BE - SEMESTER-VII (NEW) – WINTER 2021
Subject Code:3170719
Date:29/12/2021
Subject Name: Distributed System

Q.1

(a)What is a distributed system? How a distributed system does project a single system image?03

Ans: A distributed system is one, in which components located od n/w computers communicate & co-ordinate their actions only by passing messages.

- A distributed system is collection of independent entities that cooperate to solve a problem that cannot be solved individually.
- To support heterogeneous, computers & networks while offering a single system view, distributed system are often organized by means of a layer of software that is logically placed between placed between higher level consisting of users & applications & layer underneath consisting of OS.
- Middleware is software which lies between an OS and the applications running on it. Distributed System is also called as Middleware.
- An example: The world wide Web, where there are multiple components under the hood that help browsers display content but from user's point of view, all browser. are doing is accessing the web via a browser.
- Users & applications can interact with a distributed system in a consistent & uniform way, regardless of where & when interaction takes place.

(b)Briefly discuss the issues related to distributed system design.04

Design issues of distributed system –

1. **Heterogeneity:** Heterogeneity is applied to the network, computer hardware, operating system and implementation of different developers. A key component of the heterogeneous distributed system client-server environment is middleware. Middleware is a set of services that enables application and end-user to interact with each other across a heterogeneous distributed system.

2. **Openness**: The openness of the distributed system is determined primarily by the degree to which new resource-sharing services can be made available to the users.

3. **Scalability**: Scalability of the system should remain efficient even with a significant increase in the number of users and resources connected.

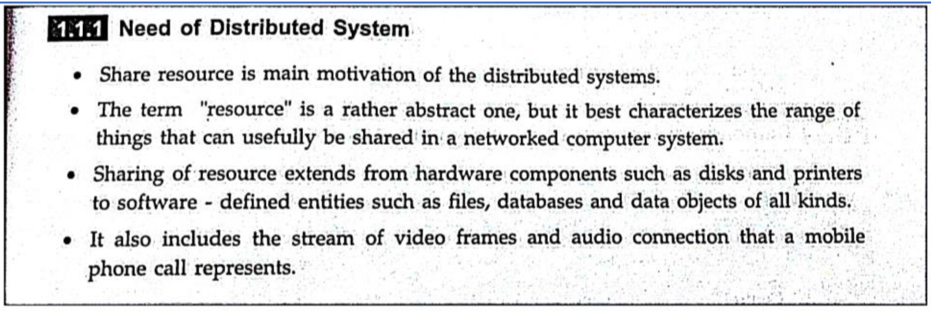
4. **Security**: Much of the info shared on DS is of high intrinsic value to the user. Security of information system has three components confidentially, integrity and availability.

5. **Failure Handling**: Failure handling is difficult in distributed systems because the failure is partial i.e. some components fail while others continue to function.

6. **Concurrency**: There is a possibility that several clients will attempt to access a shared resource at the same time i.e. read, write, and update. Each resource must be safe in a concurrent environment i.e. a distributed system must ensure that it operates correctly in a concurrent environment.

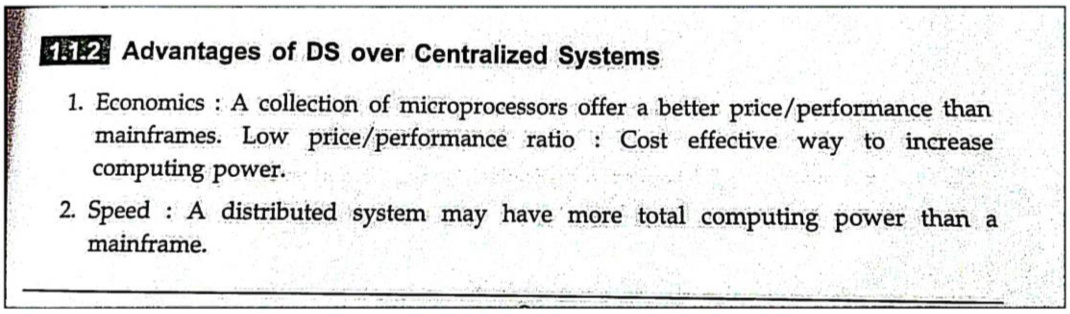
7. **Transparency:** Transparency ensures that the distributes system should be perceived as a single entity by the users or the application programmers rather than the collection of autonomous systems, which is cooperating. The user should be unaware of where the services are located and the transferring from a local machine to a remote one should be transparent.

(c)What is main motivation of distributed system? Explain advantages and disadvantages of distributed systems.



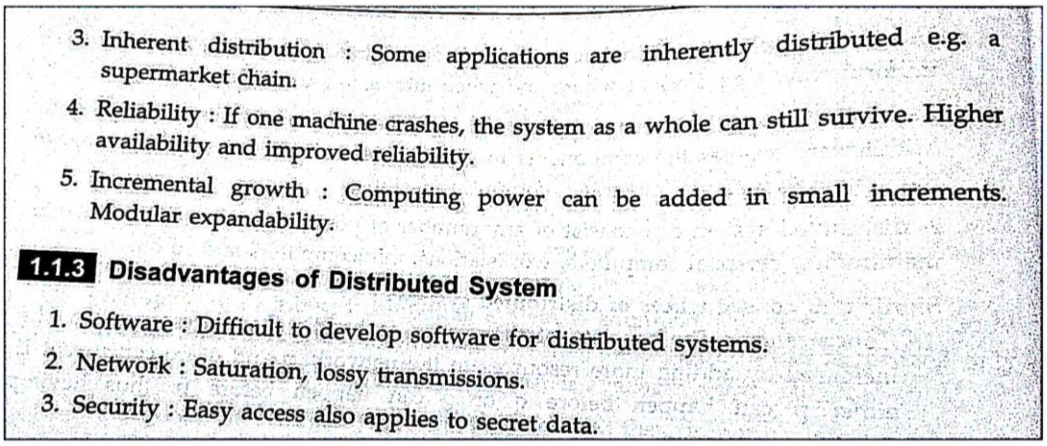
1.1.1 Need of Distributed System

- Share resource is main motivation of the distributed systems.
- The term "resource" is a rather abstract one, but it best characterizes the range of things that can usefully be shared in a networked computer system.
- Sharing of resource extends from hardware components such as disks and printers to software - defined entities such as files, databases and data objects of all kinds.
- It also includes the stream of video frames and audio connection that a mobile phone call represents.



1.1.2 Advantages of DS over Centralized Systems

1. Economics : A collection of microprocessors offer a better price/performance than mainframes. Low price/performance ratio : Cost effective way to increase computing power.
2. Speed : A distributed system may have more total computing power than a mainframe.



3. Inherent distribution : Some applications are inherently distributed e.g. a supermarket chain.
4. Reliability : If one machine crashes, the system as a whole can still survive. Higher availability and improved reliability.
5. Incremental growth : Computing power can be added in small increments. Modular expandability.

1.1.3 Disadvantages of Distributed System

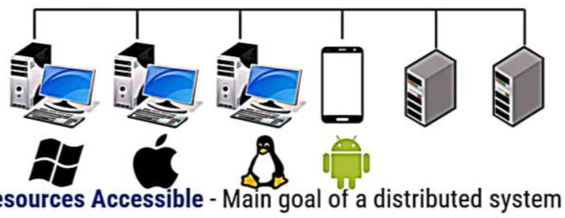
1. Software : Difficult to develop software for distributed systems.
2. Network : Saturation, lossy transmissions.
3. Security : Easy access also applies to secret data.

Q.2(a)List out the various characteristics of distributed system.

- **The relative simplicity of the software** - Each processor has a dedicated function.
- **Incremental growth** - If we need 10 percent more computing power, we just add 10 percent more processors.

- **Reliability and availability** - A few parts of the system can be down without disturbing people using the other parts.

- **Openness**: An open distributed system is a system that offers services according to standard rules that describe the syntax and semantics of those services.
  - It should be easy to configure the system out of different components.



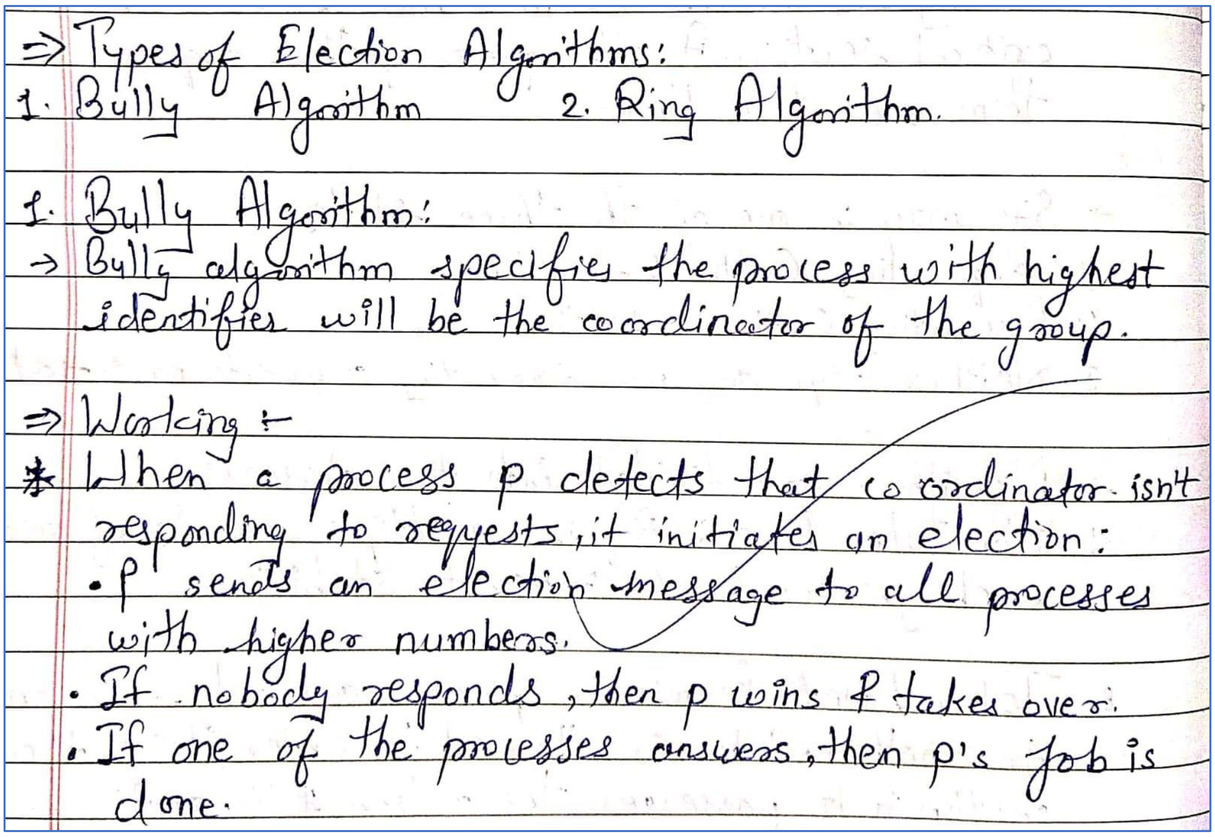
- **Making Resources Accessible** - Main goal of a distributed system
  - make it easy for the users (and applications) to access remote resources
  - to share them in a controlled and efficient way.
    - Resources - anything: printers, computers, storage facilities, data, files, Web pages, and networks, etc.

(b)What is the need for code migration? Explain the code migration issues in detail?

Ans: Code migration in the broadest sense deals with moving programs between machines, with the intention to have those programs be executed at the target

- Code migration is often used for load distribution, reducing network bandwidth, dynamic customization, and mobile agents.
- Code migration increases scalability, improves performance, and provides flexibility.
- Reasons for Code Migration:
- Performance
- Flexibility
- Overall system performance can be improved if processes are moved from heavily-loaded to lightly loaded machines.

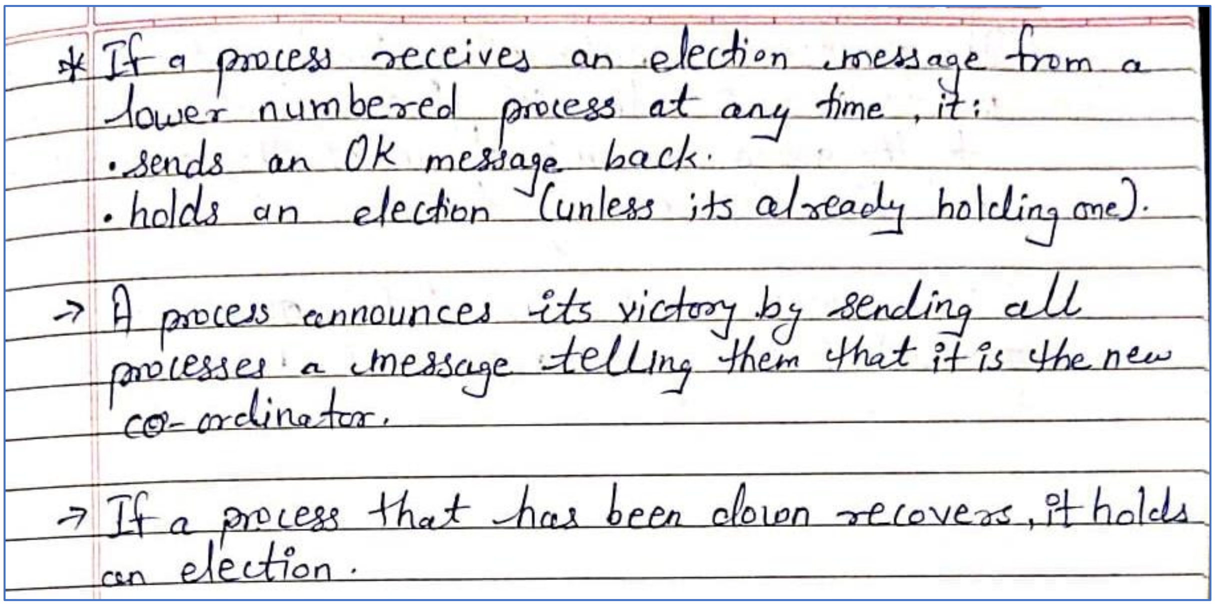
(c)Discuss and compare various election algorithms.



⇒ Types of Election Algorithms:
1. Bully Algorithm 2. Ring Algorithm.

1. Bully Algorithm:
→ Bully algorithm specifies the process with highest identifier will be the coordinator of the group.

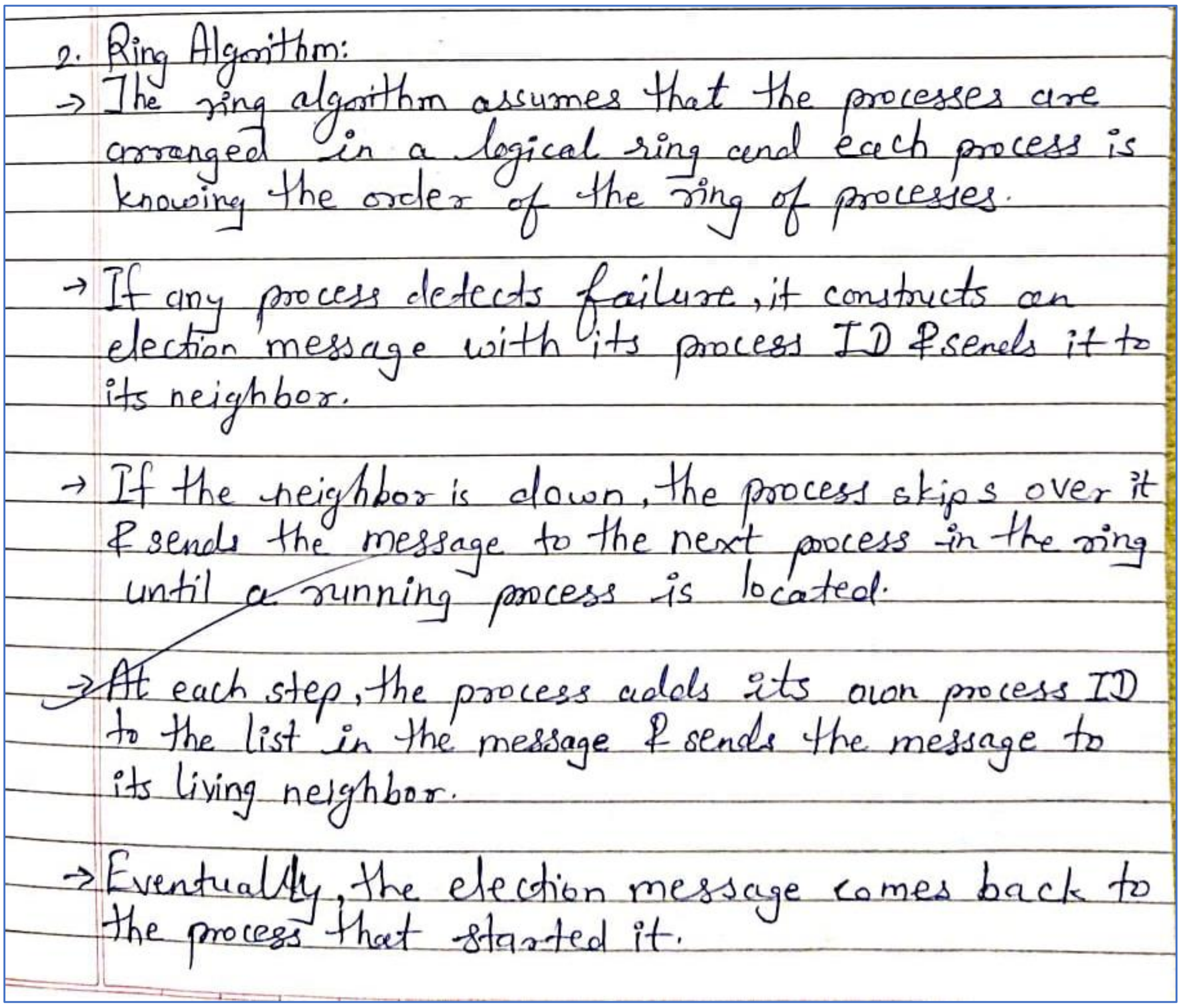
⇒ Working :-
* When a process p detects that coordinator isn't responding to requests, it initiates an election:
• P sends an election message to all processes with higher numbers.
• If nobody responds, then p wins & takes over.
• If one of the processes answers, then p's job is done.



* If a process receives an election message from a lower numbered process at any time, it:
• sends an OK message back.
• holds an election (unless its already holding one).

→ A process announces its victory by sending all processes a message telling them that it is the new co-ordinator.

→ If a process that has been down recovers, it holds an election.



2. Ring Algorithm:

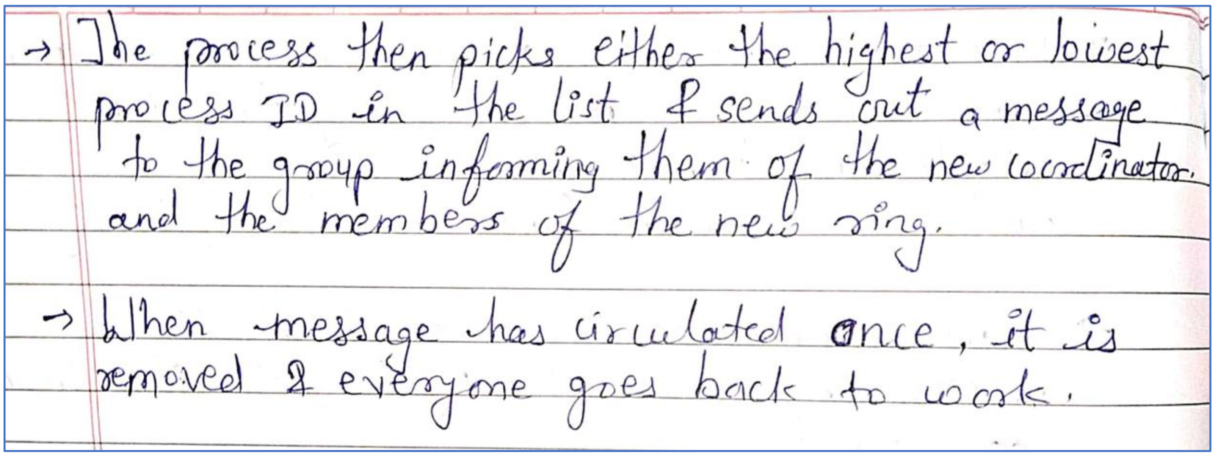
→ The ring algorithm assumes that the processes are arranged in a logical ring and each process is knowing the order of the ring of processes.

→ If any process detects failure, it constructs an election message with its process ID & sends it to its neighbor.

→ If the neighbor is down, the process skips over it & sends the message to the next process in the ring until a running process is located.

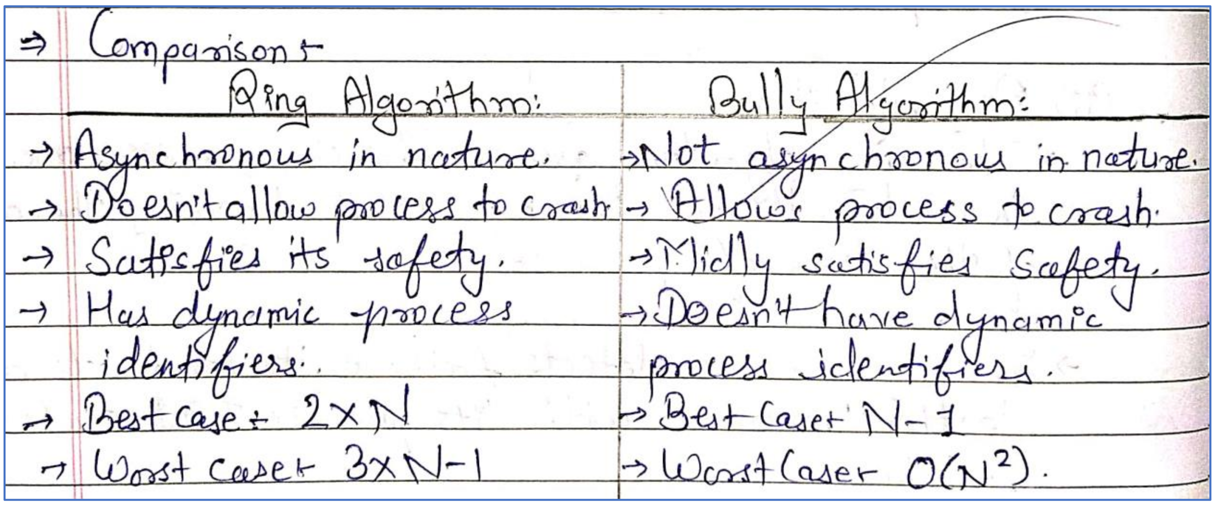
→ At each step, the process adds its own process ID to the list in the message & sends the message to its living neighbor.

→ Eventually, the election message comes back to the process that started it.



→ The process then picks either the highest or lowest process ID in the list & sends out a message to the group informing them of the new coordinator and the members of the new ring.

→ When message has circulated once, it is removed & everyone goes back to work.



⇒ Comparison:-

Ring Algorithm:	Bully Algorithm:
→ Asynchronous in nature.	→ Not asynchronous in nature.
→ Doesn't allow process to crash	→ Allows process to crash.
→ Satisfies its safety.	→ Midly satisfies Safety.
→ Has dynamic process identifiers.	→ Doesn't have dynamic process identifiers.
→ Best case:- 2×N	→ Best Case:- N−1
→ Worst case:- 3×N−1	→ Worst Case:- O(N²).

OR

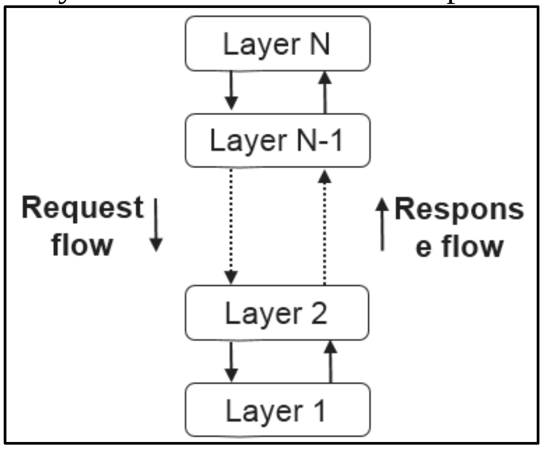
(c)List out the types of System Architectures in DS and explain it.

Ans: Important styles of architecture for distributed systems:

- Layered architectures
- Object-based architectures
- Data-centered architectures
- Event-based architectures

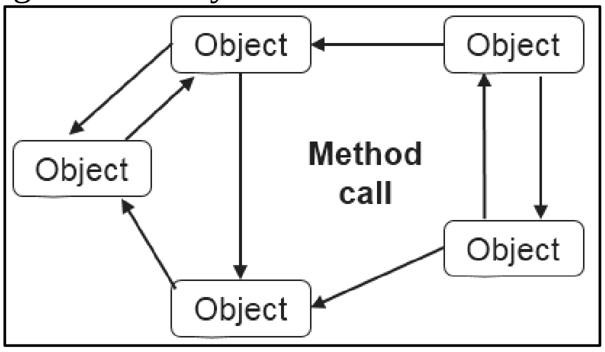
❖ **Layered architectures:**

- The Components are organized in a layered fashion where a component at layer Li is allowed to call components at the underlying layer Li-1, but not the other way around,
- This model has been widely adopted by the networking community
- An key observation is that control generally flows from layer to layer; requests go down the hierarchy whereas the results flow upward.



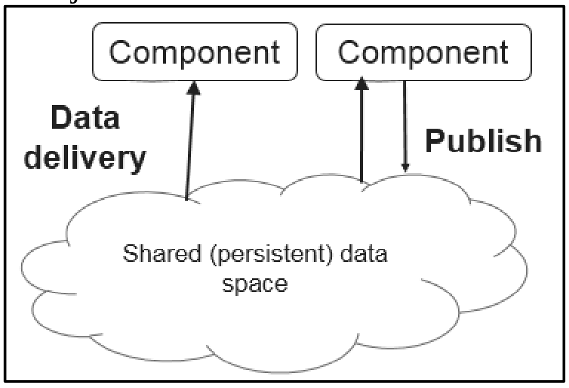
❖ **Object-based architectures**

- Each object corresponds a component
- Components are connected through a (remote) procedure call mechanism.
- The layered and object-based architectures still form the most important styles for large software systems.



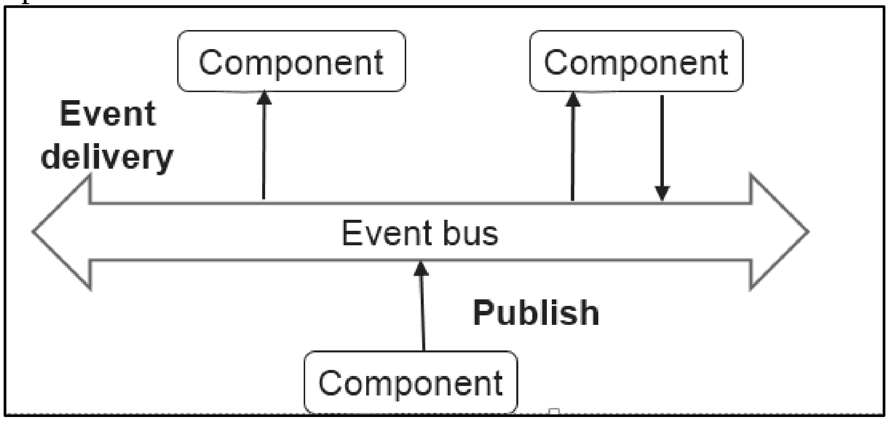
❖ **Data-centered architectures:**

- It evolves around the idea that processes communicate through a common (passive or active) repository.
- It can be argued that for distributed systems these architectures are as important as the layered and object-based architectures



❖ **Event-based architectures:**

- Processes communicate through the propagation of events.
- For distributed systems, event propagation has generally been associated with what are known as publish/subscribe systems.
- Processes are loosely coupled. In principle, they need not explicitly refer to each other. This is also referred to as being decoupled in space, or referentially decoupled.



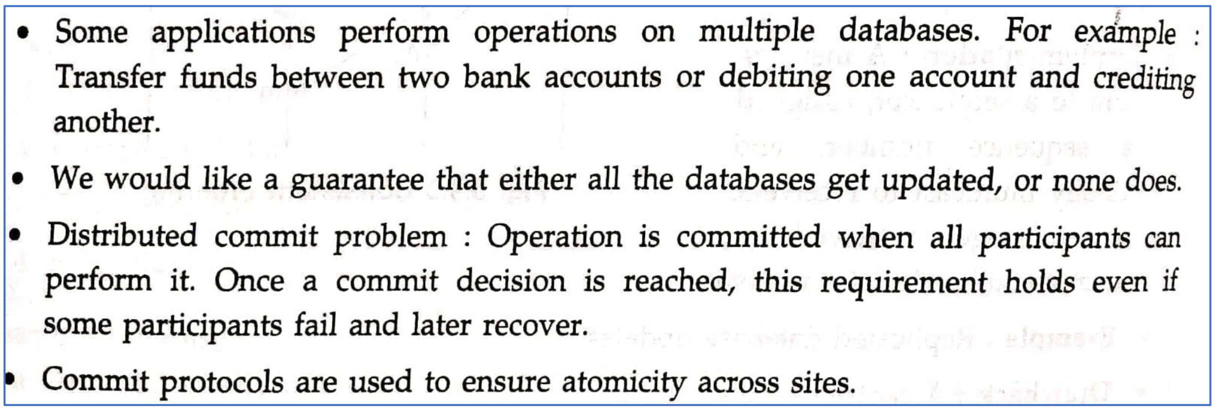
Q.3

(a)Compare: Network Operating System & Distributed Operating System.

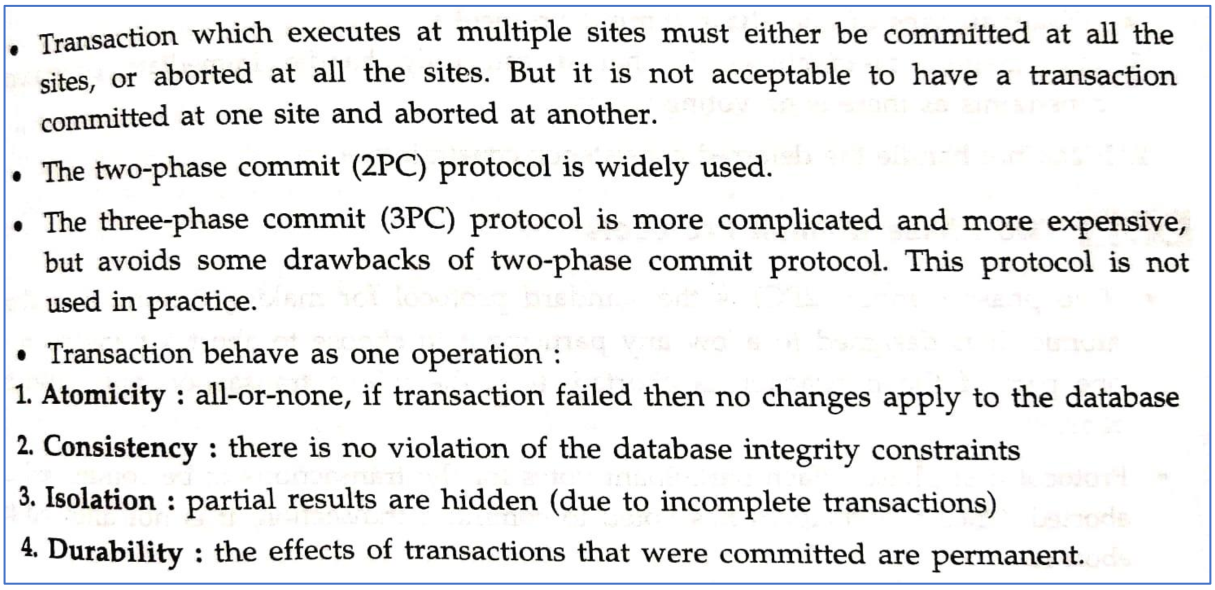
Network Operating System	***Distributed Operating System***
Network Operating System's **main objective is to provide the local services to remote client.**	Distributed Operating System's **main objective is to manage the hardware resources.**
In Network Operating System, **Communication takes place on the basis of files.**	In Distributed Operating System, **Communication takes place on the basis of messages & shared memory.**
Network Operating System is **more scalable than Distributed OS.**	Distributed Operating System is **less scalable than Network OS.**
In Network Operating System, **fault tolerance is less.**	While in Distributed Operating System, **fault tolerance is high.**
In Network Operating System the **rate of autonomy is high.**	In Distributed Operating System the **rate of autonomy is less.**
In Network Operating System **Ease of implementation is also high.**	While in Distributed Operating System **Ease of implementation is less.**
In Network Operating System, **All nodes can have different operating system.**	In Distributed Operating System, **All nodes have same operating system.**

(b) What is distributed commit and recovery in distributed systems?

Distributes Commit:



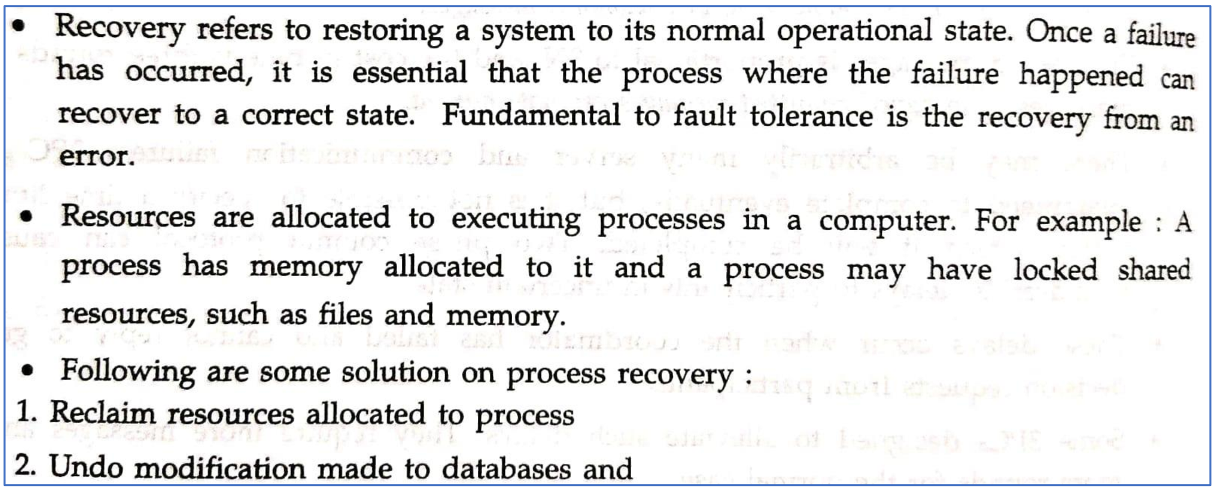
- Some applications perform operations on multiple databases. For example : Transfer funds between two bank accounts or debiting one account and crediting another.
- We would like a guarantee that either all the databases get updated, or none does.
- Distributed commit problem : Operation is committed when all participants can perform it. Once a commit decision is reached, this requirement holds even if some participants fail and later recover.
- Commit protocols are used to ensure atomicity across sites.



- Transaction which executes at multiple sites must either be committed at all the sites, or aborted at all the sites. But it is not acceptable to have a transaction committed at one site and aborted at another.
- The two-phase commit (2PC) protocol is widely used.
- The three-phase commit (3PC) protocol is more complicated and more expensive, but avoids some drawbacks of two-phase commit protocol. This protocol is not used in practice.
- Transaction behave as one operation :

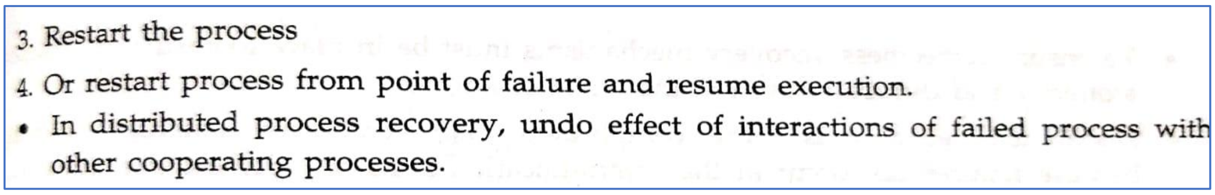
1. **Atomicity :** all-or-none, if transaction failed then no changes apply to the database
2. **Consistency :** there is no violation of the database integrity constraints
3. **Isolation :** partial results are hidden (due to incomplete transactions)
4. **Durability :** the effects of transactions that were committed are permanent.

Recovery:



- Recovery refers to restoring a system to its normal operational state. Once a failure has occurred, it is essential that the process where the failure happened can recover to a correct state. Fundamental to fault tolerance is the recovery from an error.
- Resources are allocated to executing processes in a computer. For example : A process has memory allocated to it and a process may have locked shared resources, such as files and memory.
- Following are some solution on process recovery :

1. Reclaim resources allocated to process
2. Undo modification made to databases and



3. Restart the process
4. Or restart process from point of failure and resume execution.

- In distributed process recovery, undo effect of interactions of failed process with other cooperating processes.

(c)Compare and contrast any 3 consistency models. Pg 103 in ds chap 2-10

Ans: Consistency model defines set of rules that must obey the memory consistency.

➢ A consistency model refers to the degree of consistency that has to be maintained for the shared memory data.

1. **Strict Consistency Model**
2. **Sequential Consistency Model**
3. **Linearizability**
4. **Causal Consistency Model**
5. **Pipelined Random Access Memory Consistency Model**
6. **Weak Consistency Model**
7. **Entry Consistency Model**

i.Strict Consistency model

- Any read on a data item X returns a value corresponding to the result of the most recent write on X
- This is the strongest form of memory coherence which has the most stringent consistency requirement.
- Strict consistency is the ideal model but it is impossible to implement in a distributed system. It is based on absolute global time or a global agreement on commitment of changes.

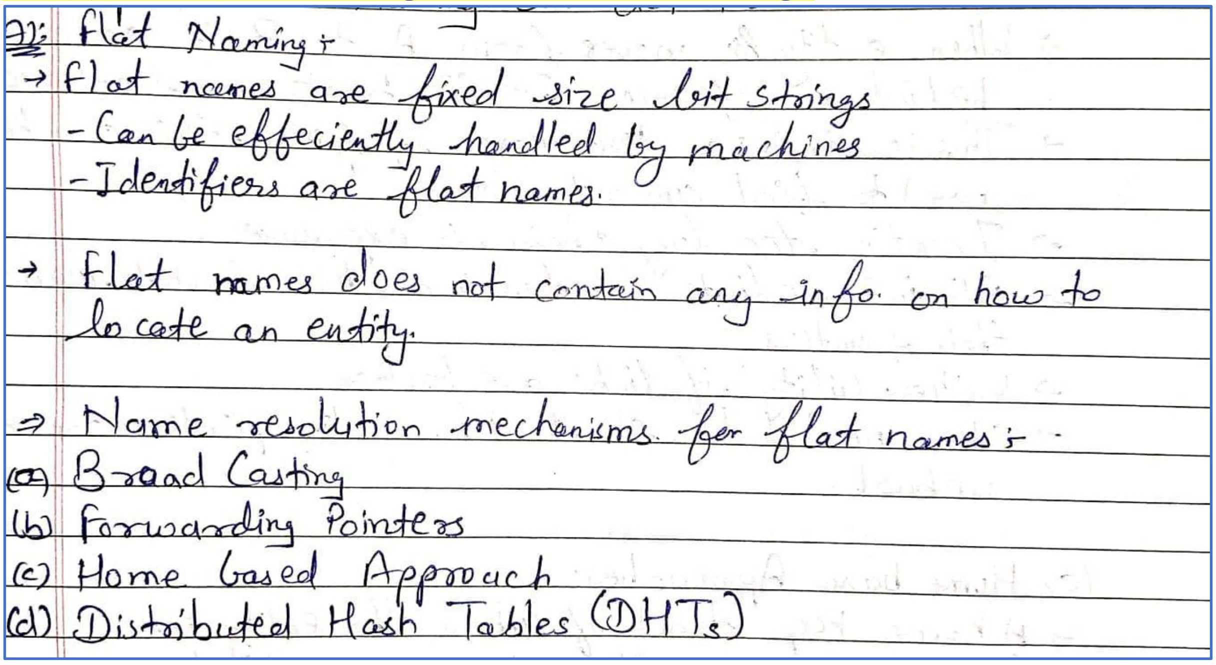
ii.Sequential Consistency

- Sequential consistency is an important data-centric consistency model which is a slightly weaker consistency model than strict consistency.
- A data store is said to be sequentially consistent if the result of any execution is the same as if the (read and write) operations by all processes on the data store were executed in some sequential order and the operations of each individual process should appear in this sequence in a specified order.
- Example: Assume three operations read(R1), write(W1), read(R2) performed in an order on a memory address. Then (R1,W1,R2), (R1,R2,W1), (W1,R1,R2) (R2,W1,R1) are acceptable provided all processes see the same ordering.

iii.Linearizability

- It that is weaker than strict consistency, but stronger than sequential consistency.
- A data store is said to be linearizable when each operation is timestamped and the result of any execution is the same as if the (read and write) operations by all processes on the data store were executed in some sequential order
- The operations of each individual process appear in sequence order specified by its program.
- If tsOP1(x)< tsOP2(y), then operation OP1(x) should precede OP2(y) in this sequence.

Q.3(a)What is flat naming and structured naming?

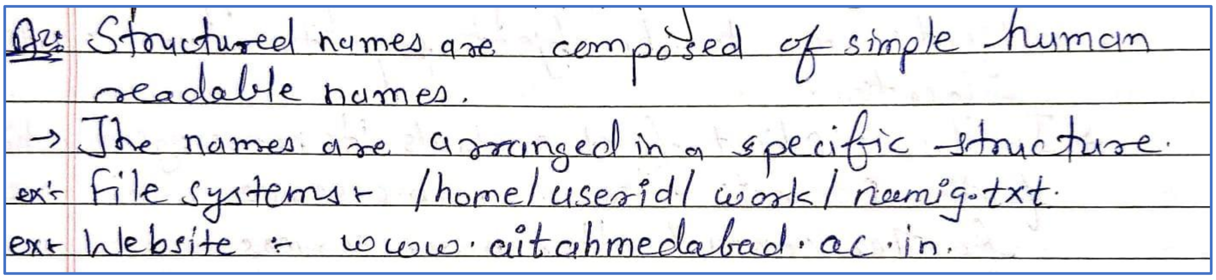


A: Flat Naming:-
→ Flat names are fixed size bit strings
 - Can be effeciently handled by machines
 - Identifiers are flat names.

→ Flat names does not contain any info. on how to locate an entity.

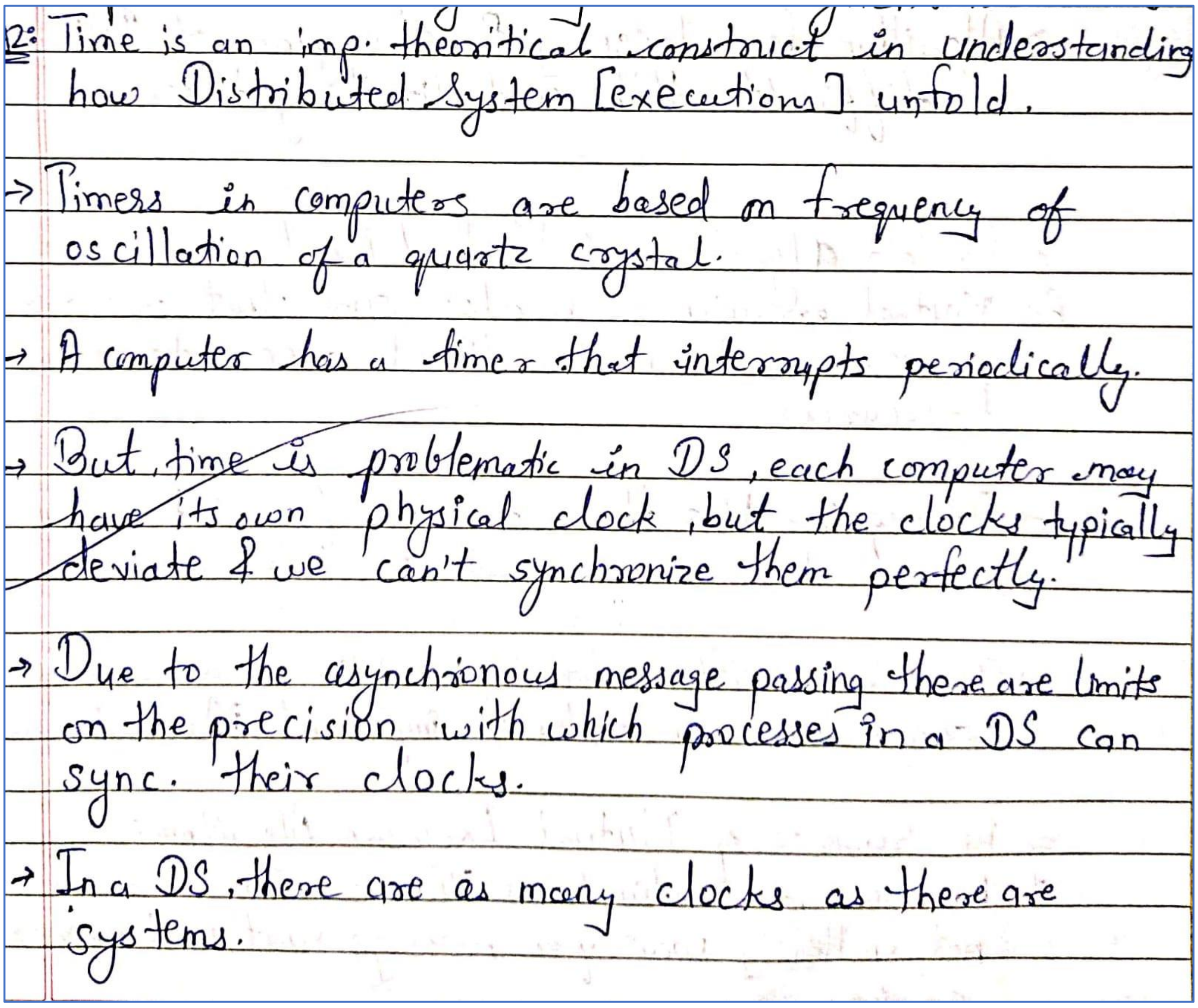
⇒ Name resolution mechanisms for flat names:-
(a) Broad Casting
(b) Forwarding Pointers
(c) Home based Approach
(d) Distributed Hash Tables (DHTs)

Structured Naming:



Ans: Structured names are composed of simple human readable names.
→ The names are arranged in a specific structure.
ex:- File systems:- /home/userid/work/namig.txt.
ex:- Website :- www.aitahmedabad.ac.in.

(b)Enumerate various issues in clock synchronization.



2: Time is an imp. theoritical construct in understanding how Distributed System [executions] unfold.

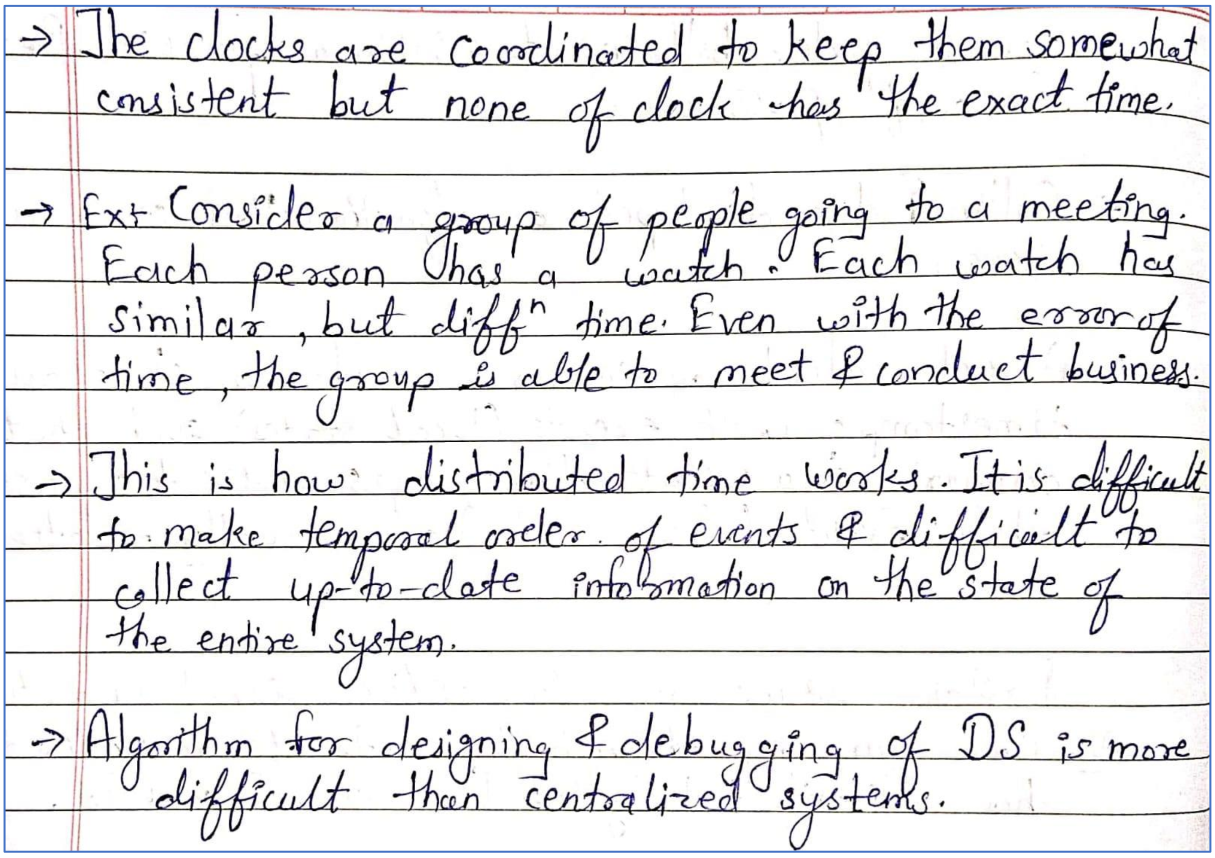
→ Timers in computers are based on frequency of oscillation of a quartz crystal.

→ A computer has a timer that interrupts periodically.

→ ~~But, time is problematic in DS, each computer may have its own physical clock, but the clocks typically deviate & we can't synchronize them perfectly.~~

→ Due to the asynchronous message passing there are limits on the precision with which processes in a DS can sync. their clocks.

→ In a DS, there are as many clocks as there are systems.



→ The clocks are coordinated to keep them somewhat consistent but none of clock has the exact time.

→ Ex: Consider a group of people going to a meeting. Each person has a watch. Each watch has similar, but diffⁿ time. Even with the error of time, the group is able to meet & conduct business.

→ This is how distributed time works. It is difficult to make temporal order of events & difficult to collect up-to-date information on the state of the entire system.

→ Algorithm for designing & debugging of DS is more difficult than centralized systems.

(c)Why mutual exclusion is more complex in distributed systems? Categorize and compare mutual exclusion algorithms.

The design of distributed mutual exclusion algorithms is complex because these algorithms have to deal with unpredictable message delays and incomplete knowledge of the system state on eliminating the mutual exclusion problem in distributed system approach based on message passing is used. Below are the three approaches based on message passing to implement mutual exclusion in distributed systems:

1.Token based approach.
2.Non-Token based approach.
3.Quorum-based approach.

Token Based Algorithm:

- A unique **token** is shared among all the sites.
- If a site possesses the unique token, it is allowed to enter its critical section
- This approach uses sequence number to order requests for the critical section.
- Each request for critical section contains a sequence number. This sequence number is used to distinguish old and current requests.
- This approach ensures Mutual exclusion as the token is unique

Non-token-based approach:

- A site communicates with other sites in order to determine which sites should execute critical section next. This requires exchange of two or more successive round of messages among sites.
- This approach uses timestamps instead of sequence number to order requests for the critical section.
- Whenever a site make request for critical section, it gets a timestamp. Timestamp is also used to resolve any conflict between critical section requests.
- All algorithm which follows non-token-based approach maintains a logical clock. Logical clocks get updated according to Lamport's scheme

Quorum based approach:

- Instead of requesting permission to execute the critical section from all other sites, Each, site requests only a subset of sites which is called a **quorum**.
- Any two subsets of sites or Quorum contains a common site.
- This common site is responsible to ensure mutual exclusion

Q.4

(a)Define IPC. What are the characteristics of IPC?

Ans: Inter process communication (IPC) is a mechanism which allows processes to communicate each other and synchronize their actions.
The communication between these processes can be seen as a method of cooperation between them.
Processes can communicate with each other using these two ways: (a) Shared Memory or Original sharing (b) Message passing or Copy sharing.

The characteristics of inter-process communication in a distributed system:

1. **Synchronous System Calls:**
   In the synchronous system calls both sender and receiver use blocking system calls to transmit the data which means the sender will wait until the acknowledgment is received from the receiver and receiver waits until the message arrives.
2. **Asynchronous System Calls:**
   In the asynchronous system calls, both sender & receiver use non-blocking system calls to transmit the data which means the sender doesn't wait from the receiver acknowledgment.
3. **Message Destination:** ***A local port is a message destination within a computer, specified as an integer. A port has exactly one receiver but many senders.***

Processes may use multiple ports from which to receive messages. Any process that knows the number of a port can send the message to it.

4. **Reliability & Integrity:** The message is sent securely without any loss in packets and any duplication or corruption.
5. **Validity:** If the messages are guaranteed to be delivered without being lost is called validity.
6. **Ordering:** Some applications require messages to be delivered in the sender order i.e the order in which they were transmitted by the sender.

(b)List the differences between RMI and RPC.

NO.	**RPC**	**RMI**
1.	RPC – Remote Procedure Call	RMI – Remote Method Invocation
2.	Is a library & OS dependent platform.	Is a java platform.
3.	RPC supports procedural programming.	RMI supports object-oriented programming.
4.	RPC is less efficient than RMI.	RMI is more efficient than RPC.
5.	RPC creates more overhead.	While it creates less overhead than RPC.
6.	The parameters which are passed in RPC are ordinary or normal data.	While in RMI, objects are passed as parameter.
7.	RPC is the older version of RMI.	RMI is the successor version of RPC.
8.	There is high Provision of ease of programming in RPC.	While there is low Provision of ease of programming in RMI.
9.	RPC does not provide any security.	While it provides client level security.
10.	It's development cost is huge.	It's development cost is fair or reasonable.
11.	There is a huge problem of versioning in RPC.	While there is possible versioning using RDMI.
12.	There is multiple codes are needed for simple application in RPC.	While there is multiple codes are not needed for simple application in RMI.

(c)What is a logical clock? Explain how logical clocks are implemented in distributed system.

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

Suppose, we have more than 10 PCs in a distributed system and every PC is doing it's own work but then how we make them work together. There comes a solution to this i.e. LOGICAL CLOCK.

Method-1:

To order events across process, try to sync clocks in one approach.

This means that if one PC has a time 2:00 pm then every PC should have the same time which is quite not possible. Not every clock can sync at one time. Then we can't follow this method.

Method-2:

Another approach is to assign Timestamps to events.

Taking the example into consideration, this means if we assign the first place as 1, second place as 2, third place as 3 and so on. Then we always know that the first place will always come first and then so on. Similarly, If we give each PC their individual number than it will be organized in a way that 1st PC will complete its process first and then second and so on.

OR

Q.4

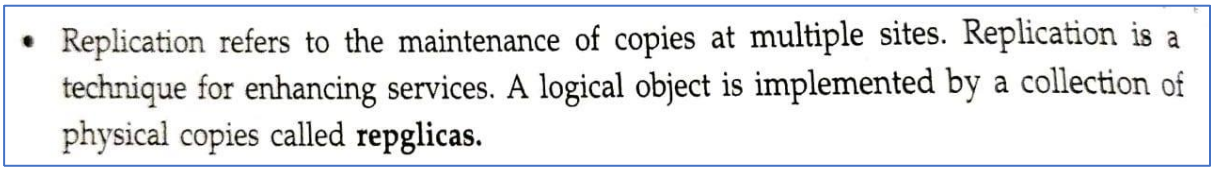
(a)What is cryptography? What is the use of cryptography?

Ans: Cryptography is defined as a means of protecting private information against unauthorized access in cases where physical security is difficult to achieve.

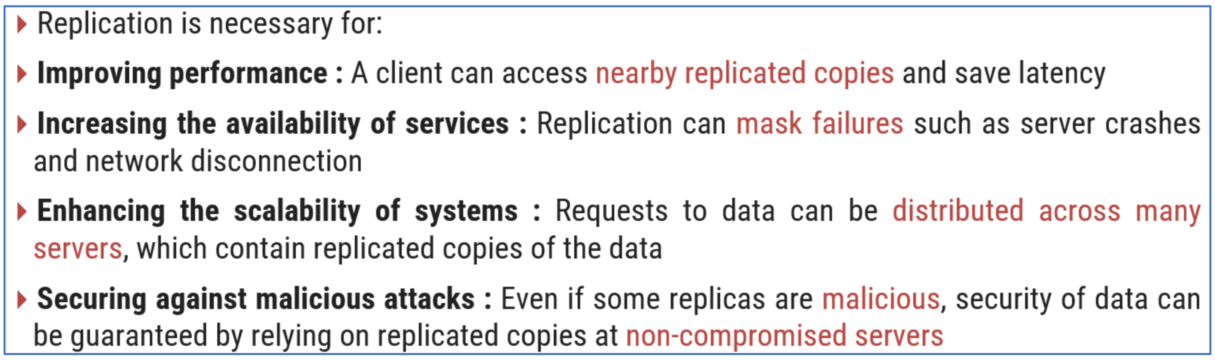
- Cryptography is carried out using two basic operations:
- **Encryption:** The process of transforming intelligible information (plaintext) into unintelligible form (cipher text).
- **Decryption:** The process of transforming the information from cipher text to plaintext.
- Cryptography is used for following:

- Computer passwords
- Digital Currencies
- Secure web browsing
- Electronic Signatures
- Authentication
- Cryptocurrencies
- End-to-end encryption

(b)What is Replication? Write about motivations for replication.



- Replication refers to the maintenance of copies at multiple sites. Replication is a technique for enhancing services. A logical object is implemented by a collection of physical copies called **repglicas.**



- Replication is necessary for:
- **Improving performance :** A client can access nearby replicated copies and save latency
- **Increasing the availability of services :** Replication can mask failures such as server crashes and network disconnection
- **Enhancing the scalability of systems :** Requests to data can be distributed across many servers, which contain replicated copies of the data
- **Securing against malicious attacks :** Even if some replicas are malicious, security of data can be guaranteed by relying on replicated copies at non-compromised servers

(c)What is RPC? Discuss the design issues for RPC.

Ans: **Remote Procedure Call (RPC)** is a powerful technique for constructing **distributed, client-server based applications**. It is based on extending the conventional local procedure calling so that the **called procedure need not exist in the same address space as the calling procedure**. The two processes may be on the same system, or they may be on different systems with a network connecting them.

1. RPC Runtime: RPC run-time system, is a library of routines and a set of services that handle the network communications that underlie the RPC mechanism. In the course of an RPC call, client-side and server-side run-time systems' code handle binding, establish communications over an appropriate protocol, pass call data between the client and server, and handle communications errors.

2. Stub: The function of the stub is to provide transparency to the programmer-written application code. On the client side, the stub handles the interface between the client's local procedure call and the run-time system, marshalling and un-marshalling data, invoking the RPC run-time protocol, and if requested, carrying out some of the binding steps. On the server side, the stub provides a similar interface between the run-time system and the local manager procedures that are executed by the server.

3. Binding: The most flexible solution is to use dynamic binding and find the server at run time when the RPC is first made.

Binding consists of two parts: Naming & Locating:

A Server having a service to offer exports an interface for it. Exporting an interface registers it with the system so that clients can use it.

A Client must import an (exported) interface before communication can begin.

4. The call semantics associated with RPC:

It is mainly classified into following choices-

- **Retry request message –**
- **Duplicate filtering –**
- **Retransmission of results –**

Q.5

(a)What is multicasting? List the characteristics of multicasting.

Ans: Multicast communication allows a process to send the same message to a group of processes.

- As multicast operations can provide the programmer with delivery guarantees that are difficult to realize for the application programmer using ordinary unicast operations.
- Group communication that simplifies building reliable efficient distributed systems.

➢ Multicast messages provides a useful infrastructure for constructing distributed systems with the following characteristics:

1. Replicated services: A replicated service consists of a group of members. Client requests are multicast to all the members of the group, each of which performs an identical operation.

2. Better performance: Performance of service is increase by using data replication. User's computer is used for replication. Each time the data change, the new value is multicast to the processes managing the replicas.

3. Propagation of event notifications: Multicast to a group may be used to notify processes when something happens. For example, a news system might notify interested users when a new message has been posted on a particular newsgroup. Group view is the lists of the current group members. When a membership change occurs, the application is notified of the new membership.

(b)What is CORBA's common Data Representation? Explain.

Ans: Common Data Representation (CDR) is **used to represent structured or primitive data types passed as arguments or results during remote invocations on Common Object Request Broker Architecture (CORBA) distributed objects**. It enables clients and servers written in different programming languages to work together.

- For example, it translates little-endian to big-endian. It assumes prior agreement on type, so no information is given with data representation in messages.

(c)Explain the common approaches to user authentication. What problems are associated with these approaches?

Ans: Authentication is the process of identifying users that request access to a system, network, or device.

- User authentication is a method that keeps unauthorized users from accessing sensitive information.
- For example, User A only has access to relevant information and cannot see the sensitive information of User B.

Common Authentication Methods are:

1. Password-based authentication:

Passwords are the most common methods of authentication. Passwords can be in the form of a string of letters, numbers, or special characters. To protect yourself you need to create strong passwords that include a combination of all possible options.

❖ Hackers can easily guess user credentials by running through all possible combinations until they find a match.

2.Multi-factor authentication:

Multi-Factor Authentication (MFA) is an authentication method that requires two or more independent ways to identify a user. Examples include codes generated from the user's smartphone, Captcha tests, fingerprints, voice biometrics or facial recognition.

❖ People may lose their phones or SIM cards and not be able to generate an authentication code.

3.Certificate-based authentication:

Certificate-based authentication technologies identify users, machines or devices by using digital certificates. A digital certificate is an electronic document based on the idea of a driver's license or a passport.

❖ People may lose their pen drive or can say disk that has the certificate stored in it.

4.Biometrics authentication:

Biometrics authentication is a security process that relies on the unique biological characteristics of an individual. Common biometric authentication methods include:

Facial recognition, **Fingerprint scanners,** voice recognition, **Eye scanners**

❖ Biometric systems can make two basic errors. A "false positive" occurs when the system incorrectly matches an input to a non-matching template and vice-versa.

5.Token-based authentication

Token-based authentication technologies enable users to enter their credentials once and receive a unique encrypted string of random characters in exchange.

❖ Token based authentication can be compromised by Man-in-the-middle, Token steal, breaches of the secret key

OR

Q.5(a)Difference between Authorization and Authentication.

Authentication	***Authorization***

Usually, the first step of a security access control	Usually comes after authentication
Verifies the user's identity	Grants or denies permissions to user
Common methods include: username, password, security question answer, code sent via SMS/email	Permissions are granted and monitored by the organization
Uses biometric data like fingerprint, face recognition, retinal scan	Common methods include: access control as per attribute & role
It's visible and changeable by the user	It's not visible or changeable by the user
Ex: Employees in a company are required to authenticate through the network before accessing their company email	Ex: After an employee successfully authenticates, the system determines what information the employees are allowed to access

(b) What is DFS? Also write the features of DFS.

Ans: (DFS):

- A distributed file system (DFS) is a file system with data stored on a server.
- The data is accessed and processed as if it was stored on the local client machine.
- The DFS makes it convenient to share information and files among users on a network in a controlled and authorized way.
- Server allows the client users to share files and store data just like they are storing the information locally.
- Servers have full control over the data and give access control to the clients.

Features of DFS:

1.Transparency in the DFS means hiding, its types are given below:

- **Structure transparency** –
  There is no need for the client to know about the number or locations of file servers and the storage devices.
- **Access transparency –**
  Both local and remote files should be accessible in the same manner.
- **Naming transparency –**
  Once a name is given to the file, it should not be changed during transferring from one node to another.
- **Replication transparency –**
  If a file is copied on multiple nodes, both the copies of the file and their locations should be hidden from one node to another.

2.User mobility:

It will automatically bring the user's home directory to the node where the user logs in.

3.Performance:

Performance is based on the average amount of time needed to convince the client requests. It should be similar to that of a centralized file system.

4.Simplicity and ease of use:

The user interface of a file system should be simple and the number of commands in the file should be small.

5.High availability:
A distributed file system should continue to function even when partial failures occur due to the failure of one or more components.
6.Scalability:
A good distributed file system should be designed to easily cope with the growth of nodes and users in the system.
7.High reliability:
The probability of loss of stored data should be minimized.
8.Data integrity:
Concurrent access requests from multiple users must be properly synchronized by the use of some concurrency control mechanism.
9.Security:
A distributed file system should be secure so that its users may trust that their data will be kept private.
10. Heterogeneity:
There should be easy access to shared data on diverse platforms (e.g. Unix workstation, Wintel platform etc.).

(c)Explain the DNS name service and bind implementation of DNS.
Ans: Domain Name Service (DNS):
- Domain Name Service (DNS) is widely used to access the Internet.
- It is an Internet directory service that provides a way to map the user-friendly name of a computer or service to its numeric address.
- The DNS defines and describes how domain names are translated into IP addresses, and how it controls email delivery.
- A client computer queries a DNS server, asking for the IP address of a computer configured to use host a.example.microsoft.com as its DNS domain name.
- The DNS server answers the query based on its local database and replies with an answer containing the requested information, which contains the IP address information for host a.example.microsoft.com.

➢ BIND Implementation:
- BIND (Berkeley Internet Name Domain) is a software collection of tools including the world’s most widely used DNS (Domain Name System) server software.
- This feature-full implementation of DNS service and tools aims to be 100% standards-compliant and is; intended to serve as a reference architecture for DNS software.
- BIND is the most commonly used DNS server software on the Internet.
- BIND provides features like load balancing, notify, dynamic update, split DNS, DNSSEC, IPv6, and more.
- BIND has the following main components: Name Server, Lightweight Resolver, Name Server Tools.