Assignment No. 01

Rubric	*Score (0 to 4)*
Delivery	
Understanding	
Readability	
Discipline	
Total	

Performed On: 03/02/25

Sign: _____

Q.1] Differentiate between RPC / RMI

Ans.

Remote Procedure Call	Remote Method Invocation
① A protocol that allows a functions / procedure to be executed on a remote system	① A Java-specific mechanism to invoke methods on remote objects.
② Language-independent	② Java specific
③ Procedural programming	③ object-oriented programming
④ Typically implemented using lower-level protocols	④ Uses Java remote method protocol for communication.
⑤ Can be used across different languages & platforms.	⑤ Limited to Java environments.
⑥ Requires manual serialization of data structures.	⑥ Automatically serializes Java objects.
⑦ Depends on underlying network process	⑦ Provides built-in security mechanisms like SSL for encryption
⑧ More complex to implement in heterogeneous environments	⑧ Easier to use in Java applications due to built-in support
⑨ Can be optimized based on protocol and implementation	⑨ Slightly slower due to java object serialization and deserialization overhead

Q.2] What are services offered by middleware?

Ans. Middleware is software that connects different application in a distributed system. It helps in communication data management, and security between computers, middleware acts bridge between operating system and application, providing essential services to ensure smooth operations.

1. Communication Services:-
- Middleware enables different application to communicate efficiently by handling message exchange and remote calls.
- Remote Procedure Call (RPC):- Allows calling a function on another machine
- Message oriented Middleware:- Uses message queues for asynchronous communication.

2. Security Service:-
- security is critical in distributed System Middleware
- Authentication:- Ensure that only authorized Users can access resources.
- Encryption:- Protect data during transmission
- Access control:- Restricts Uses permission based on roles.

3. Transaction Management:-
- In distributed System, multiple operations need to be executed together.
- Atomicity:- A transaction is either fully completed or rolled back.
- Consistency:- keeps data accrate across multiple database.
- Durability:- Ensures data remains even after a system failure.

4. Resource Management:-
- Middleware helps manages system resources effiziently by load balancing distributes processing across multiple sources.
- Scalability: Allow System to handle increases workload.
- fault tolarance: Ensures System stability even if a part of system fails

5. Data Management & Synchronization:-
- Middleware ensure data Consistency across distributed System by.
- Data replication:- Create multiple copies of data for reliability
- Data Caching:- stores frequently accessed data to reduce load terms.

6 Transparency service:-
- Middleware hides the complexity of distribute System, making them easier to use.
- location transparency: User can access resources without knowing their physical location.
- Access transparency: Ensures different data formats do not affect system usage

Q.3] Explain Raymond's tree-based algorithm for mutual exclusion.

Ans. Raymond's Algorithm is a distributed mutual exclusion algorithm that organizes processes in a logical tree structure to efficiently manage access to a shared resources. It is a token-based algorithm, meaning that a process must hold a token to enter its critical section.

*Key features of Raymond's Algorithm

1. Tree Structure: The processes are arranges in a logical tree, where each node has a parent and potentially multiple children

2. Single Token: Only one token exists in the system. A process must hold the token to enter critical section.

3. Priority to closest Requests: The token is always help by the process that last used it, ensuring a form of locality based access

4. Efficient Request forwarding: Requests for the token are forwarded up the tree towards token holder.

Example:- Consider a tree with P1 (root) having children P2 and P3, where P3 has a child P4.

1. Suppose P4 requests the token.
2. P4 → P3 → P1
3. P1 sends the token to P3, then to P4

4. P4 executes its critical sections and after completion, passes the token the next requester.

#Advantages:-
- Less message overheads
- Efficient token passing
- Scalability

Disadvantages:-
- single point of failure
- Delay due to tree traversal.

Q.4] Explain :-
(i) Any one Election Algorithm
(ii) Any one Physical Clock synchronization algorithm.

Ans.
(i) Any One Election Algorithm :-

Bully Algorithm :- The Bully Algorithm is used in distributed systems to elect a new coordinator when the existing one fails. It is called a bully algorithm because the process with the highest ID dominates and becomes the leader.

Steps of Bully Algorithm :-

1. Failure Detection : When a process detects that the coordinator is not responding, it initiates the election.

2. Election Process :-
- The process that detects failure sends an election message to all higher-ID
- If an response is received, it declares itself the leader.
- If a higher-ID process responds, that process takes over the election.

3. New Leader selection :
- The process with the highest ID wins the election and broadcasts a coordinator message to all other processes.
- Other processes acknowledge and update their record of the new leader.

(ii) Physical Clock Synchronization Algorithms:-

Cristian's Algorithm:-

Cristian's Algorithm is used to synchronization a client's clock with a time server in a distributed system.

Steps of Cristian's Algorithm:-

1. The client sends a time request message to the time server.
2. The server receives the request and responds with its current time.
3. The client estimates the round-trip delay (RTT) and adjusts its clock using:-

$$T_{adjusted} = T_{server} + \frac{RTT}{2}$$

1. The client sets its clock to the adjusted time

Advantages:-

- Simple & efficient
- Works well in LAN

Disadvantage:-

- Assumes symmetric network delay
- single accurate time server.

Q.5) Write a short note on:
(i) Compare DOS, NOS and MOS
(ii) Goals and Issues related to distribute System.

Ans.

DOS	NOS	MOS
① DOS stand for Distributed operating System	① Network operating System is full form NOS	① Mos stand for Middleware operating system
② An OS that manges multiple connected computers as a single system.	② An OS that manages across networked computer while maintaing their independence	② A software layer that enables communication and resources sharing between distributed application
③ Appear as a single system to users	③ Each system operates independently but network communication is possible.	③ Bridges different OS & application to enables interoperability
④ Transparent, all resources appear as part of the system	④ Explicit, users must manually access shared resources.	④ Provides APIs for seamless sharing and integration
⑤ fault Tolerance is High, due to distributed process management	⑤ fault tolerance is low system does not effect others.	⑤ Fault tolerence is via redundancy and error handling.
⑥ Example: Amoeba, Sprite, plan 9.	⑥ Example: windows server, Linux-based network OS.	⑥ Example:- CORBA Java RMI, .NET Remoting.

(ii) Goals and Issues related to distributed System:-

Goals of Distributed System:- A distributed System is designed to achieve the following objectives:

1. Transparency - Users should experience the system as a single entity rather than multiple interconnected components. This includes:-
   - Access Transparency
   - Location Transparency
   - Replication Transparency
   - Concurrency Transparency
   - failure Transparency.

2. Scalability:- The System should efficiently handle growth in terms of users, data, etc

3. fault Tolerance - The system should continue functioning even if some components fails.

4. Resources sharing - Resources like files, printers, and databases should be easily shared among users.

5. Concurrency - Multiple users/processes should be able to operate simultaneously without interference

6. Openness:- The system should support interoperability and be able to integrate different hardware and software components

7. security:- Mechanisms must be in place to protect data, control access and ensure privacy.

Issues in Distributed systems:- Despite their advantages, distributed systems face several challenges:-

1. Synchronization Issues:- Keeping clocks synchronized across different machines is difficult.

2. Communication failures:- Network failure can disrupt interactions between nodes, requiring fault-tolerant mechanisms.

3. Concurrency control:- Ensuring data consistency when multiple processes across shared resources is complex.

4. Deadlocks- Processes waiting indefinitely for resources held by others can lead to system hangs.

5. Security Risks- Distributed systems are more vulnerable to cyberattacks due to remote access and data sharing.

6. Load Sharing- Efficiently distributing workload across multiple nodes to prevent bottlenecks is challenging.