

QUESTIONS ON INTER-PROCESS COMMUNICATION

1) What is Inter-process communication?

Ans: Exchange of data between two or more separate, independent processes/threads.

- Operating systems provide facilities/resources for inter-process communications (IPC), such as message queues, semaphores, and shared memory.
- Distributed computing systems make use of these facilities/resources to provide application programming interface (API) which allows IPC to be programmed at a higher level of abstraction. (e.g., send and receive)
- Distributed computing requires information to be exchanged among independent processes.

2) What are the models of IPC?

Ans: Two basic models are used in IPC:

Shared memory: "shared data" are directly available to each process in their address space.

Message passing: "shared data" are explicitly exchanged.

3) What do you mean by "unicast" and "multicast" IPC?

Ans: In distributed computing, two or more processes engage in IPC using a protocol agreed upon by the processes. A process may be a sender at some points during a protocol, a receiver at other points.

- When communication is from one process to a single other process, the IPC is said to be a unicast, e.g., Socket communication.
- When communication is from one process to a group of processes, the IPC is said to be a multicast, e.g., Publish/Subscribe Message model.

4) Write operations provided in IPC?

Ans: Receive ([sender], message storage object)

- Connect (sender address, receiver address), for connection-oriented communication.
- Send ([receiver], message)
- Disconnect (connection identifier), for connection-oriented communication.

5) What is event synchronization?

Ans: Inter-process communication may require that the two processes synchronize their operations

- One side sends, and then the other receives until all data has been sent and received.
- Ideally, the send operation starts before the receive operation commences.

In practice, the synchronization requires system support.

6) Differentiate between synchronous and asynchronous communication?

Ans: The IPC operations may provide the synchronization necessary using a blocking operation issued by a process will block further processing of the process until the operation is fulfilled.

Alternatively, IPC operations may be asynchronous or non-blocking. An asynchronous operation issued by a process will not block further processing of the process. Instead, the process is free to proceed with its processing, and may optionally be notified by the system when the operation is fulfilled.

7) Explain deadlocks and timeout.

Ans: Blocking operations issued in the wrong sequence can cause deadlock. Timeout can be used to detect deadlocks. Connect and receive operations can result in indefinite blocking. For example, a blocking connect request can result in the requesting process to be suspended indefinitely if the connection is unfulfilled or cannot be fulfilled, perhaps as a result of a breakdown in the network. It is generally unacceptable for a requesting process to “hang” indefinitely. Indefinite blocking can be avoided by using timeout.

8) State IPC paradigms and implementations.

Ans: Paradigms of IPC of different levels of abstraction have evolved, with corresponding implementations.

IPC paradigms:

- remote procedure/method
- socket API
- Data transmission

IPC implementations:

- Remote Procedure Call(RPC)
- UNIX socket API, Winsock

- Serial/parallel communications.

9) Are function callback and inter-process communication same?

Ans: Function callback is passing the address (or some other identifier) of a function to another function so that it can call back for some reason.

IPC, on the other hand, is a means for processes to communicate with each other, such as shared memory, pipes, semaphores and so on.

Some IPC mechanisms may *use* callbacks but it's by no means necessary. For example, sockets don't use callbacks, they simply allow a user to call *read and write*.

10) What does route command do?

Ans: It shows static routes configured on your PC along with your base routing table. Routes are the way traffic from your computer flows to get to where it needs to be.

