensus and agreement algorithms: Problem definition – Overview of results – Agreement in a failure –free system – Agreement in synchronous systems with failures.

**UNIT V P2P & DISTRIBUTED SHARED MEMORY**  
Peer-to-peer computing and overlay graphs: Introduction – Data indexing and overlays – Chord – Content addressable networks – Tapestry. Distributed shared memory: Abstraction and advantages – Memory consistency models –Shared memory Mutual Exclusion.

**TEXT BOOKS:**

1. Kshemkalyani, Ajay D., and MukeshSinghal. Distributed computing: principles, algorithms, and systems. Cambridge University Press, 2011.

2. George Coulouris, Jean Dollimore and Tim Kindberg, ―Distributed Systems Concepts and Design‖, Fifth Edition, Pearson Education, 2012.

**REFERENCES:**

1. Pradeep K Sinha, "Distributed Operating Systems: Concepts and Design", Prentice Hall of India, 2007.

2. MukeshSinghal and Niranjan G. Shivaratri. Advanced concepts in operating systems. McGraw-Hill, Inc., 1994.

3. Tanenbaum A.S., Van Steen M., ―Distributed Systems: Principles and Paradigms‖, Pearson Education, 2007.

4. Liu M.L., ―Distributed Computing, Principles and Applications‖, Pearson Education, 2004.

5. Nancy A Lynch, ―Distributed Algorithms‖, Morgan Kaufman Publishers, USA, 2003.

**ONLINE RESOURCES:**

OR1) www.nptel.ac.in

**UNITWISE COURSE DELIVERY PLAN**

**UNIT I**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **UNIT–I INTRODUCTION** | | | | | | | | | |
| **S.No** | **Proposed Lecture**  **Date** | **Topic** | **Actual Lecture**  **Date** | **Pertaining**  **CO(s)** | **Highest Cognitive**  **Level** | **Mode of Delivery** | **Delivery Resource**  **s** | **LU Outcomes** | **Re mar**  **ks** |
| **Student will be able to** |
| **1** |  | Introduction: Definition –Relation to computer system components |  | CO1 | K1 |  | T1 |  |  |
| **2** |  | Motivation –Relation to parallel systems |  | CO1 | K1 |  | T1 |  |  |
| **3** |  | Message-passing systems versus shared memory systems |  | CO1 | K2 |  | T1 |  |  |
| **4** |  | Primitives for distributed communication –Synchronous versus asynchronous executions |  | CO1 | K2 |  | T1 |  |  |
| **5** |  | Design issues and challenges |  | CO1 | K3 |  | T1, T2 |  |  |
| **6** |  | A model of distributed computations: A distributed program –A model of distributed executions |  | CO1 | K2 |  | T1 |  |  |
| **7** |  | Models of communication networks –Global state – Cuts –Past and future cones of an event |  | CO1 | K2 |  | T1 |  |  |
| **8** |  | Models of process communications. |  | CO1 | K2 |  | T1 |  |  |
| **9** |  | Logical Time: A framework for a system of logical clocks –Scalar time –Vector time |  | CO1 | K2 |  | T1 |  |  |
| **10** |  | Physical clock synchronization: NTP. |  | CO1 | K2 |  | T1,R1 |  |  |
| **11** |  | WWW |  | CO1 | K2 |  | OR1 |  |  |

**UNIT II MESSAGE ORDERING & SNAPSHOTS**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **UNIT–II MESSAGE ORDERING & SNAPSHOTS** | | | | | | | | | |
| **S.No** | **Proposed Lecture**  **Date** | **Topic** | **Actual Lecture**  **Date** | **Pertaining**  **CO(s)** | **Highest Cognitive**  **Level** | **Mode of Delivery** | **Delivery Resource**  **s** | **LU Outcomes** | **Re mar**  **ks** |
| **Student will be able to** |
| **1** |  | Message ordering and group communication: Message ordering paradigms |  | CO2 | K2 |  | T1 |  |  |
| **2** |  | Asynchronous execution with synchronous communication |  | CO2 | K2 |  | T1 |  |  |
| **3** |  | Synchronous program order on an asynchronous system |  | CO2 | K2 |  | T1 |  |  |
| **4** |  | Group communication |  | CO2 | K1 |  | T1, T2 |  |  |
| **5** |  | Causal order (CO) |  | CO2 | K1 |  | T1 |  |  |
| **6** |  | Total order |  | CO2 | K4 |  | T1 |  |  |
| **7** |  | Global state and snapshot recording algorithms: Introduction |  | CO2 | K2 |  | T1,T2 |  |  |
| **8** |  | System model and definitions |  | CO2 | K2 |  | T1 |  |  |
| **9** |  | Snapshot algorithms for FIFO channels |  | CO2 | K4 |  | T1,R1 |  |  |
| **10** |  | Snapshot algorithms for FIFO channels |  | CO2 | K4 |  | T1, R1 |  |  |
| **11** |  | Java RMI |  | CO2 | K3 |  | T2 |  |  |

**UNIT III DISTRIBUTED MUTEX & DEADLOCK**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **UNIT–III DISTRIBUTED MUTEX & DEADLOCK** | | | | | | | | | |
| **S.No** | **Proposed Lecture**  **Date** | **Topic** | **Actual Lecture**  **Date** | **Pertaining**  **CO(s)** | **Highest Cognitive**  **Level** | **Mode of Delivery** | **Delivery Resource**  **s** | **LU Outcomes** | **Re mar**  **ks** |
| **Student will be able to** |
| **1** |  | Distributed mutual exclusion algorithms: Introduction, Preliminaries |  | CO3 | K2 |  | T1 |  |  |
| **2** |  | Lamport‘s algorithm |  | CO3 | K3 |  | T1 |  |  |
| **3** |  | Ricart-Agrawala algorithm |  | CO3 | K3 |  | T1,T2 |  |  |
| **4** |  | Maekawa‘s algorithm |  | CO3 | K4 |  | T1,T2 |  |  |
| **5** |  | Suzuki–Kasami‘s broadcast algorithm |  | CO3 | K4 |  | T1 |  |  |
| **6** |  | Deadlock detection in distributed systems: Introduction – System model – Preliminaries |  | CO3 | K1 |  | T1 |  |  |
| **7** |  | Models of deadlocks |  | CO3 | K1 |  | T1 |  |  |
| **8** |  | Knapp‘s classification |  | CO3 | K2 |  | T1 |  |  |
| **9** |  | Algorithms for the single resource model, the AND model and the OR model. |  | CO3 | K4 |  | T1 |  |  |
| **10** |  | Content Beyond syllabus  OpenMP -Parallel Programming in C |  | CO3 | K3 |  | OR1 |  |  |

**UNIT IV RECOVERY& CONSENSUS**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **UNIT–IV** | | | | | | | | | |
| **S.No** | **Proposed Lecture**  **Date** | **Topic** | **Actual Lecture**  **Date** | **Pertaining**  **CO(s)** | **Highest Cognitive**  **Level** | **Mode of Delivery** | **Delivery Resource**  **s** | **LU Outcomes** | **Re mar**  **ks** |
| **Student will be able to** |
| **1** |  | Checkpointing and rollback recovery: Introduction – Background and definitions |  | CO4 | K2 |  | T1 |  |  |
| **2** |  | Issues in failure recovery |  | CO4 | K3 |  | T1 |  |  |
| **3** |  | Checkpoint-based recovery |  | CO4 | K2 |  | T1 |  |  |
| **4** |  | Log-based rollback recovery |  | CO4 | K2 |  | T1 |  |  |
| **5** |  | Coordinated checkpointing algorithm |  | CO4 | K4 |  | T1 |  |  |
| **6** |  | Algorithm for asynchronous checkpointing and recovery |  | CO4 | K4 |  | T1 |  |  |
| **7** |  | Consensus and agreement algorithms: Problem definition |  | CO4 | K2 |  | T1 |  |  |
| **8** |  | Overview of results, Agreement in a failure- free system |  | CO4 | K4 |  | T1 |  |  |
| **9** |  | Agreement in synchronous systems with failures |  | CO4 | K2 |  | T1 |  |  |
| **10** |  | Content Beyond syllabus  Hadoop Installation |  | CO4 | K3 |  | OR1 |  |  |

**UNIT-V PEER TO PEER COMPUTING & SHARED MEMORY**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **UNIT–V PEER TO PEER COMPUTING & SHARED MEMORY** | | | | | | | | | |
| **S.No** | **Proposed Lecture**  **Date** | **Topic** | **Actual Lecture**  **Date** | **Pertaining**  **CO(s)** | **Highest Cognitive**  **Level** | **Mode of Delivery** | **Delivery Resource**  **s** | **LU Outcomes** | **Re mar**  **ks** |
| **Student will be able to** |
| **1** |  | Peer-to-peer computing and overlay graphs: Introduction |  | CO5 | K2 |  | T1,T2 |  |  |
| **2** |  | Data indexing and overlays |  | CO5 | K3 |  | T1 |  |  |
| **3** |  | Chord |  | CO5 | K3 |  | T1 |  |  |
| **4** |  | Content addressable networks |  | CO5 | K2 |  | T1 |  |  |
| **5** |  | Tapestry. |  | CO5 | K3 |  | T1,T2 |  |  |
| **6** |  | Distributed shared memory: Abstraction and advantages |  | CO5 | K2 |  | T1 |  |  |
| **7** |  | Memory consistency models |  | CO5 | K3 |  | T1 |  |  |
| **8** |  | Shared memory Mutual Exclusion. |  | CO5 | K3 |  | T1 |  |  |
| **9** |  | Pastry |  | CO5 | K2 |  | T2 |  |  |

**CS8603 - DISTRIBUTED SYSTEMS**

**UNIT I INTRODUCTION**

**POSSIBLE ASSESSMENT QUESTIONS**

|  |  |  |  |
| --- | --- | --- | --- |
| Q .No | Questions | BLOOM’S  LEVEL | CO’s |
| PART A | | | |
| 1. | **What** is a distributed system? | K1 | C314.1 |
| 2. | **Identify** the features of distributed systems? | K3 | C314.1 |
| 3. | **What** are the three types of parallel systems? | K1 | C314.1 |
| 4. | **Classify** processing modes of Flynn’s taxonomy? | K2 | C314.1 |
| 5 | **What** do you mean by non-blocking primitive? | K1 | C314.1 |
| 6 | **Compare** synchronous communication & asynchronous communication | K2 | C314.1 |
| 7 | **Explain** the send primitive and receive primitive. | K2 | C314.1 |
| 8 | **What** are the various forms of load balancing? | K1 | C314.1 |
| 9 | **What** are Ubiquitous systems? | K1 | C314.1 |
| 10 | **What** do you mean by peer-to-peer computing? | K1 | C314.1 |
| 11 | **What** are agents? | K1 | C314.1 |
| 12 | **Identify**the two basic models of process communications? | K3 | C314.1 |
| 13 | **Define** global state | K1 | C314.1 |
| 14 | **Compare** consistent cut and inconsistent cut | K2 | C314.1 |
| 15 | **Define** scalar time | K1 | C314.1 |
| 16 | **Define** vector time | K1 | C314.1 |
| 17 | **What** is NTP? | K1 | C314.1 |
| PART-B | | | |
| 1. | **Identify** the key algorithmic challenges in distributed computing. (13) | K3 | C314.1 |
| 2. | i)**Outline** Omega and Butterfly networks with example value n = 16 (7)  ii) **Explain**the functions need to address while designing a distributed computing system. (6) | K2 | C314.1 |
| 3. | **Summerize**the primitives for distributed communication.(13) | K2 | C314.1 |
| 4. | **Identify**various processing modes of flynn taxonomy. | K3 | C314.1 |
| 5 | **Explain**about the synchronous versus asynchronous communication | K2 | C314.1 |
| 6 | **Summerize** the capabilities and rules for implementation of logical clocks. | K2 | C314.1 |
| 7 | **Identify**and explain the basic properties of scalar time.(13) | K3 | C314.1 |
| 8. | **Identify**and explain the basic properties of vector time.(13) | K3 | C314.1 |
| 9. | **Explain**NTP for synchronizing system of physical clocks in distributed systems.(13) | K2 | C314.1 |
| 10. | **Explain** relation to parallel computers | K2 | C314.1 |
| 11. | **Identify** the challenges in distributed system. | K3 | C314.1 |
| PART-C | | | |
| 1 | A user arrives at a railway station that she has never visited before, carrying a PDA that is capable of wireless networking. Suggest how the user could be provided with information about the local services and amenities at that station, without entering the station’s name or attributes. **What** technical challenges must be overcome? Discuss in detail. (15) | K1 | C314.1 |
| 2 | **Identify**the requirements and aspects needed for reliable and fault-tolerant distributed systems.(15) | K3 | C314.1 |
| 3 | **Show** that all events on the surface of the past cone of an event are message send events. Likewise, show that all events on the surface of the future cone of an event are message receive events.(15) | K2 | C314.1 |

**CS8603- DISTRIBUTED SYSTEMS**

**UNITII**

**POSSIBLE ASSESSMENT QUESTIONS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Q .No** | **Questions** | **BLOOM’S LEVEL** | **CO’s** |
| **PART A** | | | |
| 1. | **What** are the message ordering paradigms? | K1 | C314.2 |
| 2. | **Compare** closed group Vs open group algorithm. | K2 | C314.2 |
| 3. | **What** iscrown criterion theorem? | K1 | C314.2 |
| 4. | **Explain** message broadcast. | K2 | C314.2 |
| 5. | **Define** time stamp. | K1 | C314.2 |
| 6. | **Compare** multiway rendezvous and binary rendezvous. | K2 | C314.2 |
| 7. | **Analyse**the roles and responsibilities of distributed systems. | K4 | C314.2 |
| 8. | **What** are the characteristics of multicast communication? | K1 | C314.2 |
| 9. | **Differentiate** multicasting Vs unicasting. | K2 | C314.2 |
| 10. | **Identify** the two popular orders for the delivery of messages in group communication. | K3 | C314.2 |
| 11. | **Identify** consistent snapshot. | K3 | C314.2 |
| 12. | **Discover** the criteria that must be met by a causal ordering protocol. | K4 | C314.2 |
| 13. | **What** are the necessary conditions to satisfy the consistent global state? | K1 | C314.2 |
| 14. | **State** the property for causal delivery of messages. | K1 | C314.2 |
| 15. | **Outline**an interpretation in terms of a cut. | K2 | C314.2 |
| 16. | **What** is consistent cut? | K1 | C314.2 |
| 17. | **Outline** marker sending rule. | K2 | C314.2 |
| 18. | **What** is marker receiving rule? | K1 | C314.2 |
| 19. | **What** is complexity? | K1 | C314.2 |
| 20. | **Show** how to prove the correctness of the algorithm. | K2 | C314.2 |
| **PART B** | | | |
| 1 | **i)Differentiate** FIFO and non-FIFO executions.(7)  ii)**Explain** on causally ordered executions (6) | K2 | C314.2 |
| 2 | **Show** with an equivalent timing diagram of a synchronous execution on an asynchronous system.(13) | K2 | C314.2 |
| 3 | **Show** with an equivalent timing diagram of a asynchronous execution on a synchronous system.(13) | K2 | C314.2 |
| 4 | **Illustrate** realizable with synchronous communication (RSC) execution.(13) | K2 | C314.2 |
| 5 | i) **Compare**the hierarchy of execution classes. (7)  ii) Examine the crown test to determine the existence of cyclic dependencies among messages.(6) | K4 | C314.2 |
| 6 | **Explain** the channels to simulate an execution using asynchronous primitives on a synchronous system.(13) | K2 | C314.2 |
| 7 | **Analyse**the channels to simulate an execution using synchronous primitives on an asynchronous system.(13) | K4 | C314.2 |
| 8 | **Explain** a simple algorithm defined by Bagrodia.(13) | K4 | C314.2 |
| 9 | **Explain** chandy and lamport algorithm (13) | K4 | C314.2 |
| 10 | **Analyze** in detail about the centralized algorithm to implement total order and causal order of messages. (13) | K4 | C314.2 |
| 11 | **Examine** the necessary and sufficient conditions for causal ordering. (13) | K4 | C314.2 |
| 12 | **Explain**in detail about the distributed algorithm to implement total order and causal order of messages. (13) | K4 | C314.2 |
| **PART-C** | | | |
| 1 | Consider a distributed system where every node has its physical clock and all physical clocks are perfectly synchronized. **Discover** an algorithm to record global state assuming the communication network is reliable.(15) | K4 | C314.2 |
| 2 | **Illustrate** the asynchronous executions and of crowns.  (a) Crown of size 2.  (b) Another crown of size 2.  (c) Crown of size 3. (15) | K2 | C314.2 |

**CS8603- DISTRIBUTED SYSTEMS**

**UNIT III**

**POSSIBLE ASSESSMENT QUESTIONS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Q .No** | **Questions** | **BLOOM’S LEVEL** | **CO’s** |
| **PART A** | | | |
| 1. | **What** are the three basic approaches for implementing distributed mutual exclusion? | K1 | C314.3 |
| 2 | **Explain** idle token. | K2 | C314.3 |
| 3 | **Outline**the conditions for maekawa’s algorithm. | K2 | C314.3 |
| 4 | **List** the three types of messages for Deadlock handling. | K4 | C314.3 |
| 5 | **What** are the essential requirements of mutual exclusion? | K1 | C314.3 |
| 6 | **Identify** the two design issues for suzuki–kasami’s. | K3 | C314.3 |
| 7 | **How** ricart–agrawala algorithm achieves mutual exclusion**.** | K2 | C314.3 |
| 8 | **Explain** howmaekawa’s algorithm achieves mutual exclusion. | K2 | C314.3 |
| 9 | **Show**in diagram the wait for graph (WFG). | K2 | C314.3 |
| 10 | **What** are the states in a process.? | K1 | C314.3 |
| 11 | **Explain** the three strategies for handling deadlocks. | K2 | C314.3 |
| 12 | **What** is broadcast algorithm? | K1 | C314.3 |
| 13 | **Outline** some algorithms of mutual exclusion | K2 | C314.3 |
| 14 | **What**is deadlock resolution? | K1 | C314.3 |
| 15 | **Develop**the facts of global state detection-based deadlock detection? | K3 | C314.3 |
| 16 | **Define**the features of Mitchell and Merritt’s algorithm. | K1 | C314.3 |
| 17 | **Compare** the various models of deadlock | K2 | C314.3 |
| 18 | **Explain** about Knapp‘s classification. | K2 | C314.3 |
| 19 | **Compare** AND model and the OR model. | K2 | C314.3 |
| 20 | **Summarize** about issues in deadlock detection. | K2 | C314.3 |
| **PART B** | | | |
| 1. | **Explain** in detail about Lamport‘s algorithm. | K4 | C314.3 |
| 2 | **Explain**Ricart-Agrawala algorithm with required pseudocode | K4 | C314.3 |
| 3 | **Explain**in detail about Maekawa‘s algorithm with example. | K3 | C314.3 |
| 4 | **Explain**about Suzuki–Kasami‘s broadcast algorithm. | K4 | C314.3 |
| 5 | **Outline** the basic need of Deadlock detection in distributed systems. | K2 | C314.3 |
| 6 | **Explain** the system model of deadlock detection in distributed systems. | K2 | C314.3 |
| 7 | **Interpret** Knapp‘s classification with example | K2 | C314.3 |
| 8 | **Compare** features of AND model and the OR model. | K4 | C314.3 |
| **PART C** | | | |
| 1 | Show that in the ricart–agrawala algorithm the critical section is accessed in increasing order of timestamp. **Does** the same hold in maekawa’s algorithm?(15) | K4 | C314.3 |
| 2 | Suppose all the processes in the system are assigned priorities which can be used to totally order the processes. **Modify** chand y*et al.*’s algorithm for the AND model so that when a process detects a deadlock, it also knows the lowest priority deadlocked process.(15) | K4 | C314.3 |
| 3 | Consider the following simple approach to handle deadlocks in distributed systems by using “time-outs”: a process that has waited for a specified period for a resource declares that it is deadlocked and aborts to resolve the deadlock. **Explain** what are the shortcomings of using this method?(15) | K4 | C314.3 |

**CS8603- DISTRIBUTED SYSTEMS**

**UNIT IV**

**POSSIBLE ASSESSMENT QUESTIONS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Q .No** | **Questions** | **BLOOM’S LEVEL** | **CO’s** |
| **PART A** | | | |
| 1. | **Define** a checkpoint. | K1 | C314.4 |
| 2 | Mention the **types** of rollback recovery schemes. | K2 | C314.4 |
| 3 | **Contrast** Coordinated and Communication-induced check pointing? | K4 | C314.4 |
| 4 | **Define** a consistent system state. | K1 | C314.4 |
| 5 | **What** is a recovery line? Which lead to domino effect? | K1 | C314.4 |
| 6 | **What** is meant by Garbage collection? | K1 | C314.4 |
| 7 | **List** the disadvantages Uncoordinated Check pointing. | K4 | C314.4 |
| 8 | **What** are the steps needed for Rollback Dependency Graph? | K1 | C314.4 |
| 9 | **How** does the garbage collection algorithm based on a rollback dependency graph work? | K2 | C314.4 |
| 10 | **List** the advantages Coordinated Check pointing. | K4 | C314.4 |
| 11 | **What** is Model-based Communication-induced check pointing? | K1 | C314.4 |
| 12 | **Define** Log-based Recovery. | K1 | C314.4 |
| 13 | **What** are the failures in Distributed Systems? | K1 | C314.4 |
| 14 | **What** is Byzantine Agreement? | K1 | C314.4 |
| 15 | **What** do you mean by Consensus Problem? | K1 | C314.4 |
| PART-B | | | |
| 1. | **Explain** about a Local Checkpoint, Different Types of Messages and the issues in Failure Recovery. | K2 | C314.4 |
| 2 | **Analyze** in brief about uncoordinated, coordinated check pointing and Communication-Induced Check pointing techniques. | K4 | C314.4 |
| 3 | **Explain** about Koo and Toueg coordinated check pointing and recovery technique. | K4 | C314.4 |
| 4 | **Summerize** about Asynchronous Check pointing and Recovery. | K2 | C314.4 |
| 5 | **Explain**Manivannan-Singhal Quasi Synchronous Check pointing and Recovery Algorithm. | K4 | C314.4 |
| 6 | **Explain** the steps for Bzantine Generals (iterative formulation), Synchronous, Message-passing. | K4 | C314.4 |
| 7 | **Explain** the code for the Phase King Algorithm. | K4 | C314.4 |
| 8 | **Explain** the code for the Epsilon Consensus (message-passing, asynchronous). | K4 | C314.4 |
| 9 | **Explain** about Two-process Wait-free Consensus using FIFO Queue, Compare & Swap | K2 | C314.4 |
| 10 | **Outline** the features of Non-blocking Universal Algorithm. | K2 | C314.4 |
| **PART-C** | | | |
| 1 | Consider the following simple check pointing algorithm. A process takes a local checkpoint right after sending a message. Createthat the last checkpoint at all processes will always be consistent. **What are the trade-offs** with this method?(15) | K4 | C314.4 |
| 2 | Give and **analyse**a rigorous proof of the impossibility of a min-process, non-blocking check pointing algorithm.(15) | K4 | C314.4 |

**CS8603- DISTRIBUTED SYSTEMS**

**UNIT V PROCESS & RESOURCE MANAGEMENT**

**POSSIBLE ASSESSMENT QUESTIONS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Q .No** | **Questions** | **BLOOM’S LEVEL** | **CO’s** |
| **PART A** | | | |
| 1 | **What** are the Desirable characteristics and performance features of P2P systems. | K1 | C314.5 |
| 2 | **What** are the steps to search for content and to determine a node from which to download the content? | K1 | C314.5 |
| 3 | **Identify** the types of indexing mechanisms? | K3 | C314.5 |
| 4 | **Tell** about P2P network topology. | K1 | C314.5 |
| 5 | **Identify** the advantages of unstructured overlays? | K3 | C314.5 |
| 6 | **What** are the main message types used by Gnutella? | K1 | C314.5 |
| 7 | **What** is content-addressible network? | K1 | C314.5 |
| 8 | **What** are the three core components of a CAN? | K1 | C314.5 |
| 9 | **Define** Distributed shared memory. | K1 | C314.5 |
| 10 | **What** are the advantages of Distributed shared memory. | K1 | C314.5 |
| 11 | **What** are the disadvantages of Distributed shared memory. | K1 | C314.5 |
| 12 | **Outline** the issues in designing a DSM system. | K2 | C314.5 |
| 13 | **Define** causality relation for shared memory systems. | K1 | C314.5 |
| 14 | **What** is Entry consistency? | K1 | C314.5 |
| 15 | **What** are minimal sequence of operations for fast mutual exclusion? | K1 | C314.5 |
| **PART-B** | | | |
| 1. | **Explain** about Data indexing and overlays. | K2 | C314.5 |
| 2 | **Identify** issues and procedures used in i) Simple lookup ii) Scalable lookup | K3 | C314.5 |
| 3 | **How** will you manage Churn in Chord distributed hash table? | K1 | C314.5 |
| 4 | **Detect** mechanisms used for Strict consistency/atomicconsistency/linearizability. | K3 | C314.5 |
| 5 | **Explain** about Sequential consistency. | K2 | C314.5 |
| 6 | **Explain** about Causal consistency and PRAM (pipelined RAM) or processor consistency. | K2 | C314.5 |
| 7 | How will you achieve mutual exclusion in Shared memory? **Explain** in Detail. | K2 | C314.5 |
| 8 | **Explain** in detail about Content addressable networks (CAN). | K2 | C314.5 |
| 9 | **Identify** how the distributed system concept is applied in Tapestry. Explain it with its techniques. | K3 | C314.5 |
| 10 | Explain about Distributed shared memory architecture and **identify** its advantages, disadvantages, and issues. | K3 | C314.5 |
| **PART C** | | | |
| 1 | User ‘A’ in delhi wishes to send a file for printing to user ‘B’ in florida, whose system is connected to a printer; while user ‘C’ from tokyo wants to save a video file in the hard disk of user ‘D’ in london. **Model** the required peer-to-peer network architecture.(15) | K3 | C314.5 |
| 2 | Explaina formal proof to justify the correctness of algorithm that **construct** sequential consistency using local read operations.(15) | K3 | C314.5 |
| 3 | **Develop**a detailed implementation of causal consistency, and provide a correctness argument for your implementation.(15) | K3 | C314.5 |

**GUEST LECTURE/SEMINAR/WORKSHOP TO MEET INDUSTRY / PROFESSIONAL REQUIREMENTS/INDUSTRIAL VISIT**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl. No.** | **Date** | **Topic** | **Resource Person/ Industry** | **Key Area Covered** | **Relevant PO** |
| 1 |  | Hadoop Installation (CO4) | Mr. Saju M.E (Phd) | Data availability | PO1, PO5, PO10 |
| 2 |  | OpenMP -Parallel Programming in C(CO3) | Ms.Shanmuga Sundari | Parallel Programming in C | PO1,PO5 |

**TOPICS BEYOND SYLLABUS BASED ON GAP ANALYSIS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Topic** | **Pertaining**  **CO(s)** | **Highest**  **Cognitive**  **Level** | **Mode of**  **Delivery** | **Delivery**  **Resources** |
| **1** |  |  |  |  |  |
| **2** |  |  |  |  |  |
| **3** |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Effective Date :** |  | **Revision No. :** |  |
|  |  |  |  |
| **PREPARED BY** | **REVIEWED BY** | | **APPROVED BY** |
|  |  | |  |
| MEENAKSHIAMMAL.R | SAHILA DEVI.R | | SAHILA DEVI R |
|  |  | |  |
| **Signature** | **Signature** | | **Signature** |
|  |  | |  |
| **COURSE INSTRUCTOR(S)** | **SUBJECT EXPERT** | | **HOD** |
|  |  |  |  |

**PRINCIPAL**