

ASSIGNMENT - 01

Bahasna B.A

2017122

III year CSE

DISTRIBUTED

COMPUTING (DC)

ASSIGNMENT-01

① INTER PROCESS COMMUNICATION:

* A system can have two types of processes

1. Independent process.
2. Cooperating process.

* Cooperating processes affect each other and may share data and information among themselves.

* IPC provides a mechanism to exchange data and information across multiple processes, which might be on single / multiple computers connected by a network.

ADVANTAGES:-

- Computational Speedup
- Modularity
- Information and data sharing
- Privilege separation

Approaches for Inter Process Communication

* Pipes

* Message que

* FIFO

* Shared Memory

* Indirect communication

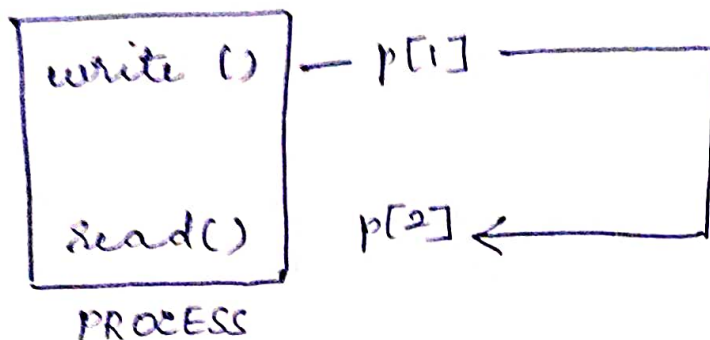
* Direct communication

* Message passing.

* Pipes:-

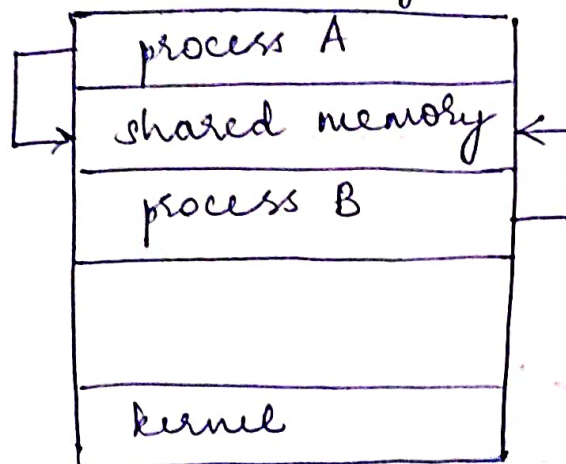
* It is a half duplex method used for IPC between two related processes.

* The filling process is writing into the pipe and the reading process is retrieved from the pipe.



* Shared Memory:-

* Multiple processes can access a common shared memory. Multiple processes communicate by shared memory, where one process makes changes at a time and then others view the change. Shared memory doesn't use kernel.



* Message Passing:

* In IPC, this is used by a process for communication and synchronisation.

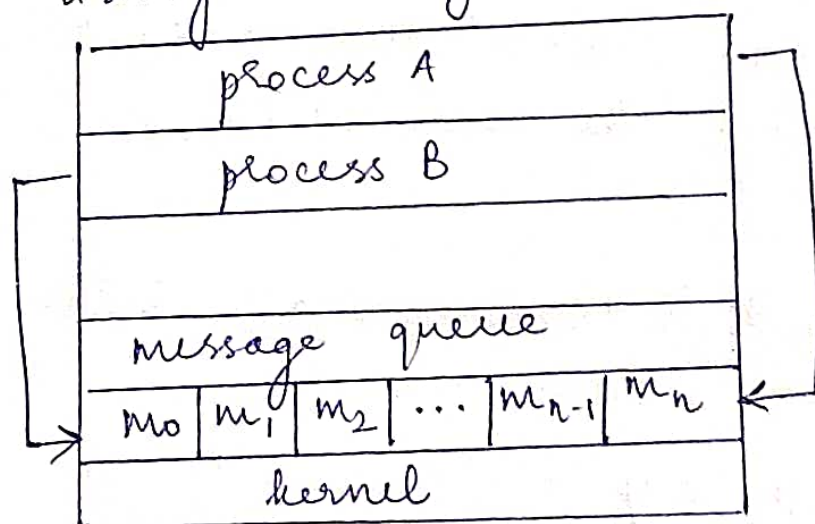
* Processes can communicate without any shared variables, therefore it can be used in a distributed environment on a network.

* It is slower than the shared memory technique.

* It has 2 actions sending (fixed size message) and receiving messages.

* Message Queues:

We have a linked list to store messages in kernel of OS and a message queue is identified using "message queue identifier".

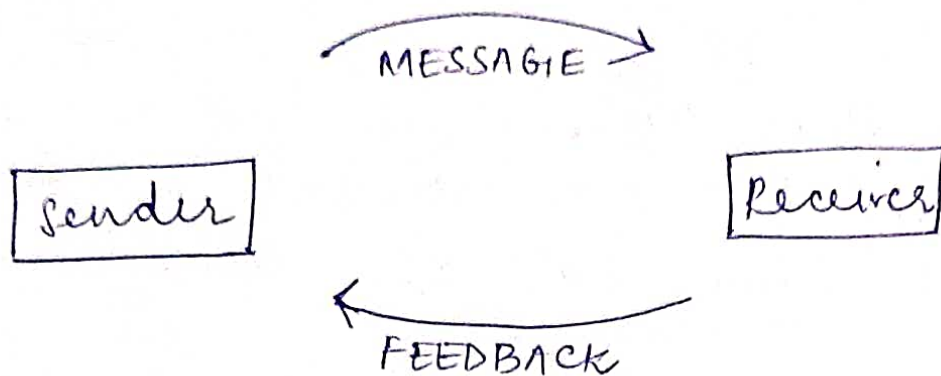


* Direct Communication:

* Process that want to communicate must name the sender or receiver.

* A pair of communicating processes must have one link between them.

* A link (generally bi-directional) establishes between every pair of communicating processes.



Indirect Communication :

* Pairs of communicating processes have shared mailboxes.

* Link is established between pair of processes.

* Sender process puts the message in the port or mailbox of a receiver process and the receiver process takes out (or deletes) the data from the mailbox.

FIFO:

* Used to communicate between two processes that aren't related.

* Full duplex - process P_1 is able to communicate with process P_2 , and vice versa.

② MARSHALLING:

Marshalling is the process of taking a collection of the data structures to transfer and format them into an external data representation type which suits for transition in a message.

External Data Representation:

APPROACHES:-

① Common Object Request Broker Architecture (CORBA):-

* CORBA is a specification developed by a Object Management group (OMG) currently the leading middleware solutions in most distributed systems. It's a specification for creating, distributing, and managing objects in distributed networks.

* CORBA describes a messaging mechanism by which objects distributed over a network can transfer messages with each other irrespective of that platform or language used to create those objects.

* This enables collaboration b/w systems on different architectures, operating systems, programming languages as well as computer hardware.

② Java's Object Serialisation:

* In Java RMI, both objects and primitive data values may be passed by arguments and results of method invocations.

* The term serialisation refers to the activity of flattening an object or a connected set of objects onto a serial form that is suitable for storing on disk or transmitting in a message.

③ XML [Extensible Markup Language]:

* XML is a markup language that was defined by the W3C Consortium for general use on the web.

* XML was initially developed for writing structured documents for the web.

* XML is used to enable clients to communicate with web services and for defining the interfaces and other properties of web services.

⑤ REMOTE METHOD INVOCATION (RMI):-

* The RMI (Remote Method Invocation) is an API that provides a mechanism to create distributed application in Java. The RMI allows an object to invoke methods on an object running to another JVM.

* The RMI provides remote communication between the applications using 2 objects

- Stub.

- Skeleton.

* RMI uses stub and skeleton object for communication with the remote object.

* A remote object is an object whose method can be invoked from another JVM.

Stub:-

The stub is an object, acts as a gateway for the client side. All the outgoing requests are routed through it. It resides at the client and represents the remote objects. When the caller invokes method on the stub object, it does the following tasks:

1. It initiates a connection with Java Virtual Machine (JVM).
2. It writes and transmits (marshals) the parameters to the remote Virtual Machine

3. It waits for the result.
4. It reads (unmarshals) the return value or exception, and
5. It finally, returns the value to the caller.

Skeleton:

The skeleton is an object, acts as a gateway for the server-side object. All the incoming requests are routed through it. When the skeleton receives the incoming request, it does the following tasks:

1. It reads the parameter for the remote method.
2. It invokes the method on the actual remote object, and
3. It writes and transmits (marshals) the result to the caller.

