



SNS COLLEGE OF TECHNOLOGY
DEPARTMENT OF AIML
23CST202- OPERATING SYSTEMS
II YEAR IV SEM AIML-B



UNIT 1 – Process Management
TOPIC – Interprocess Communication

Processes need to communicate with each other in many situations. Inter-Process Communication or IPC is a mechanism that allows processes to communicate. It helps processes synchronize their activities, share information, and avoid conflicts while accessing shared resources.

Types of Process

Let us first talk about types of processes:

- **Independent process:** An independent process is not affected by the execution of other processes. Independent processes do not share any data or resources with other processes. No inter-process communication is required in this case.
- **Co-operating process:** Interact with each other and share data or resources. A co-operating process can be affected by other executing processes. Inter-process communication (IPC) is a mechanism that allows processes to communicate with each other and synchronize their actions. The communication between these processes can be seen as a method of cooperation between them.

Inter Process Communication

Inter process communication (IPC) allows different processes running on a computer to share information with each other. IPC allows processes to communicate by using different techniques like sharing memory, sending messages or using files. It ensures that processes can work together without interfering with each other. Cooperating processes require an Inter Process Communication (IPC) mechanism that will allow them to exchange data and information.

The two fundamental models of Inter Process Communication are:

- **Shared Memory**
- **Message Passing**

An operating system can implement both methods of communication. First, we will discuss the shared memory methods of communication and then message passing. Communication between processes using shared memory requires processes to share some variable and it completely depends on how the programmer will implement it. Suppose process 1 and process 2 are executing simultaneously and they share some resources or use some information from another process. Process1 generates information about certain computations or resources being used and keeps it as a record in shared memory. When process 2 needs to use the shared information, it will check in the record stored in shared memory and take note of the information generated by process 1 and act accordingly. Processes can use shared memory for extracting information as a record from another process as well as for delivering any specific information to other processes.

Figure 1 below shows a basic structure of communication between processes via the shared memory method and via the message passing method.

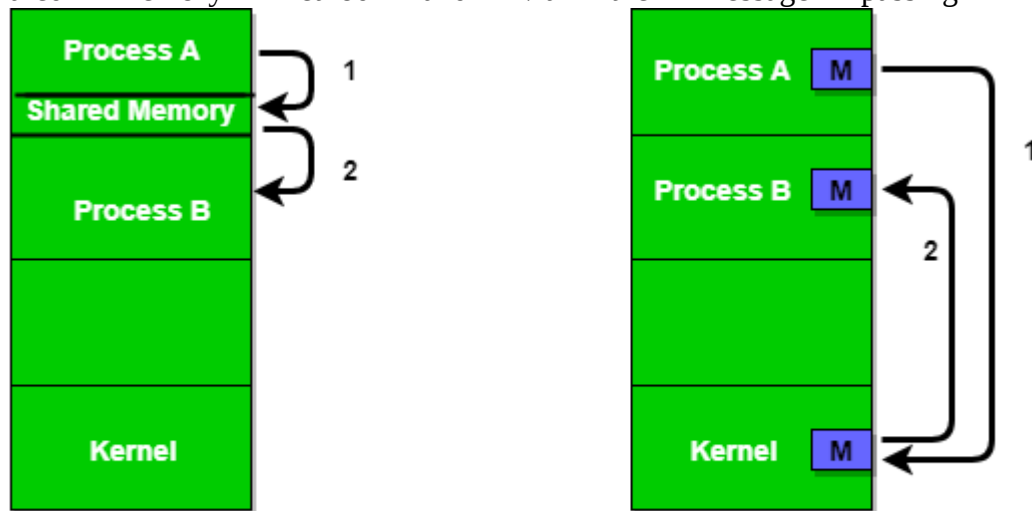


Figure 1 - Shared Memory and Message Passing

Role of Synchronization in IPC

1. Preventing Race Conditions

In a multi-process environment, multiple processes may attempt to access shared data or resources at the same time. Without proper synchronization, this can lead to race conditions, where the outcome depends on the non-deterministic order in which processes access the resource. Synchronization mechanisms like mutexes, semaphores, and locks ensure that only one process can access a resource at a time, preventing inconsistent or corrupted data.

2. Ensuring Mutual Exclusion

Mutual exclusion (mutex) is a fundamental concept in synchronization. It ensures that only one process at a time can access critical section. This prevents conflicts or inconsistent results when multiple processes attempt to modify shared data simultaneously.

3. Coordinating Process Execution

It allows processes to wait for specific conditions to be met before proceeding. For example, one process may need to wait for data from another process before continuing. Condition variables and barriers are used in such cases to synchronize the execution order of processes.

4. Preventing Deadlocks

Deadlocks occur when two or more processes are waiting indefinitely for resources held by each other. Proper synchronization techniques, such as acquiring resources in a defined order or using deadlock detection and prevention mechanisms can help avoid situations of deadlock.

5. Communication Between Processes

In IPC, synchronization ensures that messages or data exchanged between processes are correctly received and processed. It coordinates the flow of data and ensures that a producer process doesn't overwrite data before a consumer process can use it or the consumer doesn't attempt to consume data that isn't yet produced.

6. Fairness

It ensures that all processes have an equal opportunity to access shared resources. This prevents starvation where some processes are indefinitely delayed while others

continuously acquire resources. Techniques such as round-robin scheduling and fair locks can be used to ensure that no process is unfairly delayed.

Methods in Inter process Communication

Inter-Process Communication refers to the techniques and methods that allow processes to exchange data and coordinate their activities. Since processes typically operate independently in a multitasking environment, IPC is necessary for them to communicate effectively without interfering with one another. There are several methods of IPC, each designed to suit different scenarios and requirements. These methods include shared memory, message passing, semaphores, and signals, etc.

[Read more about methods of Inter Process Communication.](#)

Advantages of IPC

- It enables processes to communicate with each other and share resources, leading to increased efficiency and flexibility.
- It facilitates coordination between multiple processes and leads to better overall system performance.
- It allows for the creation of distributed systems that can span multiple computers or networks.
- It can be used to implement various synchronization and communication protocols, such as semaphores, pipes, and sockets.

Disadvantages of IPC

- It increases system complexity, making it harder to design, implement, and debug.
- It can introduce security vulnerabilities, as processes may be able to access or modify data belonging to other processes.
- It requires careful management of system resources such as memory and CPU time, to ensure that IPC operations do not degrade overall system performance.
- It can lead to data inconsistencies if multiple processes try to access or modify the same data at the same time.