

**UNIT I**  
**INTRODUCTION**

**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 Name five reasons why to build distributed system. (NOV 2016)**

The reasons to build distributed system are:

- Making Resources Accessible
- Distribution Transparency
- Openness
- Scalability
- Failure Handling

**3 Discuss the design issues in Intranet. (NOV 2016)**

To build an integrated network (an *internetwork*) many subnets must be integrated, each of which is based on different network technologies. To make this possible, the following are needed:

1. A unified internetwork addressing scheme that enables packets to be addressed to any host connected to any subnet;
2. A protocol defining the format of internetwork packets and giving rules according to which they are handled;
3. Interconnecting components that route packets to their destinations in terms of internetwork addresses, transmitting the packets using subnets with a variety of network technologies.

**4 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.

**5 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.

**6 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, fibre optic cables and other high-bandwidth circuits.

**7 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.

8 **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.

9 **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.

10 **What does the term remote invocation mean?**

In client – server approach, requests are sent in messages from clients to a server and replies are sent in messages from the server to the clients. When the client sends a request for an operation to be carried out, we say that the client invokes an operation upon the server. A complete interaction between a client and a server, from the point when the client sends its request to when it receives the server's response, is called a remote invocation.

11 **What are the challenges of distributed systems?** The main challenges of distributed system are:

- Heterogeneity       Openness       Security
- Scalability
- Failure handling       Concurrency       Transparency       QoS

12 **What is the role of middleware?**

**Name some services and examples of Middleware. (JUNE 2016)**

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. Examples: The Common Object Request Broker (CORBA), Java Remote Method Invocation (Java RMI)

13 **What is the role of Proxy Server and mobile code? (JUNE 2016)**

Proxy servers provide a shared cache of web resources for the client machines at a site or across several sites. The purpose of proxy servers is to increase the availability and performance of the service by reducing the load on the wide area network and web servers. . it also can be used to access remote web servers through a firewall.

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. Examples are Java applets. The need for mobile code are 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. The virtual machine approach provides a way of making code executable on a variety of host computers.

14 **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 re-implemented 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.

15 **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.

16 **What are the two security challenges that are not fully met by distributed systems? List the limitations of distributed system. (JUNE 2016)**

- 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.
- No global clock**

17 **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.

18 **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.

19 **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.

20 **How the availability of a system can be measured?**

The *availability* of 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.

### UNIT-I / PART-B

- 1 **Explain in detail about the real time examples of DS.**
- 2 **Explain briefly about various trends in distributed systems.**
- 3 **Explain few examples of distributed systems.**
- 4 **Briefly explain about how distributed computing serves as utility**
- 5 **Explain about the scalability and failure handling in distributed system.**

## UNIT II COMMUNICATION IN DISTRIBUTED SYSTEM

### UNIT-II / PART-A

6 **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.
  
- 7 What are the difficulties and threats for distributed systems?** The difficulties and threats for distributed systems are:
 

|                                          |                           |
|------------------------------------------|---------------------------|
| Widely varying modes of use environments | Wide range of system      |
| 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 "C for example, accessing individual database entries.
  
- 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 **Define Inter-process Communication. (JUNE 2016)**

Inter-process communication deals with the characteristics of protocols for communication between processes in distributed systems. Inter-process communication in the Internet provides both datagram and stream communication.

12 **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.

13 **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.

14 **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. XML was designed for writing structured documents for the Web. XML documents, being textual, can be read by humans.

15 **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"?>`

16 **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`

17 **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.

18 **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.

19 **Write the characteristics of multicast communication.**

(NOV 2016) The characteristics of multicast communication are:

- Fault tolerance based on replicated services
- Discovering services in spontaneous networking
- Better performance through replicated data
- Propagation of event notifications

20 **Define network virtualization. (NOV 2016)**

Network virtualization is concerned with the construction of many different virtual networks over an existing network such as the Internet. Each virtual network can be designed to support a particular distributed application. For example, one virtual network might support multimedia streaming, and coexist with another that supports a multiplayer online game, both running over the same underlying network.

An application-specific virtual network can be built above an existing network and optimized for that particular application, without changing the characteristics of the underlying network.

**UNIT-II / PART-B**

- 1 **Explain the various architectural elements and architectural pattern**
- 2 **Explain the various types of fundamental model**
- 3 **Explain the API for the internet protocol**
- 4 **Briefly explain the external data representation and multicast communication**
- 5 **Explain implementation of RPC and RMI**

**UNIT III**

**PEER TO PEER SERVICES AND FILE SYSTEM**

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 has two parts:**

- It 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:. agree 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, avoiding any need to make sure all the replicas of files remain consistent after updates.



No guarantees are required concerning the availability of individual files – if a music file is temporarily unavailable, it can be downloaded later. This reduces the requirement for dependability of individual computers and their connections to the 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. Similar 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 **List the file accessing models. (NOV 2016)**

The file accessing models of a distributed file system mainly depends on two factors-the methods used for accessing remote files and unit of data access:

- Accessing remote files
  - Remote service model
  - Data-caching model
- Unit of Data Transfer
  - File-level transfer model
  - Block-level transfer model
  - Byte-level transfer model
  - Record-level transfer model

11 **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.

12 **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.

13 **What are the major goals of Sun NFS?**

An industry standard for file sharing on local networks since the 1980s. An open standard with clear and simple interfaces. Closely follows the abstract file service model. Supports many of the design requirements already mentioned such as Transparency, Heterogeneity ,Efficiency and Fault tolerance

14 **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.

15 **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 eg, ‘...’ is not acceptable as the DNS name of a computer, whereas www.cdk.net is valid.

16 **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.

17 **Give the advantages in using name caches in file systems. (NOV 2016)**

Distributed name cache can have a substantial positive effect on distributed system performance. This is mainly due to the following characteristics of name service related activities:

- High degree of locality of name lookup
- Slow update of name information database
- On-use consistency of cached information is possible

18 **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. Eg 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.

19 **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

20 **Discuss on LDAP. (JUNE 2016)**

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. The LDAP is an open, vendor-neutral, industry standard application protocol.

**UNIT-III / PART-B**

- 1 **Explain in detail about Sun network file system?**
- 2 **Explain in detail about Andrew file system**
- 3 **Explain in detail about X.500 directory service**
- 4 **Briefly explain about the name services**
- 5 **Explain in detail about Napster and its legacy.**

**UNIT IV**  
**SYNCHRONIZATION AND REPLICATION**

**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 is clock's drift rate? (NOV 2016)**

A clock's drift rate is the change in the offset (difference in reading) between the clock and a nominal perfect reference clock per unit of time measured by the reference clock. For ordinary clocks based on a quartz crystal this is about  $10^{-6}$  seconds/second, giving a difference of 1 second every 1,000,000 seconds, or 11.6 days.

**4 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.

**5 Distinguish between physical clock and logical clock. (JUNE 2016)**

**Physical Clock**

These clocks are electronic devices that count oscillations occurring in a crystal at a definite frequency, and typically divide this count and store the result in a counter register. Clock devices can be programmed to generate interrupts at regular intervals thus time-slicing can be implemented.

The operating system reads the node's hardware clock value  $H_i(t)$ .

**Logical Clock**

A logical clock is a mechanism for capturing chronological and causal relationships in a distributed system. Distributed system may have no physically synchronous global clock, so a logical clock allows global ordering on events from different processes in such systems.

The hardware clock value is scaled and added an offset to get the software clock value  $C_i(t) = \alpha H_i(t) + \beta$ .

6 **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  $I$ .

7 **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.

8 **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  $\delta$  of one other. In that case, relationship must be evaluated for values of the variables that occur at the same time.

9 **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 2 rules namely marker sending rule and marker receiving rule. Marker receiving rule obligates a process that has not recorded its state to do so.

10 **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.

11 **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  $m'$ .

12 **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 and Bully algorithm.

13 **What do you mean by atomic transaction?**

A transaction is said to 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.

14 **Write the Happened-before relation. (JUNE 2016)**

The happened-before relation, denoted by  $\square$  can be defined as follows:

HB1: If  $\square$  process  $p_i$ :  $e \square_i e'$ , then  $e \square e'$ .

HB2: For any message  $m$ ,  $\text{send}(m) \square \text{receive}(m)$ , where  $\text{send}(m)$  is the event of sending the message, and  $\text{receive}(m)$  is the event of receiving it.

HB3: If  $e, e', e''$  are events such that  $e \square e'$  and  $e' \square e''$ , then  $e \square e''$ .

15 **What are the ACID properties of a transaction? ACID properties that a transaction should follow are:**

- 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.

16 **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.

17 **Define nested transactions. (NOV 2016)**

Nested transactions extend the above transaction model by allowing transactions to be composed of other transactions. Thus several transactions may be started from within a transaction, allowing transactions to be regarded as modules that can be composed as required.

The outermost transaction in a set of nested transactions is called the top-level transaction. Transactions other than the top-level transaction are called subtransactions. A subtransaction appears atomic to its parent with respect to transaction failures and to concurrent access. Subtransactions at the same level, can run concurrently, but their access to common objects is serialized by the locking scheme.

18 **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.

19 **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.

20 **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. preceding overlapping transactions, those that entered the validation phase before it.

#### UNIT-IV/ PART-B

- 1 **Explain in detail about internal and external synchronization of physical clock.**
- 2 **Explain in detail about Network time protocol**
- 3 **Explain the Chandy and Lamports snapshot algorithm for determining the global states of distributed systems**
- 4 **Briefly explain about any three election algorithm**
- 5 **Explain in detail about distributed mutual exclusion with suitable algorithm**

#### UNIT V *PROCESS & RESOURCE MANAGEMENT*

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?**

(JUNE 2016) There are various advantages of process migration. They are:

- Reduces average response time of processes.
- Speeds up individual jobs.
- Gains higher throughput.
- Effective utilization of resources.
- Reducing network traffic.
- Improves system security and system reliability.

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

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

- Freezing the process on its source node and restarting it on its destination node.
- Transferring the process's address space from its source node to its destination node.
- Forwarding messages meant for the migrant process.
- 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 inter-process 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 **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.

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 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 techniques. 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.

- 9 **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:** Threads 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
- 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 **Write down the goals to achieve an optimal assignment.**  
(NOV 2016) The goals to achieve an optimal assignment:
- Minimization of IPC costs.

- Quick turnaround time for the complete process.
- A high degree of parallelism
- Efficient utilization of system resources

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 algorithm 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 **List the features of scheduling algorithms. (NOV 2016)** 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.
- Stability
- Scalability
- Fault tolerance
- Fairness of service

17 **List the issues in designing load balancing algorithm. (JUNE 2016)**

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.

18 **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.

**19 State the differences between the static and dynamic load balancing algorithms.**

**Static load balancing algorithms**

It uses only information about the average behavior of the system ignoring the current state of the system.

Potential of static algorithms is limited  
These algorithms are simpler.

**Dynamic load balancing algorithms**

It reacts to the system state that changes dynamically.

It provides greater performance benefits.  
These algorithms are complex in nature

**20 State the differences between the deterministic and probabilistic load balancing algorithms.**

**Deterministic load balancing algorithms**

These algorithms use the information about the properties of the nodes and the characteristics of the processes to be scheduled.

This approach provides optimized performance.

It costs more to implement.

**Probabilistic load balancing algorithms**

These algorithms use the information about the static attributes of the system such as number of nodes, network topology etc.

It often suffers from having poor performance.

It is easier to implement.

**UNIT-V/ PART-B**

- 1 **Explain in detail about Process migration with suitable example**
- 2 **Explain in detail about address transfer mechanism**
- 3 **Explain in detail about process migration message forwarding mechanism**
- 4 **Briefly explain about thread models and its implementation**
- 5 **Explain in detail about good global scheduling algorithm**