

Critical Section or Critical Region:- That part of the program

where the shared memory is accessed is called critical section.

To avoid race condition we need Mutual Exclusion.

Mutual Exclusion:- It is some way of making sure that if one process is using a shared variable or file, the other process will be excluded from doing the same things.

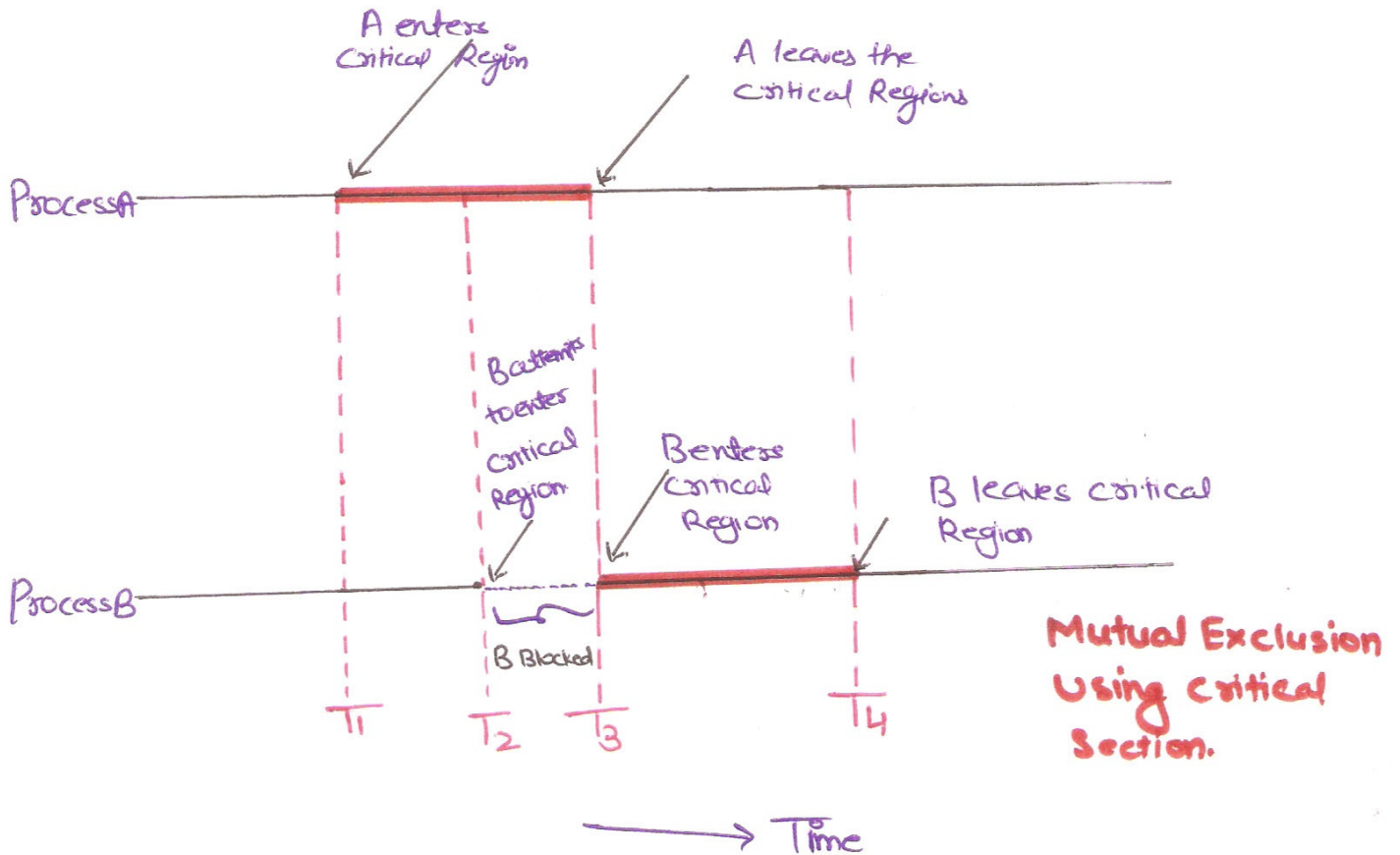
The difficulty in pointer spooler occurs because Process B started using one of the shared variables before process A was finished with it.

If we could arrange matters such that no two processes were ever in their critical regions at the same time, we could avoid race conditions.

Solution to Critical Section problem :

- 1) No two processes may be simultaneously inside their critical regions.
- 2) No assumptions may be made about speed or the number of CPUs.
- 3) No process running outside its critical region may block other processes.
- 4) No process should have to wait forever to enter its critical regions.

Process A enters its Critical Region at time T_1 . A little later, at time T_2 , Process B attempts to enter its Critical region but fails because another process is already in its Critical region and we allow one at a time only.



Consequently, B is temporarily suspend until process A **leaves the critical Region**

After Process A leaves the Critical Region, Process B enters the Critical Region.