SYED AMMAL ENGINEERING COLLEGE
(An ISO 9001: 2008 Certified Institution)
Dr. E.M.Abdullah Campus, Ramanathapuram – 623 502
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CS6601 – DISTRIBUTED SYSTEMS

Two Marks Question with Answers

UNIT 1 INTRODUCTION

PART A

1. **What is a distributed system?**

   A distributed system is one in which components located at networked computers communicate and coordinate their actions only by passing messages. The components interact with each other in order to achieve a common goal.

2. **Mention few examples of distributed systems.**

   Some examples of distributed systems are web search, Massively multiplayer online games(MMOGs), Financial trading markets, SOA based systems etc.

3. **Mention the trends in distributed systems.**

   Following are the trends in distributed systems:

   - Emergence of pervasive networking technology
   - Emergence of ubiquitous computing coupled with desire to support user mobility in distributed systems
   - Increasing demand for multimedia services
   - The view of distributed systems as a utility.

4. **What are backbones in intranets?**

   The intranets are linked together by backbones. A *backbone* is a network link with a high transmission capacity, employing satellite connections, fiber optic cables and other high-bandwidth circuits.

5. **Write short notes about webcasting.**

   Webcasting is an application of distributed multimedia technology. Webcasting is the ability to broadcast continuous media, typically audio or video, over the Internet. It is now commonplace for major sporting or music events to be broadcast in this way often attracting large numbers of viewers.

6. **Define cloud computing.**

   A cloud is defined as a set of Internet-based application, storage and computing services sufficient to support most users' needs, thus enabling them to largely or totally dispense with local data storage and application software. The term cloud computing refers to the practice of using a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer.

7. **What is a cluster computer? Mention its goals.**

   A cluster computer is a set of interconnected computers that cooperate closely to provide a single, integrated high performance computing capability. It consists of a set of loosely or tightly connected computers. Computer clusters have each node set to perform the same task, controlled and scheduled by software.

8. **Write short notes on mobile and ubiquitous computing.**

   Mobile Computing is a technology that allows transmission of data, voice and video via a computer or any other wireless enabled device without having to be connected to a fixed physical link. It involves mobile communication, mobile hardware, and mobile software.

   Ubiquitous computing (ubicomp) is a concept in software engineering and computer science where computing is made to appear anytime and everywhere. In contrast to desktop computing, ubiquitous computing can occur using any device, in any location, and in any format. This paradigm is also described as pervasive computing.

9. **What does the term remote invocation mean?**

   Remote invocation mechanism facilitates to create a distributed application. It provides a remote communication using two objects stub and skeleton. In this client-server approach remote object plays main role and it is an object whose method can be invoked from another JVM. In the client side, stub acts as a gateway. In the server side, skeleton acts as the gateway.

10. **What is the role of middleware?**

The term middleware applies to a software layer that provides a programming abstraction as well as masking the heterogeneity of the underlying networks, hardware, operating systems and programming languages. In addition to solving the problems of heterogeneity, middleware provides a uniform computational model for use by the programmers of servers and distributed applications.

11. What are the challenges of distributed systems?

The main challenges of distributed system are:

- Heterogeneity
- Openness
- Security
- Scalability
- Failure handling
- Concurrency
- Transparency
- Quality of service

12. What is mobile code? Give an example.

The term mobile code is used to refer to program code that can be transferred from one computer to another and run at the destination. Example is Java applet. The need for mobile code is that the code suitable for running on one computer is not necessarily suitable for running on another because executable programs are normally specific both to the instruction set and to the host operating system.

13. What determines the openness of distributed systems?

The openness of a computer system is the characteristic that determines whether the system can be extended and reimplemented in various ways. The openness of distributed systems is determined primarily by the degree to which new resource-sharing services can be added and be made available for use by a variety of client programs.

14. Mention the characteristics of open distributed systems.

- Open systems are characterized by the fact that their key interfaces are published.
- Open distributed systems are based on the provision of a uniform communication mechanism and published interfaces for access to shared resources.
- Open distributed systems can be constructed from heterogeneous hardware and software, possibly from different vendors. But the conformance of each component to the published standard must be carefully tested and verified if the system is to work correctly.

15. What are the 2 security challenges that are not fully met by distributed systems?

- ***Denial of service attacks*****:** One of the security problems is that a user may wish to disrupt a service for some reason. This can be achieved by bombarding the service with such a large number of pointless requests that the serious users are unable to use it. This is called a denial service*of* attack. There have been several denial of service attacks on well-known web services.
- ***Security of mobile code*****:** Mobile code needs to be handled with care. Consider someone who receives an executable program as an electronic mail attachment. The possible effects of running the program are unpredictable.

16. When a system can be described as scalable in nature?

A system is described as scalable if it will remain effective when there is a significant increase in the number of resources and the number of users. Distributed systems operate effectively and efficiently at many different scales, ranging from a small intranet to the Internet.

17. What are the challenges faced by a scalable distributed system?

- **Controlling the cost of physical resources**: As the demand for a resource grows, it should be possible to extend the system, at reasonable cost, to meet it.
- **Controlling the performance loss:** Consider the management of a set of data whose size is proportional to the number of users or resources in the system.

- **Preventing software resources running out:** An example of lack of scalability is shown by the numbers used as Internet (IP) addresses
- **Avoiding performance bottlenecks:** In general, algorithms should be decentralized to avoid having performance bottlenecks.

18. What are the techniques used for dealing failures in a distributed system.

- ***Detecting failures*****:** Some failures can be detected. For example, checksums can be used to detect corrupted data in a message or a file.
- ***Masking failures*****:** Some failures that have been detected can be hidden or made less severe.
- ***Tolerating failures*****:** Most of the services in the Internet do exhibit failures – it would not be practical for them to attempt to detect and hide all of the failures that might occur in such a large network with so many components.
- ***Recovery from failures*****:** Recovery involves the design of software so that the state of permanent data can be recovered or 'rolled back' after a server has crashed.
- ***Redundancy*****:** Services can be made to tolerate failures by the use of redundant components.

19. How the availability of a system can be measured?

The availabilityof a system is a measure of the proportion of time that it is available for use. When one of the components in a distributed system fails, only the work that was using the failed component is affected. A user may move to another computer if the one that they were using fails; a server process can be started on another computer.

20. Define Transparency. What are its types?

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

Its types are:

- Access transparency
- Location transparency
- Concurrency transparency
- Replication transparency
- Failure transparency
- Mobility transparency
- Performance transparency
- Scaling transparency

21. What are the non-functional properties of a system that affects its quality of service?

The main nonfunctional properties of systems that affect the quality of the service experienced by clients and users are reliability, security and performance. Adaptability to meet changing system configurations and resource availability has been recognized as a further important aspect of service quality.

22. What are the main technological components of a web?

Then main technological components of a web are:

- **HyperText Markup Language (HTML)**, a language for specifying the contents and layout of pages as they are displayed by web browsers
- **Uniform Resource Locators (URLs)** also known as Uniform Resource Identifiers (URIs), which identify documents and other resources stored as part of the Web
- **Hypertext Transfer Protocol (HTTP)** is client-server system architecture, with standard rules for interaction by which browsers and other clients fetch documents another resource from web servers.

23. What is HTML and HTTP?

HTML:HTML stands for HyperText Markup Language. It is a well-known mark up language used to develop web pages. It has been around for a long time and is commonly used in webpage design. HTML is written using HTML elements, which consist of tags, primarily and opening tag and a closing tag.

HTTP: HTTP, on the other hand, stands for Hypertext Transfer Protocol. It is a means of data communication for the World Wide Web. It is an application protocol for distributed, collaborative, hypermedia information systems. HTTP is the protocol to exchange or transfer hypertext.

24. **Why HTTP called as request-reply protocol?**

In case of HTTP protocol, the client sends a request message to the server containing the URL of the required resource. The server looks up the path name and, if it exists, sends back the resource's content in a reply message to the client. Otherwise, it sends back an error response such as the familiar '404 Not Found'. Hence it is called as request-reply protocol.

UNIT II COMMUNICATION IN DISTRIBUTED SYSTEM

PART A

1. **What are the issues in distributed system?**
   - There is no global time in a distributed system, so the clocks on different computers do not necessarily give the same time as one another.
   - All communication between processes is achieved by means of messages. Message communication over a computer network can be affected by delays, can suffer from a variety of failures and is vulnerable to security attacks.
2. **What are the difficulties and threats for distributed systems?**
   - Widely varying modes of use
   - Wide range of system environments
   - Internal problems
   - External threats
   - Attacks on data integrity and secrecy
   - denial of service attacks
3. **What is a physical model?**

   A physical model is a representation of the underlying hardware elements of a distributed system that abstracts away from specific details of the computer and networking technologies employed.
4. **What is meant by Distributed systems of systems/ Ultra-Large-Scale (ULS) distributed systems?**

   A system of systems (mirroring the view of the Internet as a network of networks) can be defined as a complex system consisting of a series of subsystems that are systems in their own right and that come together to perform a particular task or tasks.
5. **What are the three generations of distributed systems?**
   - Early distributed systems
   - Internet-scale distributed systems
   - Contemporary distributed systems
6. **What is a Web Service?**

   The World Wide Web Consortium (W3C) defines a web service as a software application identified by a URI, whose interfaces and bindings are capable of being defined, described and discovered as XML artifacts. A Web service supports direct interactions with other software agents using XML-based message exchanges via Internet-based protocols.
7. **What is a cache?**

A cache is a store of recently used data objects that is closer to one client or a particular set of clients. When a new object is received from a server it is added to the local cache store, replacing some existing objects if necessary.

8. **Define Mobile Agent.**

A mobile agent is a running program (including both code and data) that travels from one computer to another in a network carrying out a task on someone's behalf, such as collecting information, and eventually returning with the results. A mobile agent may make many invocations to local resources at each site it visits.

9. **What are Thin clients?**

Thin client refers to a software layer that supports a window-based user interface that is local to the user while executing application programs or, more generally, accessing services on a remote computer.

10. **What is meant by Reflection?**

A pattern that is increasingly used in distributed systems as a means of supporting both introspection (the dynamic discovery of properties of the system) and intercession (the ability to dynamically modify structure or behaviour).

11. **What do you mean by Masking failures?**

A service masks a failure either by hiding it altogether or by converting it into a more acceptable type of failure. For an example of the latter, checksums are used to mask corrupted messages, effectively converting an arbitrary failure into an omission failure. Masking can be done by means of replication.

12. **Define 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. Thus marshalling consists of the translation of structured data items and primitive values into an external data representation. Unmarshalling is the process of disassembling them on arrival to produce an equivalent collection of data items at the destination. Thus, unmarshalling consists of the generation of primitive values from their external data representation and the rebuilding of the data structures.

13. **What is meant by XML (eXtensible Markup Language)?**
    - XML is a markup language that was defined by the World Wide Web Consortium (W3C) for general use on the Web.
    - It is a markup language that defines a set of rules for encoding documents in a format which is both human-readable and machine-readable.
    - XML data is known as self-describing or self-defining which means that the structure of the data is embedded with the data.
    - XML was designed for writing structured documents for the Web.
    - XML documents, being textual, can be read by humans.

14. **What do you mean by XML prolog?**
    - Every XML document must have a prolog as its first line. The prolog must at least specify the version of XML in use (which is currently 1.0).
    - **Example:** <?XML version = "1.0" encoding = "UTF-8" standalone = "yes"?>

15. **What are XML namespaces?**

An XML namespace is a set of names for a collection of element types and attributes that is referenced by a URL. Any element that makes use of an XML namespace can specify that namespace as an attribute called xmlns, whose value is a URL referring to the file containing the namespace definitions.

Example: xmlns:pers = http://www.cdk5.net/person

16. **What do you mean by XML schemas?**

An XML schema defines the elements and attributes that can appear in a document, how the elements are nested and the order and number of elements, and whether an element is empty or can include text.

17. **What is a Remote object reference?**

A remote object reference is an identifier for a remote object that is valid throughout a distributed system. A remote object reference is passed in the invocation message to specify which object is to be invoked. Remote object references must be generated in a manner that ensures uniqueness over space and time.

18. **What is a multicast operation?**

A multicast operation is the mechanism where an operation sends a single message from one process to each of the members of a group of processes, usually in such a way that the membership of the group is transparent to the sender.

19. **What is a overlay network?**

An overlay network is a virtual network consisting of nodes and virtual links, which sits on top of an underlying network (such as an IP network).

20. **What are the design issues for RPC.**

- 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.

21. **What is a service interface?**

The term service interface is used to refer to the specification of the procedures offered by a server, defining the types of the arguments of each of the procedures.

22. **What is an IDL?**

Interface Definition Language (IDL) is designed to allow procedures implemented in different languages to invoke one another. An IDL provides a notation for defining interfaces in which each of the parameters of an operation may be described as for input or output in addition to having its type specified.

23. **What do you mean by Maybe semantics?**

With maybe semantics, the remote procedure call may be executed once or not at all. Maybe semantics arises when no fault-tolerance measures are applied and can suffer from the following types of failure:

• Omission failures-If the request or result message is lost;
• Crash failures-When the server containing the remote operation fails.

24. **What is meant by At-least-once semantics?**

With At-least-once semantics, the invoker receives either a result, in which case the invoker knows that the procedure was executed at least once, or an exception informing it that no result was received.

25. **What is meant by At-most-once semantics?**

With At-most-once semantics, the caller receives either a result, in which case the caller knows that the procedure was executed exactly once, or an exception informing it that no result was received, in which case the procedure will have been executed either once or not at all.

26. **What is meant by Garbage collection?**

It is necessary to provide a means of freeing the space occupied by objects when they are no longer needed. A language such as Java, that can detect automatically when an object is no longer accessible recovers the space and makes it available for allocation to other objects. This process is called garbage collection.

27. **What do you mean by a Servant?**

A servant is an instance of a class that provides the body of a remote object. It is the servant that eventually handles the remote requests passed on by the corresponding skeleton. Servants live within a server process.

28. **What is meant by Group communication?**

Group communication offers a service whereby a message is sent to a group and then this message is delivered to all members of the group. In this action, the sender is not aware of the identities of the receivers.

29. **Define Closed and open groups.**

A group is said to be closed if only members of the group may multicast to it. A process in a closed group delivers to itself any message that it multicasts to the group. A group is open if processes outside the group may send to it.

30. **Define Overlapping and non-overlapping groups.**
In overlapping groups, entities (process-es or objects) may be members of multiple groups, and non-overlapping groups imply that membership does not overlap (that is, any process belongs to at most one group).

31. **What is meant by FIFO ordering?**
First-in-First-out (FIFO) ordering (also referred to as source ordering) is concerned with preserving the order from the perspective of a sender process, in that if a process sends one message before another, it will be delivered in this order at all processes in the group.

32. **What is meant by Causal ordering?**
Causal ordering takes into account causal relationships between messages, in that if a message happens before another message in the distributed system this so-called causal relationship will be preserved in the delivery of the associated messages at all processes.

33. **What do you mean by publish-subscribe system?**
A publish-subscribe system is a system where publishers publish structured events to an event service and subscribers express interest in particular events through subscriptions which can be arbitrary patterns over the structured events.

UNIT III PEER TO PEER SERVICES AND FILE SYSTEM

PART A

1. **What is the use of middleware?**
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.

2. **Write about the parts available in routing algorithm?**
   - Routing algorithm must make decisions that determine the route taken by each packet as it travels through the network. In circuit-switched network layers such as X.25 and frame relay networks such as ATM the route is determined whenever a virtual circuit or connection is established.
   - In packet-switched network layers such as IP it is determined separately for each packet, and the algorithm must be particularly simple and efficient if it is not to degrade network performance.
   - It must dynamically update its knowledge of the network based on traffic monitoring and the detection of configuration changes or failures. This activity is less time-critical; slower and more computation-intensive techniques can be used.

3. **Define multicast communication?**
It is the implementation of group communication. Multicast communication requires coordination and agreement. The aim is for members of a group to receive copies of messages sent to the group . Many different delivery guarantees are possible
Example: agreement on the set of messages received or on delivery ordering.

4. **What are the Application dependencies of Napster?**
Napster took advantage of the special characteristics of the application for which it was designed in other ways:
   - Music files are never updated, all the replicas of files need to remain consistent after updates.

- No guarantees are required concerning the availability of individual files – if amusic file is temporarily unavailable, it can be downloaded later. This reduces the requirement for dependability of individual computers and their connections tothe Internet.

5. **Define Routing overlay.**

   In peer-to-peer systems a distributed algorithm known as a routing overlay takes responsibility for locating nodes and objects. The name denotes the fact that the middleware takes the form of a layer that is responsible for routing requests from any client to a host that holds the object to which the request is addressed.

6. **What is a file group?**

   A collection of files that can be located on any server or moved between servers while maintaining the same names is a file group. Simillar to a UNIX file system helps with distributing the load of file serving between several servers.File groups have identifiers which are unique throughout the system used to refer file groups and files

7. **What is flat file service interface?**

   It is RPC interface used by client modules. It is not normally used directly by user level programs. A field is invalid if the file that it refers to is not present in the server processing the request or if its access permissions are inappropriate for the operation requested.

8. **Write a note on Andrew file system?**

   AFS provides transparent access to remote shared files for unix programs running on workstations. Access to AFS files is via the normal unix file primitives, enabling existing unix programs to access AFS files without modification or recompilation.

9. **Write a note on X.500 directory service?**

   It is a directory service. It can be in the same way as a conventional name service but it is primarily used to satisfy descriptive queries, designed to discover the names and attributes of other users or system resources.

10. **What is the use of iterative navigation?**

    DNS supports the model known as iterative navigation. To resolve a name, a client presents the name to the local name server, which attempts to resolve it. If the local name server has the name, it returns the result immediately.

11. **Define multicast navigation?**

    A client multicast the name to be resolved and the required object type to the group of name servers. Only the server that holds the named attributes responds to the request.

12. **What are the major goals of Sun NFS?**

    - NFS should be deployable easily.
    - NFS should be efficient enough to be tolerable to users.
    - Sun NFS should work with existing applications.
    - To achieve a high level of support for hardware andoperating system heterogeneity.

13. **What is a Name Service?**

    A name service stores information about a collection of textual names, in the form of bindings between the names and the attributes of the entities they denote, such as users, computers, services and objects. The collection is often subdivided into one or more naming contexts: individual subsets of the bindings that are managed as a unit. The major operation that a name service supports is to resolve a name that is, to look up attributes from a given name.

14. **Define Namespace.**

    A name space is the collection of all valid names recognized by a particular service. The service will attempt to look up a valid name, even though that name may prove not to correspond to any object that is to be unbound. Name spaces require a syntactic definition to separate valid names from invalid names. For example, **'...'** is not acceptable as the DNS name of a computer, whereas www.cdk.net is valid.

15. Illustrate the importance of Caching.

Caching is key to a name service's performance and assists in maintaining the availability of both the name service and other services in spite of name server crashes. Its role in enhancing response times by saving communication with name servers is clear. Caching can be used to eliminate high-level name servers – the root server in particular.

16. Define DNS with examples

The Domain Name System is a name service design whose main naming database is used across the Internet. The objects named by the DNS are primarily computers – for which mainly IP addresses are stored as attributes.

The original top-level organizational domains called as *generic domains* in use across the Internet were:

- *com* – Commercial organizations
- *edu* – Universities and other educational institutions
- *gov* – US governmental agencies
- *mil* – US military organizations
- *net* – Major network support centres
- *org* – Organizations not mentioned above
- *int* – International organizations

17. Write short notes on Directory Services.

A service that stores collections of bindings between names and attributes and that looks up entries that match attribute-based specifications is called a directory service. Examples are Microsoft's Active Directory Services, X.500 etc. Directory services are sometimes called yellow pages services, and conventional name services are correspondingly called white pages services, in an analogy with the traditional types of telephone directory. Directory services are also sometimes known as attribute-based name services.

18. Write about LDAP.

The Lightweight Directory Access Protocol (LDAP) is a directory service protocol that runs on a layer above the TCP/IP stack. It provides a mechanism used to connect to, search, and modify Internet directories. The LDAP directory service is based on a client-server model. It is an open, vendor-neutral, industry standard application protocol.

19. What are the non-functional requirements that peer-to-peer middleware must address?

- ***Global scalability***: One of the aims of peer-to-peer applications is to exploit the hardware resources of very large numbers of hosts connected to the Internet
- ***Load balancing***: The performance of any system designed to exploit a large number of computers depends upon the balanced distribution of workload across them.
- ***Optimization for local interactions between neighboring peers.*** The middleware should aim to place resources close to the nodes that access them the most.
- ***Accommodating to highly dynamic host availability***: Most peer-to-peer systems are constructed from host computers that are free to join or leave the system at any time.

20. What is the key problem faced in peer-to-peer middleware.

A key problem in the design of peer-to-peer applications is providing a mechanism to enable clients to access data resources quickly and dependably wherever they are located throughout the network. Napster maintained a unified index of available files for this purpose, giving the network addresses of their hosts.

21. What are the characteristics of peer-to-peer systems?

- Its design ensures that each user contributes resources to the system.
- Although they may differ in the resources that they contribute, all the nodes in a peer-to-peer system have the same functional capabilities and responsibilities.
- Its correct operation does not depend on the existence of any centrally administered systems.
- A key issue for their efficient operation is the choice of an algorithm for the placement of data across many hosts and subsequent access to it in a manner that balances the workload and ensures availability without adding undue overheads.

22. **What is the use of GUID?**

A **Globally Unique Identifier** (**GUID**) is a unique reference number used as an identifier in computer software. The term GUID typically refers to various implementations of the universally unique identifier (UUID) standard. A GUID can be stored as a 16-byte (128-bit) number. GUIDs are commonly used as the primary key of database tables.

UNIT IV SYNCHRONIZATION AND REPLICATION

PART A

1. **What is clock synchronization?**

Nodes in distributed system to keep track of current time for various purposes such as calculating the time spent by a process in CPU utilization ,disk I/O etc so that the corresponding user can be charged. Clock synchronization means the time difference between two nodes should be very small.

2. **What do you mean by clock skew and clock drift?**

- **Clock skew** – Instantaneous difference between the readings of any two clocks is called clock skew. Skew occurs since computer clocks like any others tends not be perfect at all times.
- **Clock drift** – Clock drift occurs in crystal based clocks which counts time at different rates and hence they diverge. The drift rate is the change in the offset between the clock and a nominal perfect reference clock per unit of time measured by the reference clock.

3. **What do you mean by Coordinated Universal Time?**

Coordinated Universal Time generally abbreviated as UTC is an international standard for timekeeping. It is based on atomic time. UTC signals are synchronized and broadcast regularly from land based radio stations and satellites covering many parts of the world.

4. **Define External Synchronization.**

Generally it is necessary to synchronize the processes' clocks C_i with an authoritative external source of time. It is called as External Synchronization. For a synchronization bound D>0, and for a source S of UTC time, $| S(t) – C_i(t)|<D$ for i=1,2..N for all real times t in I where I is the time interval.

5. **When an object is considered to be garbage?**

An object is considered to be garbage if there are no longer any references to it anywhere in the distributed system. The memory taken up by the object can be reclaimed once it is known to be garbage. The technique used here is distributed garbage collection.

6. **What do you meant by Distributed debugging?**

In general, distributed systems are complex to debug.A special care needs to be taken in establishing what occurred during the execution. Consider an application with a variable x_i(i=1,2..N) and the variable changes as the program executes but it is always required to be within a value $ of one other. In that case, relationship must be evaluated for values of the variables that occur at the same time.

7. **Define marker receiving rule.**

Snapshot algorithm designed by Chandy and Lamport is used for determining global states of distributed systems. This algorithm is defined through two rules namely marker sending rule and marker receiving rule. Marker receiving rule obligates a process that has not recorded its state to do so.

8. **Define marker sending rule.**

Snapshot algorithm designed by Chandy and Lamport is used for determining global states of distributed systems. This algorithm is defined through 2 rules namely marker sending rule and marker receiving rule. Marker sending rule obligates processes to send a marker after they have recorded their state ,but before they send any other messages.

9. **Define total ordering?**

Common ordering requirements are important in case of multicast approach. General ordering requirements are total ordering, FIFO ordering and causal ordering. Total ordering is the case where if a correct process delivers message m before it delivers m', then any other correct process that delivers m' will deliver m before m'.

10. **Name any two election algorithms.**

An algorithm for choosing a unique process to play a particular role is called an election algorithm.Generally used election algorithms are:

- Ring based election algorithm
- Bully algorithm

11. **What do you mean by atomic transaction?**

A transaction is said be an atomic transaction if it follows All or Nothing property according to which a transaction either completes successfully in which case the effects of all of its operations are recorded in the objects or has no effect at all. In short atomic transaction is a transaction that happens completely or not at all. It does not produce partial results.

For eg: ATM machine hands you cash and deducts amount from your account or does not have any effect at all.

12. **What are the ACID properties of a transaction?**

- **Atomicity** : A transaction must be all or nothing.
- **Consistency** : A transaction takes the system from one consistent state to another consistent state.
- **Isolation**: The isolation property ensures that the concurrent execution of transactions results in a system state that would be obtained if transactions were executed serially, i.e., one after the other.
- **Durability:**Durability means that once a transaction has been committed, it will remain so, even in the event of power loss, crashes, or errors.

13. **Define the characteristics of serial equivalent transactions**.

For any pair of transactions, it is possible to determine the order of pairs of conflicting operations on objects accessed by both of them. Read and write are the operations generally considered. For two transactions to be serially equivalent it is necessary and sufficient that all pairs of conflicting operations of the two transactions be executed in the same order at all of the objects they both access.

14. **What are the advantages of nested transactions?**

The outermost transaction in a set of nested transactions is called top level transaction. Transactions other than the top level transaction are called subtransactions.

Advantages of nested transactions are:

- Subtransactions at one level may run concurrently with other subtransactions at the same level in the hierarchy. This can allow additional concurrency in a transaction.
- Subtransactions can commit or abort independently.

15. **What are the rules of committing nested transactions?**

Rules for committing of nested transactions are:

- A transaction may commit or abort only after its child transactions have completed.
- When a subtransaction completes, it makes an independent decision either to commit provisionally or to abort.
- When a parent aborts, all of its transactions are aborted.
- When a subtransaction aborts, the parent can decide whether to abort or not.

16. **Write short notes on strict two phase locking**

A simple mechanism of a serializing mechanism is the use of exclusive locks. Under a strict execution regime, a transaction that needs to read or write an object must be delayed until other transactions that wrote the same object have committed or aborted. To enforce this rule, any locks applied during the progress of a transaction are held until the transaction commits or aborts. This is called *strict two-phase locking*. The presence of the locks prevents other transactions reading or writing the objects.

17. **What are the drawbacks of locking?**

Drawbacks of locking mechanism are:

- Lock maintenance represents an overhead that is not present in systems that do not support concurrent access to shared data.
- The use of locks can result in deadlock in some cases.
- To avoid cascading aborts, locks cannot be released until the end of the transaction. It significantly reduces the potential for accuracy.

18. **Define the approach of two phase commit protocol.**

Two phase commit protocol is designed to allow any participant to abort its part of a transaction. In the first phase of the protocol, each participant votes for the transaction to be committed or aborted. In the second phase of the protocol, every participant in the transaction carries out the joint decision.

19. **How is distributed dead lock detected?**

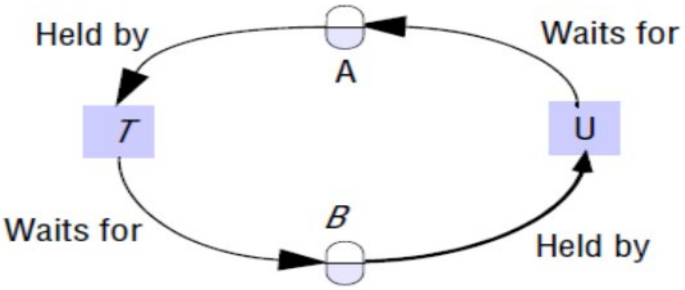
In a distributed system involving multiple servers being accessed by multiple transactions, a global wait-for-graph has to be constructed. If and only if there is a cycle in the wait-for-graph a distributed deadlock is said to be detected. Hence Detection of a distributed deadlock requires a cycle to be found in the global transaction wait-for graph that is distributed among the servers that were involved in the transactions.

20. **What is a phantom deadlock?**

A deadlock that is ‘detected’ but is not really a deadlock is called a phantom deadlock. In distributed system if there is a deadlock, the necessary information will eventually be collected in one place and a cycle will be detected. As this procedure will take some time, there is a chance that one of the transactions that holds a lock will meanwhile have released it, in which case the deadlock will no longer exist. This is a sample case for phantom deadlocks.

21. **What is wait-for-graph?**

A wait-for graph can be used to represent the waiting relationships between current transactions. In a wait-for graph the nodes represent transactions and the edges represent wait-for relationships between transactions. Following figure represents a wait-for-graph with transactions T and U.



22. **Define Edge chasing**

A distributed approach to deadlock detection uses a technique called edge chasing or path pushing. In this approach, the global wait-for graph is not constructed, but each of the servers involved has knowledge about some of its edges. The servers attempt to find cycles by forwarding messages called probes, which follow the edges of the graph throughout the distributed system

23. **What is the role of replication in distributed systems?**

Replication is defined as the maintenance of copies of data at multiple computers. It is a key to the effectiveness of distributed systems in that it can provide enhanced performance, high availability and fault tolerance.

UNIT – V PROCESS & RESOURCE MANAGEMENT

PART - A

1. **What is process migration?**

Process migration is the relocation of a process from its current location (source node) to another node(destination node).The process can be either a non-preemptive or preemptive process. Selection of the process to be migrated, selection of the destination node and the actual transfer of the selected process are the three steps involved in process migration.

2. **What are the advantages of process migration?**

Various advantages of process migration are:

- Reduces average response time of processes.
- Speeds up individual jobs.
- Gains higher throughput.
- Effective utilization of resources and reduces network traffic.

3. **What are the activities involved in process migration?**

Migration of a process is a complex activity that involves many sub-activities. They are:

1. Freezing the process on its source node and restarting it on its destination node.
2. Transferring the process's address space from its source node to its destination node.
3. Forwarding messages meant for the migrant process.
4. Handling communication between cooperating processes that have been separated as a result of process migration.

4. **Mention the levels of transparency in process migration.**

Transparency is an important requirement for a system that supports process migration.
The two levels of transparency are:

- **Object access level**: Minimum requirement for a system to support non-preemptive process migration.
- **System call and interprocess communication level**: This facility is required to support preemptive process migration.

5. **What is Threads?**

Threads are an efficient way to improve application performance through parallelism. Each thread of a process has its own program counter,its own register states, and its own stack.But all the threads shares the same address space.Threads are often referred to as lightweight process.

6. **What are the main advantages of using threads instead of multiple processes?**

Threads has its own program counter,its own register states, and its own stack but shares the same address space
Advantages of threads over multiple processes are:

- **Context Switching:** Threads are very inexpensive to create and destroy, and they are inexpensive to represent. For eg: they require space to store, the PC, the SP, and the general-purpose registers, but they do not require space to share memory information, Information about open files of I/O devices in use, etc. In other words, it is relatively easier for a context switch using threads.
- **Sharing**: Treads allow the sharing of a lot resources that cannot be shared in process, for example, sharing code section, data section, Operating System resources like open file etc

7. **Mention the models used to organize the threads of a process.**

- **Dispatcher-workers model** – In this model, process consists of a single dispatcher thread and multiple worker threads.
- **Team model** – In this model, all threads behave as equals and there is no dispatcher-worker relationship for processing client's requests.
- **Pipeline model** – In this model, the threads of a process are organized as a pipeline where the output data from one thread is used for processing by the other threads.

8. **Define critical region.**

A segment of code in which a thread may be accessing some shared variable is called critical region. Multiple threads should not access the same data simultaneously. Hence the execution of critical regions in which the same data is accessed by the threads must be mutually exclusive in time.

9. **Define mutex variable.**

Mutex variable is like a binary semaphore that is always in one of the two states locked or unlocked. Mutex variables are used to implement mutual exclusion technqiues. A thread that wants to execute in a critical region performs a lock operation over the mutex variable which has to be in unlocked state.

10. **Mention some library procedures for managing the threads.**

Some of the library procedures for managing threads are:

- pthread_create – Creates a new thread in the same address space as the calling thread.
- pthread_exit – Terminates the calling thread.
- pthread_join – It makes the calling thread to block itself and waits until thread specified in the routine's argument terminates.
- pthread_detach – Used by the parent thread to disown a child thread.
- pthread_cancel – Used by a thread to kill another thread.

11. **Mention the types of mutex variables.**

Generally the types of mutex variables supported are:

- Fast – Fast mutex variable causes a thread to block when the thread attempts to lock an already locked mutex variable.
- Recursive – It allows a thread to lock an already locked mutex variable.
- Nonrecursive – It neither allows a thread to lock an already locked mutex variable nor causes the thread to block.

12. **Write short notes on resource management.**

Distributed systems are characterized by resource multiplicity and system transparency. Efficient resource management is implemented by a resource manager. The resource manager schedules the processes to make use of the system resources in such a manner that resource usage,resource time,network congestion and scheduling overhead are optimized.

13. **What are the features of global scheduling algorithm?**

Desirable features of a good global scheduling algorithm are:

- No a priori knowledge about the processes to be executed by the scheduling algorithm.
- Dynamic in nature.
- Quick decision making capability.
- Balanced system performance and scheduling overhead.
- Stablility
- Scalability
- Fault tolerance
- Fairness of service

14. **What is Task assignment approach?**

Task assignment approach is the technique used for scheduling processes of a distributed system.In this approach each process submitted by a user for processing is viewed as a collection of related tasks and these tasks are scheduled to suitable nodes so as to improve performance

15. **Define Load balancing approach.**

Load balancing algorithm or load leveling algorihm aims for better resource utilization. Load balancing approach tries to balance the total system load by transparently transferring the workload from heavily loaded nodes to lightly loaded nodes in order to ensure overall good performance.

16. **What are the issues in designing load balancing algorithm.**

Designing a good load balancing algorithm is a difficult task because of the following issues:

- Load estimation policies – Estimates the workload

- Process transfer policy – Determines whether to execute a process locally or remotely.
- State information exchange policy – Deals about the load information exchange.
- Location policy – Selects the node to which the process to be sent.
- Priority assignment policy – Determines the priority of execution of local and remote process.
- Migration limiting policy – Decides the number of times a process can migrate.

17. What is load-sharing approach?

Load sharing approach also aims for efficient resource utilization. It attempts to ensure that no node is idle while processes wait for service at some other node. In this approach it is sufficient to know whether a node is busy or idle. Load sharing approach does not attempt to balance the average workload on all the nodes of the system.

18. State the differences between the static and dynamic load balancing alogrithms.

Static load balancing algorithms	Dynamic load balancing algorithms
It uses only information about the average behavior of the system ignoring the current state of the system.	It reacts to the system state that changes dynamically.
Potential of static algorithms is limited	It provides greater performance benefits.
These algorithms are simpler.	These algorithms are complex in nature

19. State the differences between the deterministic and probabilistic load balancing alogrithms.

Deterministic load balancing algorithms	Probabilistic load balancing algorithms
These algorithms use the information about the properties of the nodes and the characteristics of the processes to be scheduled.	These algorithms use the information about the static attributes of the system such as number of nodes, network topology etc.
This approach provides optimized performance.	It often suffers from having poor performance.
It costs more to implement.	It is easier to implement.

20. What is threshold?

Most of the load balancing algorithm use threshold policy to decide whether a node is lightly or heavily loaded. Threshold value of a node is the limiting value of its workload. A new process at a node is accepted locally for processing if the workload of the node is below its threshold value at that time. Threshold value is determined either by static or dynamic policy.

21. What is static policy and dynamic policy?

- **Static policy**: In this method, each node has a predefined threshold value on its processing capability. Main advantage of this method is that no exchange of state information among the nodes is required in deciding the threshold value.
- **Dynamic policy**: In this method, the threshold value of a node is calculated as a product of the average workload of all the nodes and a predefined constant. It gives a more realistic value for the threshold value based on the information exchange among the nodes.

22. Write the priority assignment policies.

Priority assignment policies need to be devised in order to implement process migration in a distributed operating system. It can be any one of the following:

- Selfish – Local processes given higher priority.
- Altruistic – Remote processes given higher priority.

- Intermediate: Priority of the processes depends on the number of local processes and number of remote processes at concerned node.

23. Write about sender-initiated location policy.

Sender-initiated location policy is a location policy where heavily loaded nodes search for lightly loaded nodes to which work may be transferred. In this method, the node either broadcasts the message or randomly probes the other nodes to find whether it is a lightly loaded node to accept one or more processes.

24. What is receiver-initiated policy?

In this location policy, lightly loaded nodes search for heavily loaded nodes from which work may be transferred . When the load's value of a node falls below the threshold value, it broadcasts a message indicating its willingness to receive processes for executing or randomly probes the other nodes to search to find heavily loaded nodes.