Name - Vidit puri goswami
DS - mid term

5 July

Sec A

Q1 - Ans - Peer - to - Peer
Q2 - Ans - Server
Q3 - Ans - Communication ~~within the process~~ between two processes
Q4 - Ans - Program counter
Q5 - Ans - Scalability

Sec - B

Q2 Three features of DS.

1) Concurrency
Multi - programming
Multi - processing

Parallel exertions in distributed systems
1) Many users using the same resources. application interaction

2) Many servers responding to client requests.

2) Scalability
how the system handles growth.

Small system - two computers and a file server on a single network
Large system - Current Internet

Scalability
- Software should not change to support growth.
- research areas - for large, high performance networks.

Avoid centralization to support scalability.

3) Transparency.
transparency of
- access
- location
- Concurrency
- replication
- failure
- migration
- performance
- Scaling

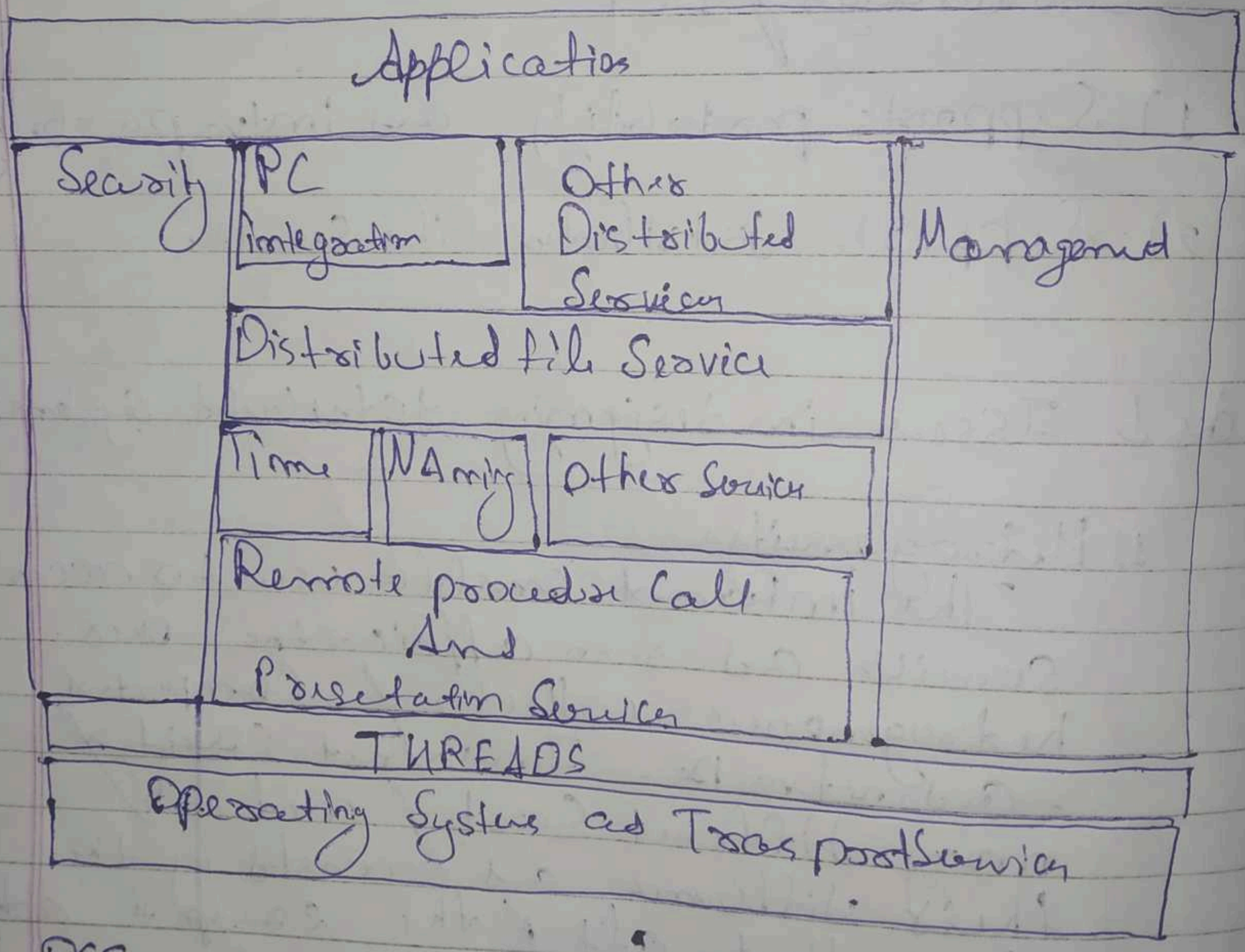
Ans 3 - DCE Distributed Computing environment is a client/server architecture, defined by the open software foundation (OSF) that provides open system platform to address challenges of distributed computing

DCE is middleware software that sits between the application and operating System layer.

DCE architecture

- DCE cells:

The DCE system is highly scalable To accommodate such large system DCE uses the concept of Cells. A cell is a basic unit of operation in the DCE which breaks down a large system into smaller manageable units



- DCE components:

D.C.E use various technology that form the Components such as:

1) Thread package
2) Remote procedure call
3) Name service
4) Distributed file service
5) Time service
6) Scheduling and Synchronization
7) Security service

Advantages of DCF

1) Supports portability and interoperability.
2) Supports distributed file service

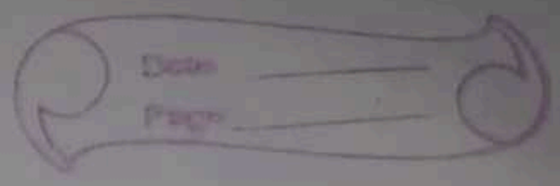
Ans 5 Issues in designing distributed systems:-

1) Heterogeneity:-
The Internet enables users to access services and run applications over a heterogeneous collection of computers and networks. The Internet consists of many different sorts of network; their differences are masked by the fact that all of the computers attached to them use the Internet protocols to communicate with each other.

2) openness :-
The openness of a compute System is a characteristic that determines whether the System can be extended as re-implanted in various way.

3) Security :-
Many of the information resources that are made available and maintained in distributed Systems have a high intrinsic value to their user. Therefore security of them is considerable important.

4) Scalability.
Distributed Systems operate effectively and efficiently at many different scales, ranging from a small intranet to internet.
A system is described as scalable if it will remain effective when there is a significant increase the no of resources and the no of users.



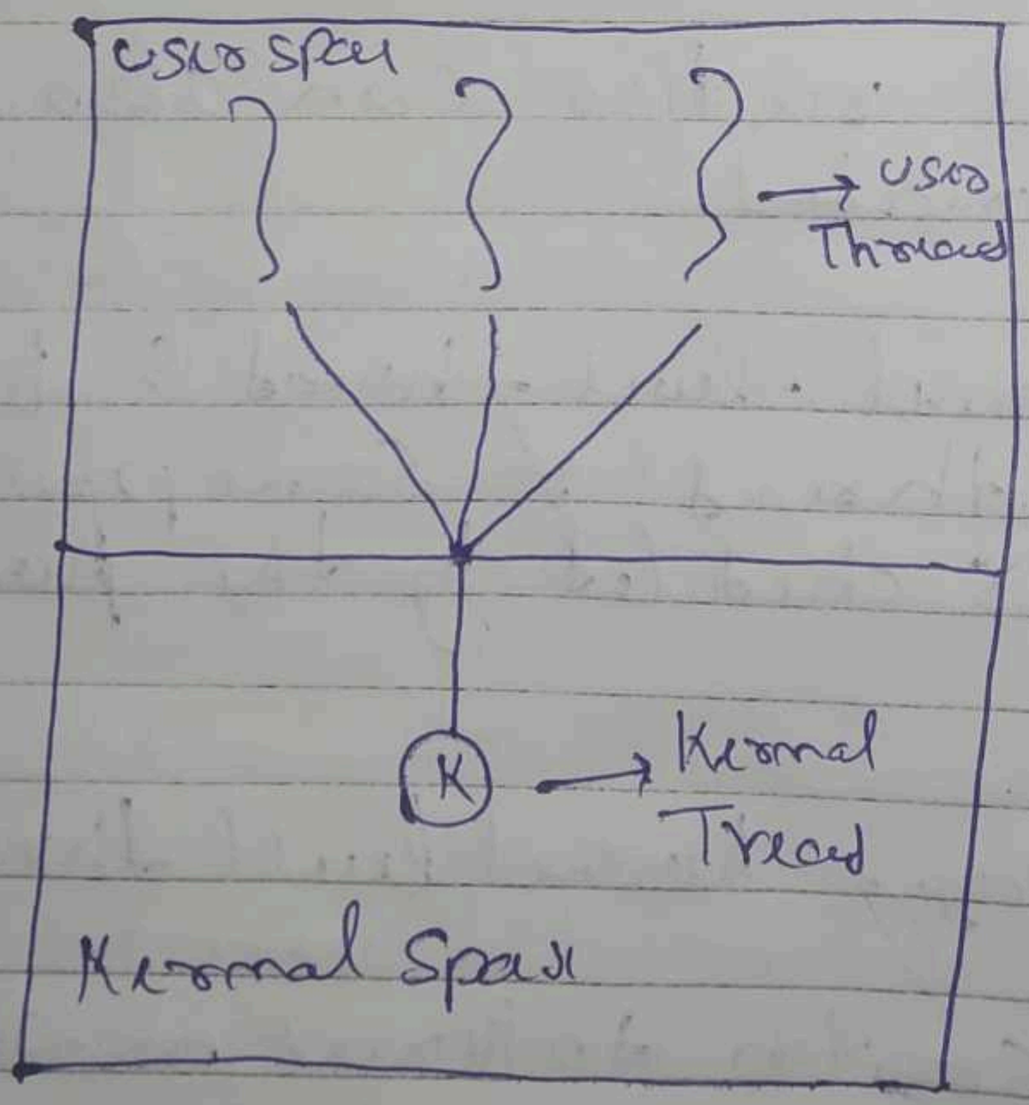
5) failure handling :-
Computer systems sometimes fail.
when failures occur in hardware
or software, programs may produce
incorrect results or may stop
before they have completed the
intended computation.
Failure in a distributed system is
partical.

6) Transparency :-
Transparency can be achieved at
two different levels. Easiest to do is
to hide the distribution from users.
The concept of transparency
can be applied to several aspects of
a distributed system.

a) location
b) Migration
c) Replication
d) concurrency
e) parallelism

7 Performance
Always the hidden data is the
background is the issue of Performance.
Building a transparent, flexible,
scalable distributed System, must
important like in its performance.
In particular, when running a
particular application on distributed
System, it should not be appreciably
worse than running the same
application on a single processor.
Unfortunately, achieving this is
easier said than done.

Ans 6 - Kernal level thread.



Kernal-level threads are handled by the

Operating System directly aid the thread manager is done by the Kernel. The context information for the process as well as the process threads is all managed by the Kernel. Because of this, Kernel level threads are slower than user level threads.

Advantages of Kernel level threads.

- Several multiple threads of the same process can be scheduled on different processors in Kernel level threads.
- The Kernel routine can also be multithreaded.
- If a Kernel-level thread is blocked another thread of same process can be scheduled by the Kernel

Disadvantages of Kernel level threads.

- A mode switch to Kernel mode is required to transfer control from one thread to another in a process.

- Kernel level threads are slower to create as well as manage as compare to user level threads.

Sec - C

Ans 1- Transparency :- The distributed system should be perceived a single entity by the users or the application programmers rather than as a collection of autonomous systems, which are cooperating. The user should be unaware of where the resources are located and also the transfer from a local machine to remote one should also be transparent.

Types of transparencies :-

1) Access transparency :- clients should be unaware of the distribution of the files. The files could be present on a totally different set of servers which are physically distant apart and a single set of operations should be provided to access them remote as well as local files

2) Location transparency:
Clients should see a uniform file namespace. files or groups of file may be relocated without changing their pathnames.
A location name contains no information about the named object's physical location.

3) Concurrency transparency :- Users and Applications should be able to access shared data or objects without interference between each other.

4) Replication :- This kind of transparency should be mainly in cooperation for the distributed file system, which replicate the data set two or more sites for more reliability.
The client should not be aware that a replicated copy of data exist.

5) Failure transparency :- Enable the concealment of faults, allowing user and application programs to complete their tasks despite the failure of hardware or software components.

Fault tolerant is provided by the mechanism ~~made~~ related to access transparency.

6) Migration transparency :- This allows the used to user movement of the movement of information or process within a system without affecting the operation of the user and the application beta can summary.

7) Performance transparency :- Allows the system to be reconfigure to improve the performance as the load varies.

8) Scaling transparency :-
A system should be able to grow with affecting application algorithms. graceful growth as evolution is an important requirement for most enterprises.

A-2 OSI System Interconnection (OSI) model

A-3 Omission failure :- Omission failure are caused access the servers due to lack of reply or response from the servers access the distributed system. There are difference issue raised due to this omission failure and the key among them are server not listening or a typical buffer overflow issues access the servers of the distributed systems.

2) Timing failure :- Timing failure are called access the servers distributed system. The usual behavior of those timing failure would be like that the servers response time towards the client request would be more than the expected range. control flow out of the response may be caused due to these timing failure and the corresponding clients may give up as they

can't wait for the request dispose from the servers as thus the servers operations on failed due to this.

3) Vault failure.
If a server value ↑ unavailable, client that have been work with this servers

3) arbitar failure :- ∞ (Byzantim failure)
Then failure cause the servers to behave arbitary in nature and the server to respose in an arbitary paesim at arbitary time accross the distributed systems. Output from the servers would be inappropriately as there could be chance of malicious users as duplicate message from the server site as the client get arbitary of unwanted duplicate update from the servers due to their failure.