

DHANALAKSMI COLLEGE OF ENGINEERING, CHENNAI
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
CS6601 - DISTRIBUTED SYSTEMS

UNIT – I: Introduction

PART – A (2 marks)

1. Define – Distributed System

A distributed system is a collection of independent computers that appears to its users as a single coherent system. A distributed system is one in which components located at networked communicate and coordinate their actions only by passing message.

2. List the advantages of distributed systems over centralized systems.

The advantages of distributed systems over centralized systems are

- a) economics,
- b) speed,
- c) inherent distribution,
- d) reliability and
- e) Incremental growth.

3. Mention the examples of distributed system.

The examples of distributed system are

- a) internet,
- b) intranets,
- c) Mobile and ubiquitous computing.

4. What is mobile and ubiquitous computing?

Mobile: computing devices are being carried around.

Ubiquitous: little computing devices are all over the place.

5. Mention the challenges in distributed system.

The challenges in distributed system are

- a) Heterogeneity,
- b) Openness,
- c) Security,

- d) Scalability,
- e) Failure handling,
- f) Concurrency and
- g) Transparency.

6. What are the advantages of Distributed Systems?

The advantages of distributed systems are

- a) Performance,
- b) Distribution,
- c) Reliability (fault tolerance),
- d) Incremental growth,
- e) Sharing of data/resources and
- f) Communication.

7. What are the disadvantages of Distributed Systems?

(M – 16)

The disadvantages of distributed systems are

- a) Difficulties of developing distributed software,
- b) Networking problems and
- c) Security problems.

8. Write the difference between mobile and ubiquitous computing.

Mobile computing is the performance of computing tasks while the user is on the move, or visiting places other than their usual environment.

Ubiquitous computing is the harnessing of many small, cheap computational devices that are present in users' physical environments, including the home, office and even natural settings.

9. Why we need openness?

The degree to which a computer system can be extended and re-implemented is openness.

IEEE = Institute of Electrical and Electronic Engineers

ex: IEEE 802.11 WLAN, IEEE 802.3 Ethernet, W3C = World Wide Web Consortium

10. What is failure handling in distributed systems?

Any process, computer or network may fail independently of the others. Therefore each component needs to be aware of the possible ways in which

11. What is the architectural model?

An architectural model defines the way in which the components of system interact with one another and the way in which they are mapped onto an underlying network of computers.

12. What is the fundamental model?

Fundamental models that help to reveal key problems for the designers of distributed systems. Their purpose is to specify the design issues, difficulties and threats that must be resolved in order to develop distributed systems that fulfill their tasks correctly, reliably and securely. The fundamental model provides abstract views of just those characteristics of distributed systems that affect the dependability characteristics - correctness, reliability and security.

13. What is meant by inter-process communication?

Inter process communication is concerned with the communication between processes in a distributed system, both in its own right and as support for communication between distributed objects. The Java API for inter process communication in the internet provides both datagram and stream communication.

14. What is marshalling and unmarshalling?

Marshalling is the process of taking a collection of data items and assembling them into a form suitable for transmission in a message. Unmarshalling is the process of disassembling them on arrival to produce an Equivalent collection of data items at the destination.

15. What is QOS in distributed systems?

It is not sufficient to provide access to services in distributed systems. In particular, it is also important to provide guarantees regarding the qualities associated with such service access. Examples of such qualities include parameters related to performance, security and reliability.

16. Give some difficulties and threats for distributed systems.

- Widely varying modes of use.
- Wide range of system environments.
- Widely in performance.
- Internal problems.

17. What down the characteristics of inter-process communication?

- i. Synchronous and asynchronous communication
- ii. Message destination
- iii. Reliability
- iv. Ordering

18. What are the issues relating to datagram communication?

- i. Message size
- ii. Blocking
- iii. Time out
- iv. Receive anyway

19. What is the use of middleware?

(M– 15)

Middleware is a layer of software whose purpose is to mask heterogeneity and to provide a convenient programming model to application programmers. Middleware is represented by processes or objects in a set of computers that interact with each other to implement communication and resource sharing support for distributed applications.

20. What is meant by group communication?

Group communication is a multicast operation is more appropriate- this is an operation that sends a single message from one process to each of the members of a group of process, usually in such a way that the membership of the group is transparent to the sender.

21. Name some services and examples of Middleware.

(M – 16)

Examples of middleware are CORBA by OMG, DCOM by Microsoft and Java by RMI by Sun.

Middleware services are as follows:

- a) Naming, b) Security, c) Transactions, d) Persistent storage, and e) Event notification.

22. Name five reasons why to build distributed system.

(N – 16)

The five reasons to build distributed system are

- a) Resource sharing,
- b) Reliability,
- c) Communication,
- d) Computation speed up and
- e) Incremental growth.

23. Discuss the design issues in intranet.

(N – 16)

The design issues in intranet are

- a) Too many organizations ignore end-users,
- b) A new generation of employees is changing the way people work,
- c) Organizations underestimate the cost and complexity of portal redevelopment projects,
- d) Intranets lack fresh, relevant content and
- e) Organizations have no governance model, or have one but don't follow it.

24. What do you mean by transparency?

Transparency is defined as the concealment from the user and the application programmer of the separation of components in a distributed system, so that system is perceived as a whole rather than as a collection of independent components.

25. What are the different forms of transparency?

Eight forms of transparency are

- a) Access transparency,
- b) Location transparency,
- c) Concurrency transparency,
- d) Replication transparency,
- e) Failure transparency,
- f) Mobility transparency,
- g) Performance transparency and
- h) Scaling transparency.

26. What is Access transparency?

This means the users should not need or be able to recognize whether a resource is remote or local. This should allow users to access remote resource in the sameway as a local resource.

27. What is Location transparency?

The two main aspects of location transparency are name transparency, which is the name of the resource should not reveal any hint as to the physical location of the resource and user mobility is, refers to the fact that no matter which machine a user is logged onto.

28. What is Replication transparency?

All distributed operating systems have the provision to create replicas (additional copies) of files and other resources on different nodes of a distributed system and is transparent to the users. The two important issues are naming of replicas and control of replication.

29. What is Failure transparency?

Failure transparency deals with masking from the users' partial failures in the system, such as a communication link failure, a machine failure, or a storage device crash. A DOS having failure transparency property will continue to function, perhaps in a degraded form, in the face of partial failures.

30. What is Migration transparency?

For better performance, reliability and security, an object that is capable of being moved is often migrated from one node to another in a distributed system. The aim is to ensure that the movement of the object is handled automatically by the system.

Part – B (16 Marks)

1. Explain about fundamental model?
2. Explain about architectural model?
3. Write about API for the Internet Protocols?
4. Explain about UDP datagram and TCP stream communication?
5. Explain about client server communication?
6. Explain about CORBA's CDR and Java Object Serialization?
7. With neat diagram explain the various client server models.
8. Describe possible occurrence of security threats in internet.
9. Write short note on: i) External data representation and marshalling
ii) Client-Server communication
10. Explain in details: The interprocess communication in UNIX

UNIT – II: Communication in Distributed System

1. What is meant by client server communication?

The client–server model of computing is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients.

2. What is the use of RMI registry?

The RMI registry is used to store a list of available services. A client uses the registry to make its proxy object, and the Registry is responsible for giving appropriate information to the client so that it can hook up with the server that implements the service.

3. Difference between synchronous and asynchronous communication.

In synchronous form of communication, the sending and receiving processes synchronize at every message. In this case, both send and receive are blocking operations. Whenever a send is issued the sending process is blocked until the corresponding receive is issued. Whenever receive is issued, the process blocks until a message arrives.

In asynchronous form of communication, the use of the send operation is non-blocking in that the sending process is allowed to proceed as soon as the message has been copied to a local buffer and the transmission of the message proceeds in parallel with the sending process. The receive operation can have blocking and non-blocking variants.

4. What are the design issues for RPC?

Three issues that is important in understanding this concept:

- The style of programming promoted by RPC – programming with interfaces;
- The call semantics associated with RPC;
- The key issue of transparency and how it relates to remote procedure calls.

5. What is the difference between RMI and RPC?

Remote Procedure Call or the RPC and the Remote Method Invocation or RMI are both message passing techniques in the Inter Process Communication (IPC). But there are two basic differences between the two methods:

1. RPC supports procedural programming. (i.e. only remote procedures can be invoked.) RMI is object-based. As the name suggests, it is invoked on remote objects.
2. In RPC, the parameters that are passed are ordinary data structures, whereas in RMI object can be passed as parameters.

6. What is request-reply protocol?

The protocol we describe here is based on a trio of communication primitives, doOperation, getRequest and sendReply, this request-reply protocol matches requests to replies. It may be designed to provide certain delivery guarantees. If UDP datagram are used, the delivery guarantees must be provided by the request-reply protocol, which may use the server reply message as an acknowledgement of the client request message

7. What is RPC in distributed systems?

A remote procedure call (RPC) represents a major intellectual breakthrough in distributed computing, with the goal of making the programming of distributed systems look similar, if not identical, to conventional programming – that is, achieving a high level of distribution transparency. This unification is achieved in a very simple manner, by extending the abstraction of a procedure call to distributed environments

8. What are the different parts of distributed object model?

- i. remote object reference
- ii. remote interface
- iii. actions
- iv. exceptions
- v. garbage collections.

9. What is lightweight RPC?

LRPC is a communication facility designed and optimized for communication between protection domains on the same machine.

10. Define – Distributed Object

The term distributed objects usually refers to software modules that are designed to work together, but reside either in multiple computers connected via a network or in different processes inside the same computer. One object sends a message to another object in a remote machine or process to perform some task.

11. What is transparency in RPC?

A transparent RPC mechanism is one in which local procedures and remote procedures are indistinguishable to programmers. RPC uses two types of transparency: syntactic transparency and semantic transparency.

12. What are events and notifications?

The manipulation of an object that results in a change of state is called an event, and the objects that display a view of the current state are send notifications when an event occurs.

Ex: user clicking a button is an event that triggers a notification to the button object.

13. What is purpose of protection?

The purpose of protection system is to prevent accidental or intentional misuse of a system. In multiprogramming operating systems protection is required because of shared resources.

14. What is java RMI?

It is a mechanism that allows a java program running on one computer to apply a method to an object living on a different computer. RMI is an implementation of the distributed object programming model similar to COBRA, but similar and specialized to the java language.

15. What is the main object used in jini specification?

- i. event generator
- ii. remote event listener
- iii. remote event
- iv. third party agents

16. What is the role of Proxy server and mobile code?

(M – 16)

A proxy server provides copies of resources which are managed by other servers. They are typically used as caches for web resources. They maintain a cache of recently visited web pages or other resources.

Mobile code is the codes that are sent from one computer to another and run at the destination. Applets are a well-known and widely used example of mobile code.

17. Define – Inter Process Communication

(M – 16)

A complex programming environment often uses multiple cooperating processes to perform related operations. These processes must communicate with each other and share resources and information. The kernel must provide mechanisms that make this possible. These mechanisms are collectively referred to as inter process communication.

18. Write the characteristics of multicast communication.**(M – 16)**

The characteristics of multicast communication are

- a) Replicated services,
- b) Finding services and their interfaces,
- c) Better performance

19. Define – Network Virtualization**(M – 16)**

Network virtualization or network virtualization (see spelling differences) is the process of combining hardware and software network resources and network functionality into a single, software-based administrative entity, a virtual network.

20. What are the different types of system models?

The different types of system models are

- a) Physical models,
- b) Architectural models,
- c) Fundamental models,
- d) Interaction models,
- e) Failure models and
- f) Security models.

21. Define Physical Models.

A physical model is a representation of the underlying hardware elements of distributed system that abstracts away from specific details of the computer and networking technologies employed.

22. What are the three generations of distributed systems?

The three generations of distributed systems are

- a) Early distributed systems,
- b) Internet-scale distributed systems and
- c) Contemporary distributed systems.

23. What is an Architectural model?

Architectural models describe a system in terms of the computational and communication tasks performed by its computational elements; the computational elements being individual computers or aggregates of them supported by appropriate network interconnections.

24. List the architectural element.

The architectural elements are

- a) Communicating entities
 - i) Objects
 - ii) Components
 - iii) Web services
- b) Communication paradigms
 - i) Direct communication
 - 1. Inter process communication
 - 2. Remote invocation
 - 3. Indirect communication
 - ii) Indirect communication
 - 1. Group communication
 - 2. Publish - subscribe systems
 - 3. Message queues:
 - 4. Tuple spaces
 - 5. Distributed shared memory
- c) Roles and responsibilities
- d) Placement

25. What do you mean by Inter-process communication?

Inter-process communication refers to the relatively low-level support for communication between processes in distributed systems, including message-passing primitives, direct access to the API offered by Internet protocols (socket programming) and support for multicast communication.

26. Write about Remote invocation?

Remote invocation represents the most common communication paradigm in distributed systems, covering a range of techniques based on a two - way exchange between communicating entities in a distributed system and resulting in the calling of a remote operation, procedure or method.

27. What do you mean by Group communication?

Group communication is concerned with the delivery of messages to a set of recipients and hence is a multiparty communication paradigm supporting one-to-many communication. Group communication relies on the abstraction of a group which is represented in the system by a group identifier.

Recipients elect to receive messages sent to a group by joining the group. Senders then send messages to the group via the group identifier, and hence do not need to know the recipients of the message. Groups typically also maintain group membership and include mechanisms to deal with failure of group members.

Part – B (16 Marks)

1. Explain Events and Notification with example?
2. What is RMI? How it is implemented?
3. Explain how communication is done between Distributed Objects?
4. What are threads? Explain the various types of multi- threaded architectures with neat sketch?
5. Explain the architecture of kernel suitable for a distributed system?
6. Discuss in detail the about communication and invocations.
7. Explain distributed object.
8. Describe remote object and its remote interface.
9. Write note on the design issues from RMI.
10. With a neat sketch explain remote procedure call.

UNIT – III: Peer to Peer Services and File System

1. What is distributed file system?

A Distributed File System is a file system with distributed storage and users. DFS provides transparency of location, access, and migration of files. DFS systems use cache replies for efficiency and fault tolerance.

2. What are the goals of distributed file system?

1. Network transparency: Clients should be able to access remote files using the same operations that apply to local files.
2. High availability: Users should have the same easy access to files, irrespective of their physical location.

3. What is high availability and high reliability?

High availability: DFS should continue to function even when partial failures occur due to the failure of one or more components, such as a communication link failure, a machine failure or a storage device crash.

High reliability: In a good distributed file system, the probability of loss of stored data should be minimized as far as practicable.

4. What is transparency?

Transparency: Following are the desirable transparency:

- a. Structure transparency: DFS normally uses multiple file servers for performance, scalability and reliability reasons. Each file server is normally user process or sometimes a kernel process that is responsible for controlling a set of secondary storage devices of the node on which it runs.
- b. Access transparency: Both local and remote files should be accessible in the same way.
- c. Naming transparency: The name of file should give no hint as to where the file is located.
- d. Replication transparency: The clients do not need to know the existence or locations of multiple file copies.

5. Define – Mounting

It is one way merge different name space. The mount mechanism binds together several filename spaces into a single hierarchical structured name space.

6. What are the characteristics of name service?

The characteristics of name service are

- a) they are larger integers or bit strings,
- b) system oriented names also referred as unique identifier,

- c) length is variable and
- d) System oriented names are automatically generated.

7. What is URL?

Uniform Resource Locator is a standard for specifying any kind of information on the internet.

8. What is name service?

It is used by client process to obtain attributes of resources or object when given their names.

9. What is name space?

A name space is a collection of all valid recognized by a particular service. It allows simple but meaningful names to be used. Two types are their i.e., flat name space and hierarchal names.

10. Give disadvantage of flat name spaces.

The disadvantage of flat name spaces is

- a) Flat name spaces can't generalize to large sets of machines because of the single set of identifiers,
- b) Single central name authority was overloaded and
- c) Frequently name address binding changes were costly and cumbersome.

11. What is name resolution?

DNS is designed as a client server application. A host that needs to map an address to a name or a name to an address calls a DNS client named a resolver.

12. What is directory service?

(M – 15)

It is a software application or a set of applications that stores and organizes information about a computer networks users and network resources, and that allows network administrators to manage user's access to those resources.

13. What is global name service?

When the naming database grows from small to large scale, the structure of name space may change. The service should accommodate it.

14. Name the approaches used for name cache implementation.

Two commonly used approaches for name cache implementation are

- a) A cache per process and
- b) A single cache for all processes of a node.

15. Give the two widely used distributed systems.

The two widely used distributed systems is

- a) Sun Network File System and
- b) Andrew File System.

16. What are the modules present in file system?

The modules present in file system are

- a) Directory module,
- b) File module,
- c) Access control module,
- d) File access module,
- e) Block module and
- f) Device module.

17. What are advantages and disadvantages of centralized approach?

Advantages:

It is simple and easy implement.

It is only method used for generating unstructured global unique identifier.

Disadvantages:

Suffers from poor efficiency and poor reliability.

Single global unique identifier generator may become a bottleneck for larger name space.

18. What are the general name service requirements?

A lone lifetime

High availability

Fault isolation

Tolerance of mistrust

19. What is Global Name Service?

Global Name Service (GNS) designed and implemented by Lampson and colleagues at the DEC systems research center in 1986. It provides facilities for resource location, email addressing and authentication.

20. Describe the characteristics of Peer to Peer systems.

(M – 16)

All processes (objects) play similar role. Do not require a server process. Processes (objects) interact without particular distinction between clients and servers. The pattern of communication depends on the particular application.

21. Discuss on LDAP.

(M – 16)

The Lightweight Directory Access Protocol (LDAP) is an open, vendor-neutral, industry standard application protocol for accessing and maintaining distributed directory information services over an Internet Protocol (IP) network.

22. Give the advantages in using name caches in file systems.

(N – 16)

The advantages of using name caches in file systems are

- a) can be used in diskless workstation and
- b) accessing a cache in memory is much faster.

23. List the file accessing models.

(N – 16)

The file accessing models are

- a) accessing remote files
 - i) remote service model
 - ii) data caching model
- b) unit of data transfer
 - i) file level transfer model
 - ii) block level transfer model
 - iii) byte level transfer model
 - iv) record level transfer model

24. Define peer-to-peer systems.

Peer-to-peer systems represent a paradigm for the construction of distributed systems and applications in which data and computational resources are contributed by many hosts on the internet, all of which participate in the provision of a uniform service.

25. What is the goal of peer-to-peer systems?

The goal of peer-to-peer systems is to enable the sharing of data and resources on a very large scale by eliminating any requirements for separately managed servers and their associated infrastructure.

26. What is the Weakness of peer-to-peer systems?

Peer-to-peer systems cannot rely on guaranteed access to individual resources, although they can be designed to make the probability of failure to access a copy of a replicated object arbitrarily small.

27. What is the use of Napster files sharing system?

Napster files sharing system which provided a means for users to share files. Napster became very popular for music exchange soon after its launch in 1999. At its peak, several million users were registered and thousands were swapping music files simultaneously.

28. What is the need to design peer-to-peer middleware systems?

Peer-to-peer middleware systems are designed specifically to meet the need for the automatic placement and subsequent location of the distributed objects managed by peer-to-peer systems and applications.

29. What are the non-functional requirements addressed by peer-to-peer middleware?

The non-functional requirements addressed by peer-to-peer middleware are

- a) Global scalability,
- b) Load balancing,
- c) Optimization for local interactions between neighboring peers,
- d) Accommodating to highly dynamic host availability,
- e) Security of data in an environment with heterogeneous trust and
- f) Anonymity, deniability and resistance to censorship.

30. What are the tasks of routing overlay?

The tasks of routing overlay are

- a) Routing of requests to objects,
- b) Insertion of objects,
- c) Deletion of objects and
- d) Node addition and removal.

31. What is the limitation of Napster?

Napster used a unified index of all available music files. For the application in question, the requirement for consistency between the replicas was not strong, so this did not hamper performance, but for many applications it would constitute a limitation. Unless the access path to the data objects is distributed, object discovery and addressing are likely to become a bottleneck.

32. Define – Naming

A name in a distributed system is a string of bits or characters that is used to refer to an entity. An entity in a distributed system can be practically anything. Typical examples include resources such as hosts, printers, disks and files.

33. Define – Namespace

Names in a distributed system are organized into what is commonly referred to a name space. A name space can be represented as a labeled, directed graph with two types of nodes.

A leaf node represents a named entity and has the property that it has no outgoing edges.

A directory node has number of outgoing edges, each labeled with a name.

34. Define – Root Node

A node which has only outgoing and no incoming edges is called the root node of the naming graph.

35. Define – Absolute Path Name and Relative Path Name

Each path in a naming graph can be referred to by the sequence of labels corresponding to the edges in that path, such as

$$N: \langle \text{label-1}, \text{label-2}, \dots, \text{label-n} \rangle$$

where N refers to the first node in the path. Such a sequence is called a path name. If the first node in a path name is the root of the naming graph, it is called an absolute path name. Otherwise, it is called a relative path name.

36. Define – Name Resolution

Name spaces offer a convenient mechanism for storing and retrieving information about entities by means of names. More generally, given a path name, it should be possible to look up any information stored in the node referred to by that name. The process of looking up a name is called name resolution.

37. What is mount point and mounting point?

The directory node storing the node identifier is called a mount point.

The directory node in the foreign name space is called a mounting point.

Part – B (16 Marks)

1. Briefly explain the architecture and server operation of NFS?
2. Discuss in detail the design and implementation of name services and Domain Name services.
3. Briefly explain the architecture NFS and DFS?
4. Briefly explain the file service architecture?
5. Explain the requirements of Distributed file system?
6. Explain with neat sketch simple dealing room system..
7. Explain in details the characteristics of file system.
8. Write note on : i) Name service ii) Domain name system
9. Explain directory services
10. Explain the algorithm of mutual exclusion.

UNIT – IV: Synchronization and Replication

1. What is the goal of transaction?

The goal of transactions is to ensure that all of the objects managed by a server remain in a consistent state when they are accessed by multiple transactions and in the presence of server crashes.

2. What is dirty read?

The dirty read problem is caused by the interaction between a read operation in one transaction and an earlier write operation in another transaction on the same object.

3. Write any rules for committing the nested transactions?

For a sub transaction to acquire a write lock on an object, no other active transaction can have a read or write lock on that object, and the only retainers of read and write locks on that object are its ancestors.

4. State ACID properties.

ACID properties are

- a) Atomicity,
- b) Consistency,
- c) Isolation and
- d) Durability.

5. What are the three approaches for concurrency control?

The three approaches for concurrency control are

- a) Locking,
- b) Optimistic concurrency control and
- c) Timestamp ordering.

6. How can you recover from transactions?

The strategy for recoverability is to delay commits until after the commitment of any other transaction whose uncommitted state has been observed. In our example, U delays its commit until T commits. In the case that T aborts, then U must abort as well.

7. What are the advantages of nested transactions?

Sub transactions at one level may run concurrently with other sub transactions at the same level in the hierarchy. This can allow additional concurrency in a transaction. When sub transactions run in different servers, they can work in parallel. Sub transactions can commit or abort independently. In comparison with a single transaction, a set of nested sub transactions is potentially more robust.

8. What is the use of locks?

Serializing mechanism uses exclusive locks. In this locking scheme, the server attempts to lock any object that is about to be used by any operation of a client's transaction.

9. Describe about two-phase locking.

The first phase of each transaction is a 'growing phase', during which new locks are acquired. The second phase is a 'shrinking phase', during which the locks are released. This is called two-phase locking.

10. How locking is done in strict two-phase locking?

Any locks applied during the progress of a transaction are held until the transaction commits or aborts. This is called strict two-phase locking.

11. What are the types of locks?

Read locks, write locks and commit locks.

12. What do you mean by lock promotion?

Lock promotion refers to the conversion of a lock to a stronger lock.

13. Define – Deadlock.

Deadlock is a state in which each member of a group of transactions is waiting for some other member to release a lock.

14. What do you mean by deadlock prevention?

An apparently simple but not very good way to overcome the deadlock problem is to lock all of the objects used by a transaction when it starts.

15. How will you detect the deadlock?

Deadlocks may be detected by finding cycles in the wait-for graph. Having detected a deadlock, a transaction must be selected for abortion to break the cycle.

16. What are differences between the two-version locking scheme and an ordinary read-write locking scheme?

Read operations in the two-version locking scheme are delayed only while the transactions are being committed, rather than during the entire execution of transactions – in most cases, the commit protocol takes only a small fraction of the time required to perform an entire transaction.

Read operations of one transaction can cause delays in committing other transactions.

17. What are the drawbacks of deadlocks?

The use of locks can result in deadlock. Deadlock prevention reduces concurrency severely, therefore deadlock situations must be resolved either by the use of timeouts or by deadlock detection. Neither of these is wholly satisfactory for use in interactive programs.

18. What are the phases of transactions?

The phases of transactions are

- a) Working phase,
- b) Validation phase and
- c) Update phase.

19. Compare forward and backward validation.

Forward validation allows flexibility in the resolution of conflicts, whereas backward validation allows only one choice – to abort the transaction being validated.

Backward validation compares a possibly large read set against the old write sets, whereas forward validation checks a small write set against the read sets of active transactions.

Backward validation has the overhead of storing old write sets until they are no longer needed. On the other hand, forward validation has to allow for new transactions starting during the validation process.

20. Define – Starvation.

The prevention of a transaction ever being able to commit is called starvation.

21. Describe about optimistic concurrency control.

In optimistic schemes, a transaction proceeds until it asks to commit, and before it is allowed to commit the server performs a check to discover whether it has performed operations on any objects that conflict with the operations of other concurrent transactions, in which case the server aborts it and the client may restart it. The aim of the check is to ensure that all the objects are correct.

22. Describe about timestamp ordering concurrency control.

In timestamp ordering, a server records the most recent time of reading and writing of each object and for each operation, the timestamp of the transaction is compared with that of the object to determine whether it can be done immediately or must be delayed or rejected. When an operation is delayed, the transaction waits; when it is rejected, the transaction is aborted.

23. Define – Multiversion timestamp ordering.

In multiversion timestamp ordering a list of old committed versions as well as tentative versions is kept for each object. This list represents the history of the values of the object. The benefit of using multiple versions is that read operations that arrive too late need not be rejected.

24. Describe about atomic commit protocols.

A simple way to complete the transaction in an atomic manner is for the coordinator to communicate the commit or abort request to all of the participants in the transaction and to keep on repeating the request until all of them have acknowledged that they have carried it out. This is an example of a one phase atomic commit protocol.

25. Describe about two-phase commit protocol.

The two-phase commit protocol is designed to allow any participant to abort its part of a transaction. Due to the requirement for atomicity, if one part of a transaction is aborted, then the whole transaction must be aborted.

Part-B (16 Marks)

1. Explain briefly about multicast communication?
2. Explain snapshot algorithm
3. Explain briefly about network time protocol
4. Explain briefly about Election algorithm?
5. Explain about clocks, events and process state?
6. Explain briefly about network time protocol
7. Explain about synchronization of physical clock?
8. Explain briefly about Distributed mutual exclusion
9. Explain about distributed debugging?
10. Explain about the logical time and logical clocks.

Unit – V: Process & Resource Management

1. What is the goal of process management?

Process management goal is to make the best possible use of the processing them among all processes.

2. What do you mean by process migration?

Process migration is the relocation of a process from its current location to another node.

3. What is the goal of using threads?

The threads are used to provide fine-grained parallelism for better utilization of the processing capability of the system.

4. What are the desirable features of a good process migration mechanism?

Features for process migration mechanism are

- i) Transparency
- ii) Minimal interferences
- iii) Efficiency
- iv) Minimal residue dependencies
- v) Robustness
- vi) Communication between the co-processes

5. What is the advantage and disadvantage of total freezing?

Advantage:

Simple and easy to implement.

Disadvantage:

If a process is suspended for a long time during migration, timeouts occur and if the process is interactive, the delay will be noticed by the user.

6. What are the mechanism used for handing co-processes?

Two mechanisms are used

- (i) Disallowing separation of co-processes
- (ii) Home node or origin site concept.

7. What are the states of threads?

Running,
Block,
Ready and
Terminated.

8. What are the ways to organize threads of a process?

Three ways to organize threads of a process are

- (i) Dispatcher worker model
- (ii) Team model
- (iii) Pipeline model

9. How can you create threads?

Threads can be created by statically and dynamically.

- 1. Static approach: The number of threads of a process remains fixed for its entire lifetime.
- 2. Dynamic approach: The number of threads of a process keeps changing dynamically.

10. How the threads are terminated?

A thread may either destroy itself when it finishes its job by making an exit call (or) killed from outside by using the kill command and specifying the thread identifier as its parameter.

11. What are two commonly used mutual exclusion techniques?

- 1. Mutex variable and
- 2. Condition variable.

12. What are the techniques used for scheduling processes of a distributed system?

- 1. Task assignment approach,
- 2. Load-balancing approach and
- 3. Load-sharing approach.

13. Define processor thrashing.**(M – 16)**

A scheduling algorithm is said to be unstable if it can enter a state in which all the nodes of the system are spending all of their time migrating processes without accomplishing any useful work in an attempt to properly schedule the processes for better performance. This form of fruitless migration of processes is known as processor thrashing.

14. What is the difference between static and dynamic algorithms?

Static algorithms use only information about average behavior of the system, ignoring the current state of the system.

Dynamic algorithms react to the system state that change dynamically.

15. Differentiate deterministic and probabilistic algorithms.

Deterministic algorithms use the information about the properties of the nodes and the characteristics of the processes to be scheduled to deterministically allocate processes to node.

A probabilistic load balancing algorithms uses information regarding static attributes of the system such as number of nodes, the processing capability of each node, the network topology and so on, to formulate simple process placement rules.

16. What is the difference between centralized and distributed scheduling algorithm?

In a centralized dynamic scheduling algorithm the responsibility of scheduling physically resides on a single node.

In a distributed dynamic scheduling algorithm, the work involved in making process assignment decisions is physically distributed among the various nodes of the system.

17. Differentiate between cooperative and non-cooperative algorithm.

In non-cooperative algorithms, individual entities act as autonomous entities and make scheduling decisions independently of the actions of other entities.

In cooperative algorithms, the distributed entities cooperate with each other to make scheduling decisions.

18. What is local process and remote process?

A local process is one that is processed at its originating node and remote process is one that is processed at a node different from the one which is originated.

19. What are the methods used for determining the threshold value?

Two methods are used to determine the threshold value

1. State policy: In this method, each node has a predefined threshold value depending on its processing capability.
2. Dynamic policy: In this method, the threshold value of a node is calculated as a product of the average work load of all the nodes and a predefined constant.

20. State the single threshold policy.

In single threshold policy, a node accepts new processes if its load is below the threshold value and attempt to transfer local processes and rejects remote execution requests if its load is above the threshold value.

21. State the double threshold policy.

The double threshold policy is also known as high-low policy.

The high-low policy uses two threshold values called high mark and low mark, which divide the space of possible load states of a node into three regions

1. Overloaded
2. Normal
3. Underloaded

22. Name any four state information exchange policies in load balancing algorithm.

- i) Periodic broadcast
- ii) Broadcast when state changes
- iii) On-demand exchange
- iv) Exchange by polling

23. Name the priority assignment policies in load balancing algorithm.

Priority assignment policies are

- i) Selfish
- ii) Altruistic
- iii) Intermediate

24. What are the two commonly used state information exchange in load sharing approach?

- i) Broadcast when state changes.
- ii) Poll when state changes.

25. What are the two location policies in load sharing approach?

- i) Sender initiated location policy and
- ii) Receiver initiated location policy.

26. List the issues in designing load balancing algorithms.

(M – 16)

The issues in designing load balancing algorithms are

- a) Load estimation policy for Load-balancing algorithms,
- b) Process transfer policy for Load-balancing algorithms,
- c) Location policies for Load-sharing algorithms,
- d) State information exchange policies for Load-sharing algorithms,
- e) Priority assignment policy for Load-balancing algorithms and
- f) Migration limiting policy for Load-balancing algorithms.

27. Write any two advantages of process migration.

(M – 16)

The advantages of process migration are

- a) Reducing average response time of processes,
- b) Speeding up individual jobs,
- c) Gaining higher throughput,
- d) Utilizing resources effectively,
- e) Reducing network traffic,
- f) Improving system reliability and
- g) Improving system security.

28. Write down the goals to achieve an optimal assignment.

(N – 16)

The goals to achieve an optimal assignment are

- a) minimization of IPC costs,
- b) quick turnaround time for the complete process,
- c) a high degree of parallelism and
- d) Anefficient utilization of system resources in general.

29. List the features of scheduling algorithms.

(N – 16)

The features of scheduling algorithms are

- a) keep the CPU utilization as high as possible,
- b) throughput must be high,
- c) response time must be high and
- d) Waiting time must be minimal.

30. Define – Altruistic Priority Assignment Policies

Remote processes are given higher priority than local processes. Under this policy remote processes incur lower delays than local processes. It achieves best response time performance of the three policies.

Part – B (16 Marks)

1. Explain byzantine problem in synchronous system?
2. Explain the design and implementation issues of DSM?
3. Explain the CORBA architecture and RMI?
4. Explain sequential consistency and IVY case study?
5. Explain about Release consistency and Munin case study?
6. Explain briefly about CORBA Services?
7. Describe in detail about distributed deadlock?
8. Explain the concurrency control in distributed transactions.
9. Explain in detail about comparison of methods of concurrency control.
10. Explain Time stamp ordering in detail.