

Search...

[Aptitude](#)[Engineering Mathematics](#)[Discrete Mathematics](#)[Operating System](#)[DB](#)[Sign In](#)

Inter Process Communication (IPC)

Last Updated : 23 Apr, 2025

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**

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

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.

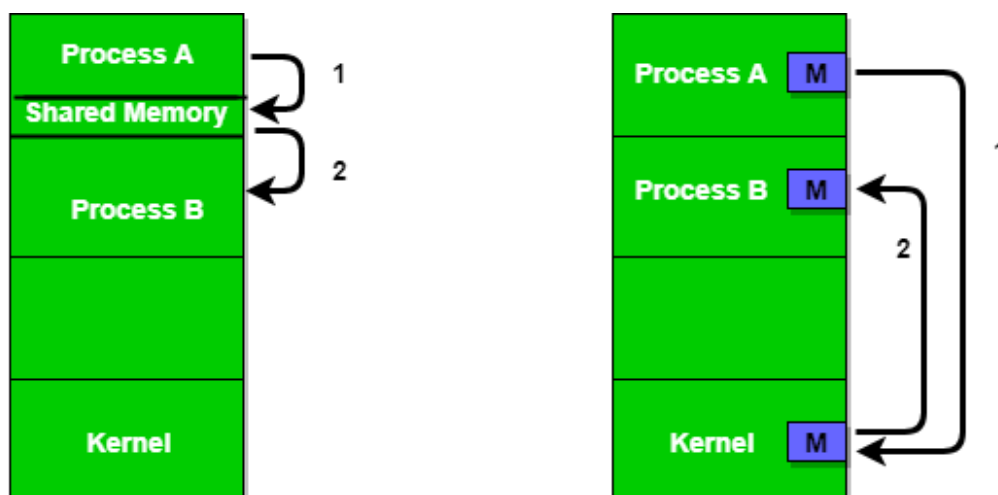


Figure 1 - Shared Memory and Message Passing

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

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

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

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.

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

- 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.

[Comment](#)[More info](#)[Advertise with us](#)

Next Article

Context Switching in Operating
System

Similar Reads

Methods in Inter process Communication

Inter Process communication (IPC) refers to the mechanisms and techniques used by operating systems to allow different processes to...

15+ min read

Interprocess Communication in Distributed Systems

Interprocess Communication (IPC) in distributed systems is crucial for enabling processes across different nodes to exchange data and...

15+ min read

Message based Communication in IPC (inter process communication)

Prerequisites - Cloud computing, Load balancing in Cloud Computing, Inter-process Communication In the development of models and...

15+ min read

Message Passing Model of Process Communication

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

15+ min read

Client Server Communication in Operating System

In an Operating System, Client Server Communication refers to the exchange of data and Services among multiple machines or processes. I...

15+ min read

How Communication happens using OSI model

The Open System Interconnection (OSI) model is a standard “reference model” created by an International Organization for Standardization (ISO...

15+ min read

Communication to I/O Devices in Operating System

The foundation of efficient computing rests on robust interaction between users and an operating system through Input/Output (I/O) devices in...

15+ min read

Cooperating Process in Operating System

Pre-requisites: Process Synchronization In an operating system, everything is around the process. How the process goes through several...

14 min read

Socket in Computer Network

A socket is one endpoint of a two way communication link between two programs running on the network. The socket mechanism provides a...

12 min read

Classical IPC Problems

Inter-Process Communication (IPC) is necessary for processes to communicate and share data. While basic communication between...

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Corporate & Communications Address:**

A-143, 7th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305)

Registered Address:

K 061, Tower K, Gulshan Vivante
Apartment, Sector 137, Noida, Gautam
Buddh Nagar, Uttar Pradesh, 201305



Advertise with us

Company

About Us
Legal
Privacy Policy
Careers
In Media
Contact Us
GfG Corporate Solution
Placement Training Program

Languages

Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial

Data Science & ML

Data Science With Python

Explore

Job-A-Thon Hiring Challenge
GfG Weekly Contest
Offline Classroom Program
DSA in JAVA/C++
Master System Design
Master CP
GeeksforGeeks Videos

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
DSA Interview Questions
Competitive Programming

Web Technologies

HTML

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Pandas
NumPy
NLP
Deep Learning

NextJS
NodeJs
Bootstrap
Tailwind CSS

Python Tutorial

Python Programming Examples
Django Tutorial
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question

DevOps

Git
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar

Preparation Corner

Company-Wise Recruitment Process
Aptitude Preparation
Puzzles
Company-Wise Preparation

Machine Learning/Data Science

Complete Machine Learning & Data Science Program - [LIVE]
Data Analytics Training using Excel, SQL, Python & PowerBI -
[LIVE]

Computer Science

GATE CS Notes
Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

Databases

SQL
MYSQL
PostgreSQL
PL/SQL
MongoDB

More Tutorials

Software Development
Software Testing
Product Management
Project Management
Linux
Excel
All Cheat Sheets

Programming Languages

C Programming with Data Structures
C++ Programming Course
Java Programming Course

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).