

...cont'd

1

Process and Process Management

12/18/2025

Inter Process Communication (IPC)

2

IPC is the mechanism provided by the operating system that allows processes to communicate with each other.

This communication could involve a process **letting** another process know that some event has occurred or the transferring of data from one process to another.

IPC enables processes to communicate with each other to share information

■ Advantages of IPC -- **Information sharing,**

Inter Process Communication (IPC)

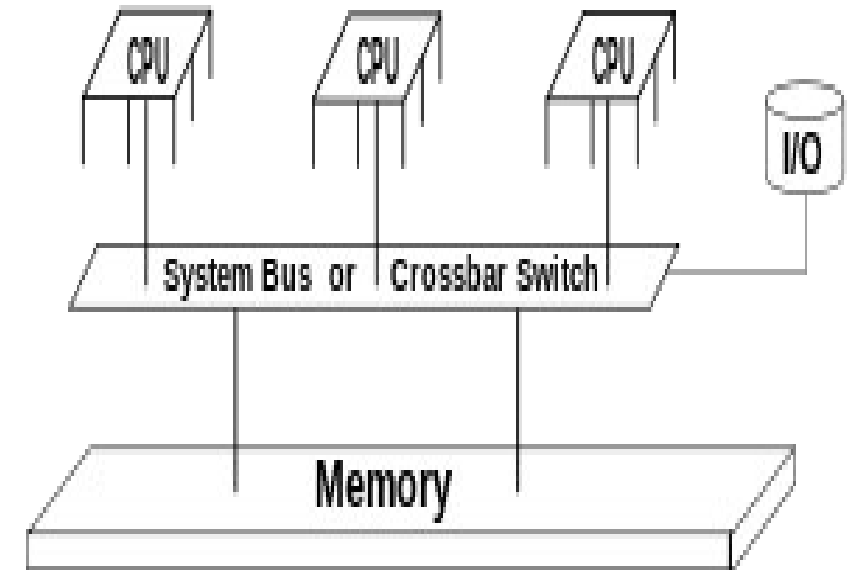
3

3 ways – Shared memory –
Message Passing – Signals

■ Shared Memory

One process will create an area in RAM which the other process can access. Both processes can access shared memory like a regular working memory

Limitation : Error prone. Needs synchronization between



12/18/2025

IPC (cont'd)

4

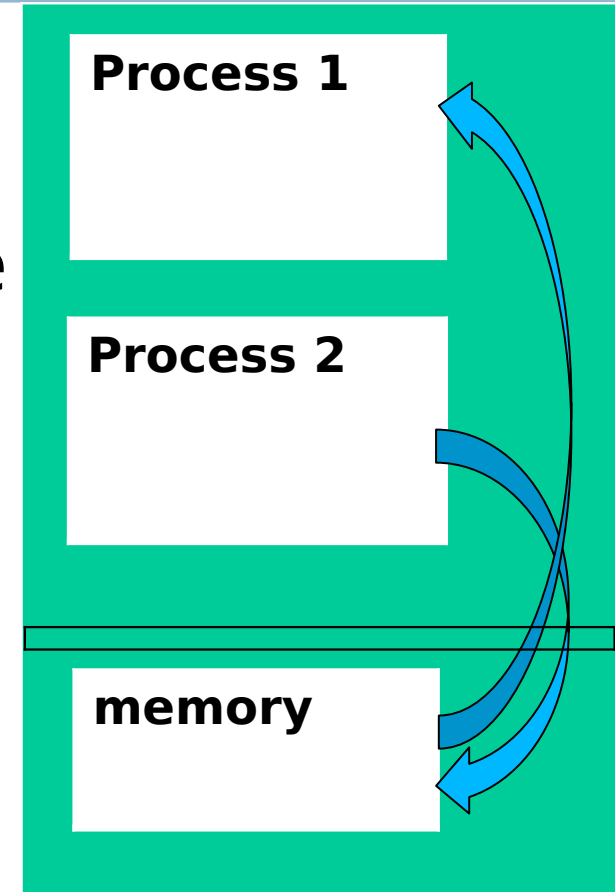
Message Passing

- Shared memory created in the kernel
- System calls such as send and receive used for communication

-> Cooperating : each send must have a receive

Advantage : Explicit sharing, less error prone

Limitation : Slow.



IPC (cont'd)

5

Signals

- Asynchronous unidirectional communication between processes
- Signals are a small integer – eg. 9: kill, 11: segmentation fault
- Send a signal to a process – `kill(pid, signum)`
- Process handler for a signal
 - `sighandler_t signal(signum, handler);`
 - Default if no handler defined

12/18/2025

Process

Synchronization

There are two types of Processes in an Operating Systems:-

1. Independent Process - The process that does not affect or is affected by the other process while its execution then the process is called Independent Process.

Ex. The process that does not share any shared variable, database, files, etc.

2. Cooperating Process - The process that affect or is affected by the other process while execution, is called a Cooperating Process.

Process Synchronization

(cont'd)

- Processes can execute concurrently
 - May be interrupted at any time, partially completing execution
- Concurrent access to shared data may result in data inconsistency
- Maintaining data consistency requires mechanisms to ensure the orderly execution of cooperating processes

Illustration of the problem:

Suppose that we wanted to provide a solution to the producer – consumer problem that fills all the buffers. We can do so by having an integer counter that keeps track of the number of full buffers. Initially, counter is set to 0. It is incremented by the producer after it produces a new buffer

Process Synchronization (cont'd)

8

- **Race Conditions** is a situation that may occur inside a critical section. This happens when the result of multiple thread execution in critical section differs according to the order in which the threads execute.
- This condition can be avoided using the technique called Synchronization or **Process Synchronization**, in which we allow only one process to enter and manipulates the shared data in Critical Section.

Process Synchronization (cont'd)

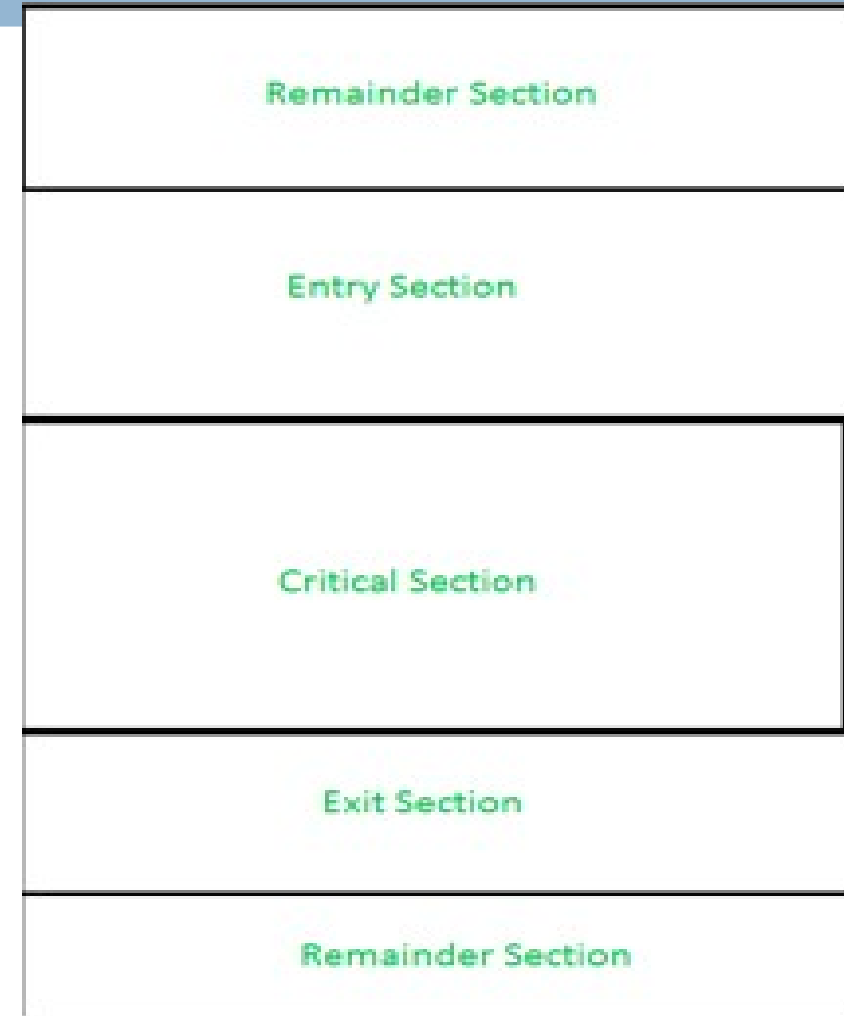
9

- Consider system of n processes ($p_0, p_1, \dots p_{n-1}$)
- Each process has **critical section** segment of code
 - Process may be changing common variables, updating table, writing file, etc
 - When one process in critical section, no other may be in its critical section
- **Critical section problem** is to design protocol to solve this

Process Synchronization (cont'd)

10

- Each process must ask permission to enter critical section in **entry section**, may follow critical section with **exit section**, then **remainder section**



12/18/2025

Process Synchronization (cont'd)

11

■ **Entry Section -**

It is part of the process which decide the entry of a particular process in the Critical Section, out of many other processes.

■ **Exit Section -**

It checks that a process that after a process has finished execution in Critical Section can be removed through this Exit Section and allowing the other process that are waiting in the Entry Section, to enter into the Critical Sections.

■ **Remainder Section -**

12/18/2025

Process Synchronization (cont'd)

12

Critical Section problems must satisfy these three requirements:

1. Mutual Exclusion - It states that no other process is allowed to execute in the critical section if a process is executing in critical section.

2. Progress - When no process is in the critical section, then any process from outside that request for execution can enter in the critical section without any delay. Only those process can enter that have requested and have finite time to enter the process.

3. Bounded Waiting - An upper bound must exist on the number of times a process enters so that other processes

12/18/2025

Process Synchronization (cont'd)

13

Classical Problems of Synchronization

Below are some of the classical problem depicting flaws of process synchronization in systems where cooperating processes are present.

- **Bounded Buffer (Producer-Consumer) Problem**
- **Dining Philosophers Problem**
- **The Readers Writers Problem**

Process Synchronization (cont'd)

14

Bounded Buffer Problem

- This problem is generalized in terms of the **Producer Consumer problem**, where a finite buffer pool is used to exchange messages between producer and consumer processes.
- Because the buffer pool has a maximum size, this problem is often called the Bounded buffer problem.
- Solution to this problem is, creating two counting **semaphores** (variable or abstract data type used to control access to a common resource) "full" and "empty" to keep track of the current number of full and empty buffers respectively.

12/18/2023

Process Synchronization (cont'd)

15

Dining Philosophers Problem

- The dining philosopher's problem involves the allocation of limited resources to a group of processes in a deadlock-free and starvation-free manner.
- There are five philosophers sitting around a table, in which there are five chopsticks/forks kept beside them and a bowl of rice in the centre,
- When a philosopher wants to eat, he uses two chopsticks - one from their left and one from their right. When a philosopher wants to think, he keeps down both chopsticks at their original place.

12/18/2025

Process Synchronization (cont'd)

16

The Readers Writers Problem

- In this problem there are some processes(called readers) that only read the shared data, and never change it, and there are other processes(called writers) who may change the data in addition to reading, or instead of reading it.
- There are various type of readers-writers problem, most centred on relative priorities of readers and writers.

Process Synchronization (cont'd)

17

Reading Assignment

- Hardware and Software Approaches to process synchronization
- Tools that are used to solve process synchronization problems