Report

Implementing Critical Section:

To implement critical section, the three conditions 1.Mutual Exclusion 2.Progress 3.Bounded waiting needs to be satisfied.
1.Mutual Exclusion
2.Progress
3.Bounded waiting

- When we have Cooperating processes, we want them to synchronise without data inconsistency
- Consider system with n Threads({P0,P1,P2….Pn)
- Each Thread has a segment of code called a Critical Section
- In which the thread may be changing common variable, updating a task, writing a file and so on.
- Critical Section is a segment of code in a process will be changing common variables, updating, writing and so on in a shared region
- When one thread is changing data in shared region(executing in critical section), no other thread is to be allowed to execute in its critical section.

Concepts and their implementation in the program:

1. Mutual Exclusion:

Concept: If a process Pi is executing in its critical section, then no other processes can execute in the critical section

Implementation: For this I have implemented locks before and after critical section using MUTEX Locks. I have created Acquire lock and Release lock concept in the critical section in my program so that no two threads can enter into critical section at once.

2. Progress:

Concept: If no process executing in its critical section, then now if 2 or more threads wants to enter critical section, Only those processes that are not executing in remainder section can participate in decision making. This selection cannot be postponed indefinitely

Implementation: For example, I have created 3 threads in my program, if no thread is in CS, if 2 or more threads wants to enter CS, Which are not running in remainder section can have the ability to participate in decision making.

3. Bounded Waiting:

Concept: When a process is waiting to enter CS before which request has been granted there must be a limit on no of times that other processes are allowed to enter CS. After one thread has already made a request before its request has been granted. Without bounded waiting there will be starvation

Implementation: Here Bounded waiting is also implemented through program we can see that After one thread has already made a request before its request has been granted , mutex locks, where if a thread is waiting to enter CS before which request has been granted there must be a limit on no of times that other processes are allowed to enter CS.