Unit-5

Device Drivers and Inter-process Communication

Device Management is another important function of the operating system. Device management is responsible for managing all the hardware devices of the computer system. It may also include the management of the storage device as well as the management of all the input and output devices of the computer system. It is the responsibility of the operating system to keep track of the status of all the devices in the computer system. The status of any computing devices, internal or external may be either free or busy. If a device requested by a process is free at a specific instant of time, the operating system allocates it to the process.

Device management in operating system known as the management of the I/O devices such as a keyboard, magnetic tape, disk, printer, microphone, USB ports, scanner, camcorder etc., as well as the supporting units like control channels.

The main functions of device management in the operating system

An operating system manages communication with the devices through their respective drivers. The operating system component provides a uniform interface to access devices of varied physical attributes. For device management in operating system:

- Keep tracks of all devices and the program which is responsible to perform this is called I/O controller.
- Monitoring the status of each device such as storage drivers, printers and other peripheral devices.
- Enforcing preset policies and taking a decision which process gets the device when and for how long.
- Allocates and deallocates the device in an efficient way. De-allocating them at two levels: at the process level when I/O command has been executed and the device is temporarily released, and at the job level, when the job is finished and the device is permanently released.
- Optimizes the performance of individual devices.

Types of devices

The OS peripheral devices can be categorized into 3: Dedicated, Shared, and Virtual. The differences among them are the functions of the characteristics of the devices as well as how they are managed by the Device Manager.

Dedicated devices:-

Such type of devices in the device management in operating system are dedicated or assigned to only one job/process at a time until that job/process releases them. Devices like printers, tape drivers, plotters etc. demand such allocation scheme since it would be awkward if several users share them at the same point of time. The disadvantages of such kind of devices is the inefficiency resulting from the allocation of the device to a single user for the entire duration of process execution even though the device is not put to use 100% of the time.

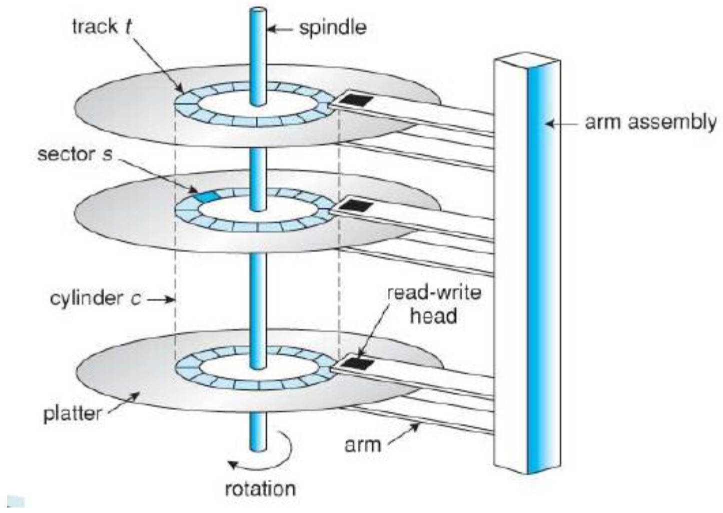
Shared devices:-

These devices can be allocated to several processes. Disk-DASD can be shared among several processes at the same time by interleaving their requests. The interleaving is carefully controlled by the Device Manager and all issues must be resolved on the basis of predetermined policies.

Virtual Devices:-

These devices are the combination of the first two types and they are dedicated devices which are transformed into shared devices. For example, a printer converted into a shareable device via spooling program which re-routes all the print requests to a disk. A print job is not sent straight to the printer, instead, it goes to the disk(spool)until it is fully prepared with all the necessary sequences and formatting, then it goes to the printers. This technique can transform one printer into several virtual printers which leads to better performance and use.

❖ Disk Scheduling



Disk Structure

A magnetic disk contains several platters. Each platter is divided into circular shaped tracks. Tracks are subdivided into sectors. A storage capacity of the disk is measured in GB. The disk rotates 60 to 200 times per second. The transfer rate is the rate at which data transfer between disk and computer.

Positioning time or random-access time is the time necessary to position the arm to the desired cylinder called the **seek time**. Time taken for the sector to rotate to the disk head is **rotational latencies.** Both seek time and rotational latencies are of several milliseconds.

The distance between the disk head and the platter is very less (microns) and the head sometimes comes in contact with the disk surface damaging it. This accident is called the **head crash**.

Basic Terms

- **Seek Time: –** Seek time is defined as the time which is used to find the disk arm to a stated track where the data must be read and write. The better disk algorithm is one that gives less average seek time.

- **Transfer Time: –** Transfer time is defined as the time which is used for data transfer. Transfer time depends on two things:
  1. Bytes which we have to transfer
  2. The rotation speed of the disk.

- **Rotational Latency: –** Rotational latency means the time which is needed to rotate the desired sector into a position so that a read/write heads can be accessed. The better disk scheduling algorithm is that which takes less rotational latency.
- **Disk Access Time: –**

$$\boxed{\text{Disk Access Time} = \text{Rotational Latency} + \text{Seek Time} + \text{Transfer Time}}$$

- **Disk Response Time: –** Disk response time is defined as the average time, which is used by an individual request, waiting for input/output operations. The best scheduling algorithm is one that gives less variance response.

➢ Disk Scheduling Algorithms

Disk scheduling algorithms are the algorithms that are used for scheduling a disk. Generally, the scheduling refers to a time-table for completing any task or a job. With the help of the operating system, disk scheduling is performed. We use disk scheduling to schedule the Input/output requests that arrive for the disk.

Disk scheduling is important because of the following reasons:

1. The slowest part of the computer system is the hard drive. So, to access the hard drive conveniently or effectively, we need disk scheduling.
2. There may be chances that two or more requests can be distant from each other. Therefore, more disk arm movement can happen. Thus, we need disk scheduling for such handling case
3. Sometimes, there are various I/O requests which arrive from the different processes. But at a time, the disk controller can only serve one I/O request. So, in this way, the other requests have to wait in the waiting queue, and scheduling is needed for those processes that are waiting in the waiting queue.

Objective of Disk Scheduling Algorithm

The objectives of the disk scheduling algorithm are:

1. Less traveling head time.
2. Fairness.
3. Throughput must be high.

Purpose of Disk Scheduling

The purpose of disk scheduling is to choose a disk request from the input/output requests queue and then scheduling the remaining request means which has to be processed.

➢ FCFS (First-Come-First-Serve) Disk Scheduling Algorithm

FCFS (First-Come-First-Serve) is the easiest disk scheduling algorithm among all the scheduling algorithms. In the FCFS disk scheduling algorithm, each input/output request is served in the order in which the requests arrive. In this algorithm, starvation does not occur because FCFS address each request.

Advantages of FCFS Disk scheduling Algorithm

1. In FCFS disk scheduling, there is no indefinite delay.
2. There is no starvation in FCFS disk scheduling because each request gets a fair chance.

Disadvantages of FCFS Disk Scheduling Algorithm

1. FCFS scheduling is not offered as the best service.
2. In FCFS, scheduling disk time is not optimized.

Example of FCFS Disk Scheduling Algorithm

A disk contains 200 tracks (0-199) and the request queue contains track no: 93, 176, 42, 148, 27, 14, 180. The current position of the read/write head is 55. Calculate the total number of track movements of read/write head using FCFS scheduling.

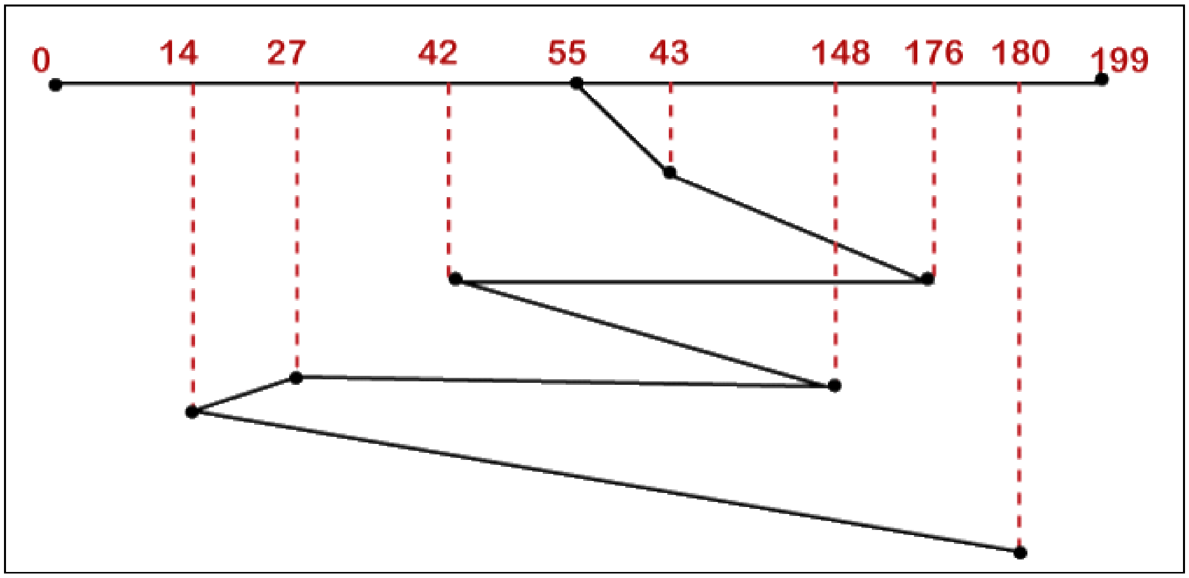


Figure: FCFS Disk Scheduling

Total Number of cylinders moved by the head = (176-55) + (176-42) + (148-42) + (148-14) + (180-14)

= 121+134+106+134+166

=661

➢ SSTF (Shortest Seek Time First) Disk Scheduling Algorithm

SSTF is another type of scheduling algorithm. In this type of disk scheduling, the job which has less seek time will be executed first. So, in SSTF (shortest seek time first) scheduling, we have to calculate the seek time first. And after calculating the seek time, each request will be served on the basis of seek time. The request which is close to the disk arm will be first executed. There are some drawbacks in FCFS. To overcome the limitations that arises in the FCFS. SSTF scheduling is implemented.

Advantages of SSTF Disk Scheduling

1. In SSTF disk scheduling, the average response time is decreased.
2. Increased throughput.

Disadvantages of SSTF Disk Scheduling

1. In SSTF, there may be a chance of starvation.
2. SSTF is not an optimal algorithm.
3. There are chances of overhead in SSTF disk scheduling because, in this algorithm, we have to calculate the seek time in advanced.
4. The speed of this algorithm can be decreased because direction could be switched frequently.

Example of SSTF Disk Scheduling

Consider a disk that contains 200 tracks (0-199). The request queue includes track number 82, 170, 43, 140, 24, 16, 190, respectively. The current position of the read/write head is 50.

Solution

Seek Time: –Seek time is the time required to move the desired track.

Seek time = Destination – Source

or

= Source – Destination

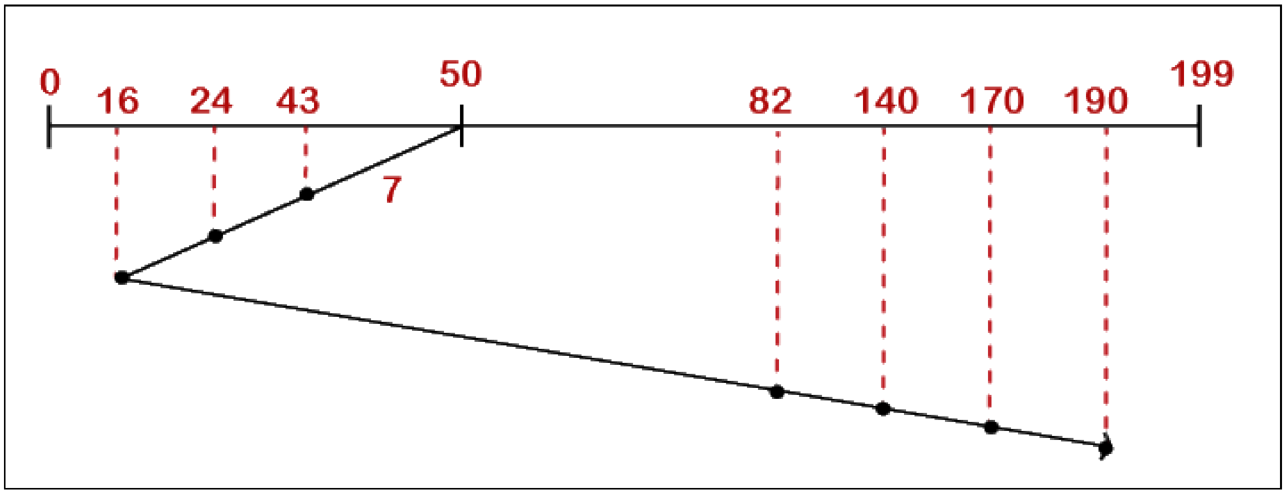


Figure: SSTF Disk Scheduling

Total Number of cylinders moved by the head = (50-16) + (190-16)
= 208

- **SCAN Disk Scheduling Algorithm**

The SCAN disk scheduling algorithm is another type of disk scheduling algorithm. In this algorithm, we move the disk arm into a specific direction (direction can be moved towards large value or the smallest value). Each request is addressed that comes in its path, and when it comes into the end of the disk, then the disk arm will move reverse, and all the requests are addressed that are arriving in its path. Scan disk scheduling algorithm is also called an elevator algorithm because its working is like an elevator.

Advantages of SCAN Disk Scheduling Algorithm

1. In SCAN disk scheduling, there is a low variance of response time.
2. In this algorithm, throughput is high.
3. Response time is average.
4. In SCAN disk scheduling, there is no starvation.

Disadvantages of SCAN Disk Scheduling Algorithm

1. SCAN disk scheduling algorithm takes long waiting time for the cylinders, just visited by the head.
2. In SCAN disk scheduling, we have to move the disk head to the end of the disk even when we don’t have any request to service.

Example of SCAN Disk Scheduling Algorithm

Consider a disk containing 200 tracks (0-199) and the request queue includes the track number 93, 176, 42, 148, 27, 14, 180, respectively. The current position of read//write head is 55, and direction is towards the larger value. Calculate the total number of cylinders moved by the head using SCAN disk scheduling.

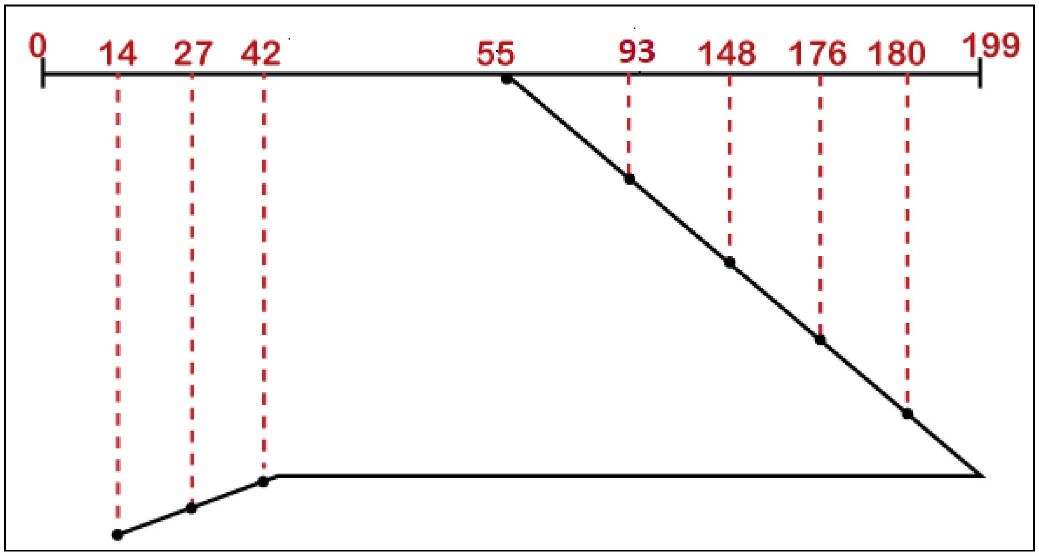


Figure: SCAN Disk Scheduling

Total Number of cylinders moved by head = (199-50) + (199-14)

= 329

- **C-SCAN Disk Scheduling Algorithm**

C-SCAN stands for Circular-SCAN. C-SCAN is an enhanced version of SCAN disk scheduling. In the C-SCAN disk scheduling algorithm, the disk head starts to move at one end of the disk and moves towards the other end and service the requests that come in its path and reach another end. After doing this, the direction of the head is reversed. The head reaches the first end without satisfying any request and then it goes back and services the requests which are remaining.

Advantages of C-SCAN Disk Scheduling Algorithm

1. C-SCAN offers better uniform waiting time.
2. It offers a better response time.

Disadvantages of C-SCAN Disk Scheduling Algorithm

1. In C-SCAN disk scheduling, there are more seek movements as compared to SCAN disk scheduling.
2. In C-SCAN disk scheduling, we have to move the disk head to the end of the disk even when we don't have any request to service.

Example of C-SCAN Disk Scheduling Algorithm

Consider, a disk contains 200 tracks (0-199) and the request queue contains track number 82, 170, 43, 140, 24, 16,190, respectively. The current position of R/W head is 50, and the direction is towards the larger value. Calculate the total number of cylinders moved by head using C-SCAN disk scheduling.

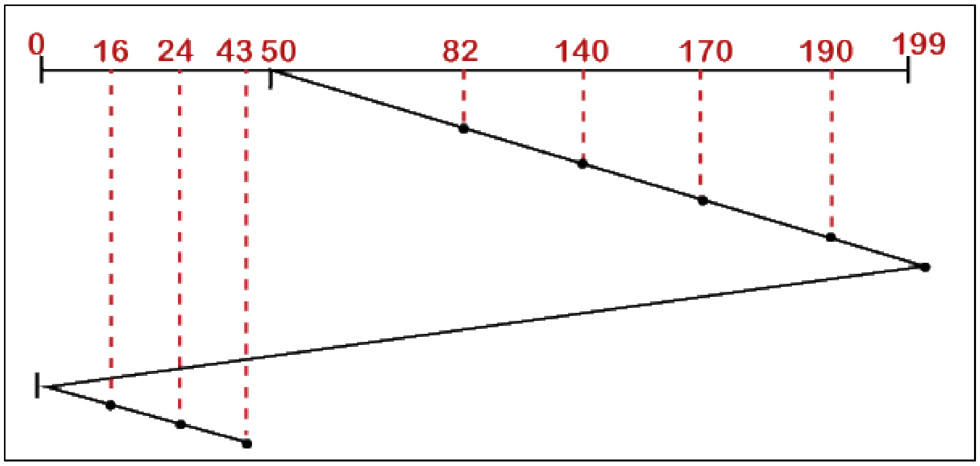


Figure: C-SCAN Disk Scheduling

Total Number of cylinders moved by head = (199-50) + (199-0) + (43-0)
= 149+199+43
= 391

➢ Look Disk Scheduling

Look disk scheduling is another type of disk scheduling algorithm. Look scheduling is an enhanced version of SCAN disk scheduling. Look disk scheduling is the same as SCAN disk scheduling, but in this scheduling, instead of going till the last track, we go till the last request and then change the direction.

Advantages of Look Disk Scheduling

1. In Look disk scheduling, there is no starvation.

2. Look disk scheduling offers low variance in waiting time and response time.
3. Look disk scheduling offers better performance as compared to the SCAN disk scheduling.
4. In look disk scheduling, there is no requirement of disk head to move till the end to the disk when we do not have any request to be serviced.

Disadvantages of Look Disk Scheduling

1. In look disk scheduling, there is more overhead to find the end request.
2. Look disk scheduling is not used in case of more load.

Example of Look Disk Scheduling

Consider a disk contains 200 tracks (0-100). The request queue includes track number 82, 170, 43, 140, 24, 16, 190, respectively. The current position of the read/write head is 50. The direction is towards the larger value. Calculate the total number of cylinders moved by head using look disk scheduling.

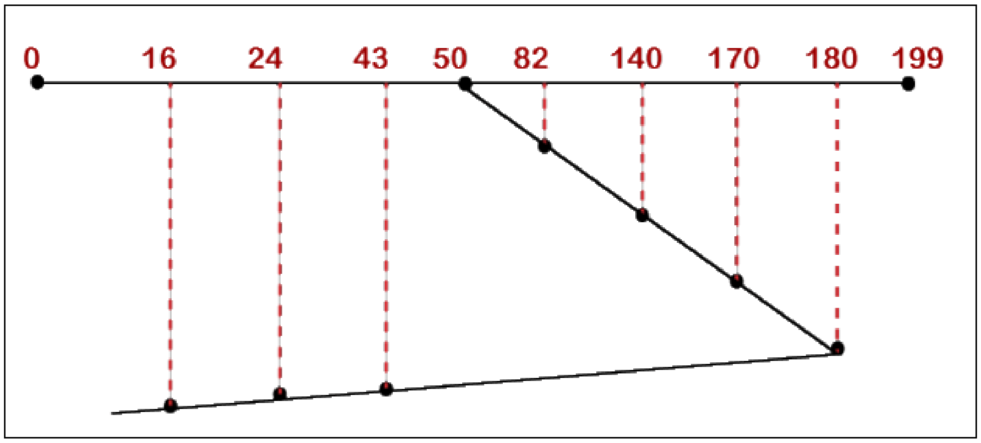


Figure: Look Disk Scheduling

Total number of cylinders moved by the head = (190-50) + (190-16)
= 314

➢ C-Look Disk Scheduling

C-look means circular-look. It takes the advantages of both the disk scheduling C-SCAN, and Look disk scheduling. In C-look scheduling, the disk arm moves and service each request till the head reaches its highest request, and after that, the disk arm jumps to the lowest cylinder without servicing any request, and the disk arm moves further and service those requests which are remaining.

Advantages of C-Look Disk Scheduling

1. There is no starvation in C-look disk scheduling.
2. The performance of the C-Look scheduling is better than Look disk scheduling.
3. C-look disk scheduling offers low variance in waiting time and response time.

Disadvantages of C-Look Disk Scheduling

1. In C-Look disk scheduling there may be more overhead to determine the end request.
2. There is more overhead in calculations.

Example of C-Look Disk Scheduling

Consider a disk containing 200 tracks (0-100). The request queue contains the track number 93, 176, 42, 148, 27, 14,183 respectively. The current position of the R/W head is 55. The direction is towards the larger value. Calculate the total number of cylinders moved by head using look disk scheduling.

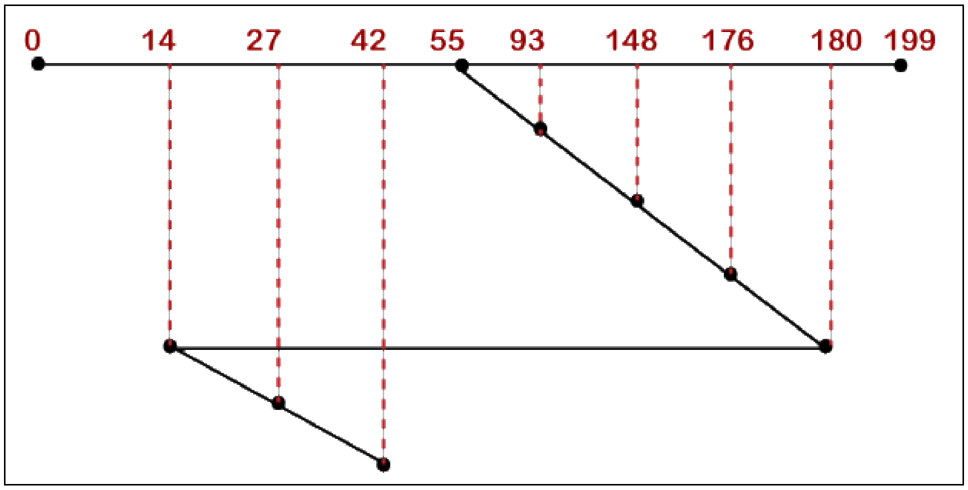


Figure: C-Look Disk Scheduling

Total number of cylinders moved by the head = (180-55) + (180-14) + (42-14)
= 125 + 166 + 28
= 319

❖ Inter Process Communication

Inter Process Communication is a mechanism which allows processes to communicate with each other and synchronize their actions. Whatever process is present in the system, they can communicate with each other. It is a method of cooperation.

There are two types of processes –

1. Independent process: An Independent process is not affected by other executing processes.

2. Cooperating process: A Cooperating process can be affected by other executing processes.

For inter process communication, it is compulsory that all the processes are cooperating processes.

Some of the benefits of using inter process communication are –

1. ***Information Sharing:*** Multiple processes can share same information to perform some tasks. In such scenarios, inter process communication helps. There may be a scenario when a process needs to access remote process. In such case, this method of communication helps.

2. ***Communication speed:*** Computational speed will also increase if inter process communication method is used to communicate between processes.

3. ***Modularity***: An architecture is break down into different cooperating modules to increase the efficiency. All the modules cooperate using inter process communication method.

Processes can communicate with each other by using two ways –

1. Shared Memory

2. Message Memory

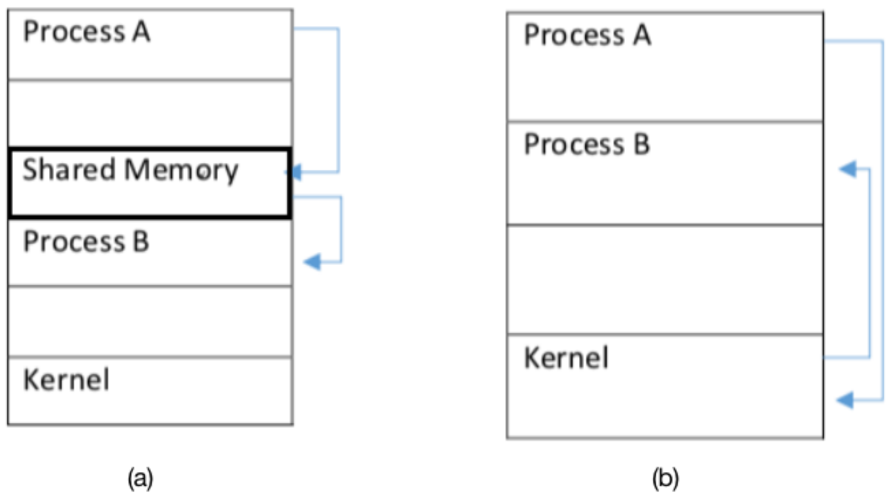
1. Shared Memory

Shared memory is an efficient way to share data between processes. One process will create a memory portion which other processes can access if allowed.

Example:

Assume that there are two processes – Process A and Process B.

Both processes communicate using shared memory as shown below.



Shared Memory Example

There are two parts (a) and (b). Both represent shared memory techniques.

Image (a)

Process A generate information about certain resources and keeps records in shared memory. When process B needs to use that information, it will check the record stored in shared memory and take note of the information generated by process A and act accordingly. Thus, processes can use shared memory for extracting information as a record from other process as well as for delivering any specific information to other process.

Image (b)

Whenever process A uses some shared memory, it sends information to the kernel (operating system). When process B wants to perform some operation, it first checks the Kernel if any other device is using that resources or not. If any process is using that resource, it will take other resources which is free.

2. Message Passing

In message passing, there is no use of shared memory. If two processes A and B want to communicate with each other, at first, they establish a communication link. After this, they can start exchanging messages using basic primitives. They need atleast 2 basic primitives –

(a) Send (message, destination) or Send(message)
(b) Receive (message, host) or Receive(message)

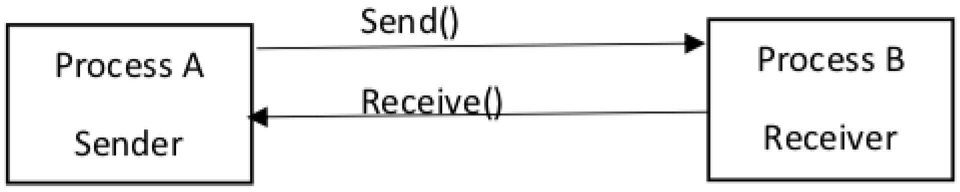
A standard message has two parts –

(a) Header
(b) Body

Header contains message type, source id, destination id, message length and control information. Control information contains sequence number, priority, action to do if runs out of space etc.

Body contains the actual message.

Generally, any message is sent using FIFO style.



IPC example

In this call, the sender and receiver processes address each other by names.

Mode of communication between two processes can take place in two ways:

1. Direct Addressing

2. Indirect Addressing

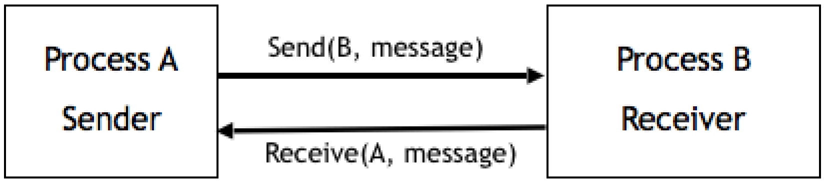
1. Direct Addressing of Message Passing

In this type, the two processes need to know the name of each other to communicate. This become easy if they have the same parent.

Example:
If process A send message to process B, then, basic primitives will be –

Send (B, message);

Receive (A, message);

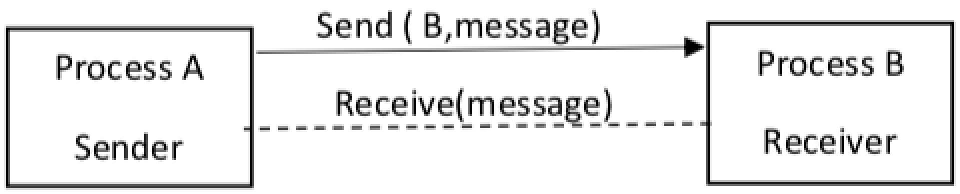


Direct Addressing Example

At first, Processes A and B establish a communication links. Then, they start sending messages. In this example, process A is sending message to process B. So, it will send **Send(B, message)**. In other case, if process A wants to receive message, it will receive using basic primitives **Receive(A, message)**.

A link is established automatically between every pair of process that want to communicate. Process only need to know each other's identity. In this type, link is associated with exactly two processes. Usually link is bidirectional but it can be unidirectional. Here, receiver knows the identity of sender. This type of direct communication is known as **symmetric addressing.**

Asymmetric addressing: In this type, only sender will name receiver for sending the message. There is no need for receiver to name the sender for receiving the message. This addressing is also known as asynchronous addressing.



Asymmetric Addressing Example

Here, processes A and B are communicating through Asymmetric Addressing. Note that Process A is sending message using Send(B, message) but receiving message using Receive(message) instead of Receive(A, message).

2. Indirect Addressing of Message Passing

In this addressing, message is sent and received using a mailbox. A mailbox can be abstractly viewed as an object into which message may be placed. Message may be extracted by a process.

In this type, sender and receiver processes should share a mailbox to communicate.

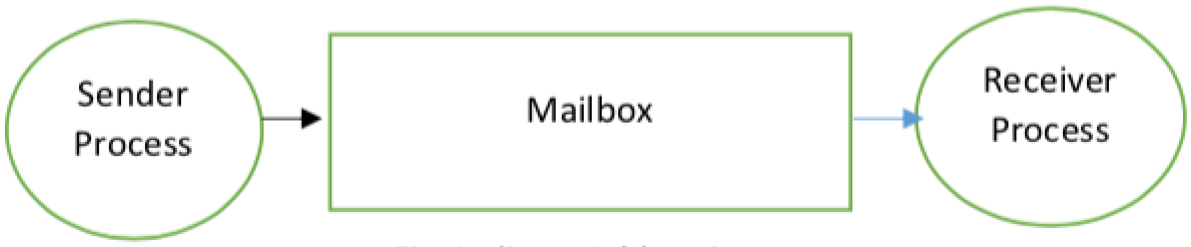


Fig: Indirect Addressing

Indirect Addressing Example

Below are the types of communication link made through mailbox:

1. **One to one link:** One sender wants to communicate with one receiver. So, only one link will be established.

2. **Many to one link:** Multiple senders want to communicate with single receiver. Example: In client-server system, there are one server process and many client processes. Here, mailbox is known as a port.

3. **One to Many link:** One sender wants to communicate with multiple receivers. That is broadcasting a message.

4. **Many to Many link:** Multiple senders want to communicate with multiple receivers.

Examples of IPC systems

Some examples of Inter process communication are –

- Posix : uses shared memory method.
- Mach : uses message passing
- Windows XP : uses message passing using local procedural calls

Communication in client/server Architecture:

There are various mechanism:

- Pipe
- Socket
- Remote Procedural calls (RPCs)