# Inter Process Communication (IPC)

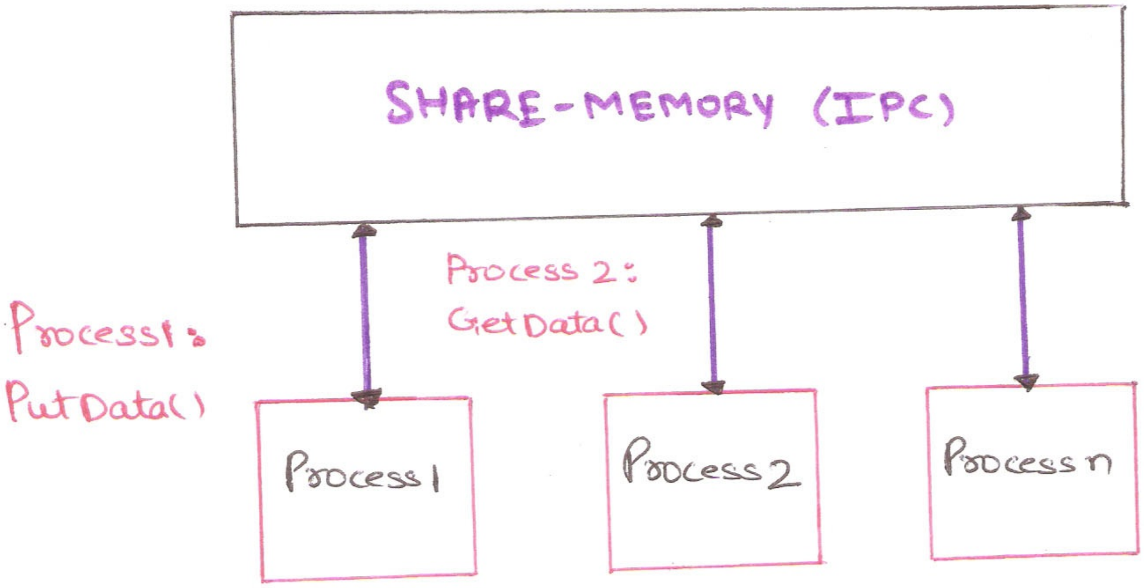
## Race Conditions:-

### What is Inter Process Communication ?

IPC is a Capability Supported by operating System that allows One process to Communicate with another process.

The processes Can be running on the Same Computer or on different Computers Connected through a network.

IPC enables One application to Control another application, and for Several applications to share the Same data without interfering with one another.



We need a well Structured way to facilitate interprocess Communication which

→ which Maintain integrity of the System.

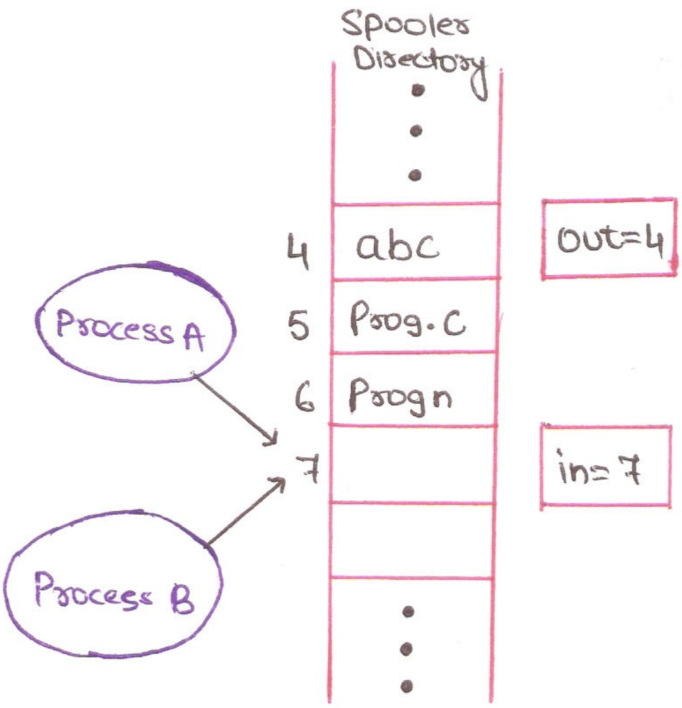
→ Ensure predictable behavior.

# RACE CONDITIONS:-

The situation where two or more processes are reading or writing some shared data & the final results depends on who runs precisely when are called RACE CONDITIONS.

Lets take an example:-

A print spooler:- When a process wants to print a file, it enters the file name in a special spooler directory. Another process, the Printer Daemon, periodically checks to see if there are any files to be printed, and if there are, it prints them and removes their names from the directory.



~~Ex~~ Imagine that our spooler directory has a large no. of slots, numbered 0,1,2, ... each one capable of holding a file name. Also imagine that there are two shared variables,

out: Points to next file to be printed.

in: Points to next free slot in the directory

Slots 0 to 3 Files already printed

Slots 4 to 6 Files Names which has to be printed.

## Now The Main issue Comes:

Process A reads in and stores the value, 7, in a local variable [illegible] the CPU decides that process A has run long enough, so it switches to process B.

Process B also reads in, and also gets a 7, so it stores the name of its in slot 7 and updates in to be an 8. Then it goes off and does other things.

Eventually, process A runs again, starting from the place it left off last time, It looks 'next-free slot', finds a 7 there, and writes its file name ~~7~~ in slot 7, erasing the name that process B just put there. Then it computes next-free_slot+1 which is 8, and sets in to 8.

The spooler directory is now internally consistent, so the printer daemon will not notice anything wrong, but process B will never receive any output.