

02/02/22

DS-Tutorial-04

U19CS009

Page No.:

Date: / /



Q-1

D/B remote procedural call and local call

1.	Remote Procedural Calls	Local Calls
----	-------------------------	-------------

- a) RPCs are used to pass message between components of distributed system applications LPCs provide a mechanism for establishing different parts of an application located on a single computer/machine to communicate with each other.
- b) RPC is relatively slower as compared to LPC, as it uses network
- c) In RPC, programs can be executed from different machines. In LPCs programs are executed by host machine.
- d) Remote calls may fail because of unpredictable network problems Local calls are not affected by network problems
- e) Less reliable More reliable

- Q-2 Remote procedure call (RPC) allows a program to execute a on another machine (i.e., other than hosts as well). Programmers write the same code no matter what the procedure is local or remote.

Fo

Following is the sequence of events

- 1.) Client procedure calls client stub in normal case
- 2.) Client stub builds message, calls local OS.
- 3.) Client's OS sends message to remote OS.
- 4.) Remote OS gives message to server stub.
- 5.) Server stub unpacks parameters, call server.
- 6.) Server does work, returns result to stub.
- 7.) Server stub packs it in message, calls local OS.
- 8.) Server's OS sends message to client's OS
- 9.) Client's OS gives message to Client stub
- 10.). Client stub unpacks results, returns to client.

Q-3 Server management is the process of monitoring and maintaining servers to operate at peak performance. It also encompasses the management of hardware, software, security and backups.

The primary goals of an effective server management strategy are as follows:-

- o Minimize and hopefully eliminate server slowdowns as well as downtime
- o Build secure server environments
- o Ensure that server continues to meet the needs of an organization as it evolves.

Q-4 TYPES of Communication are:-

a) Data Oriented v/s Control Oriented

- o Data Oriented -> facilitates the data exchange between threads

Q) 2

a) Control oriented → associates a transfer of control with every data transfer remote procedure call (RPC) and remote invocation (RMI).

- eg → Hardware & OS provide data-oriented comm. I were
→ Higher level infra. provides control-oriented comm. middleware

b) Synchronous → sender blocks until message received.

→ Mostly sender is blocked until message is processed and reply is received.

→ Sender & receiver must be active at the same time.

eg → Client-Server comm. generally used synchronous comm.

b) 2) Asynchronous → sender continues to execute after sending message (does not block waiting for reply).

→ Message may be queued if receiver not active

→ message may be processed later at receiver's convenience.

c) 1 Transient → message discarded if delivered to receiver immediately.

→ A message will be delivered only if a receiver is active.

→ If no active receiver, an undelivered message will be dropped.

→ eg → HTTP request

Q) 2

Persistent → A message will be stored in the system until it can be delivered to the intended recipient.

eg. - email