

Paper Name : Operating System

Prepared by : Debanjali Jana

Introduction to process synchronization

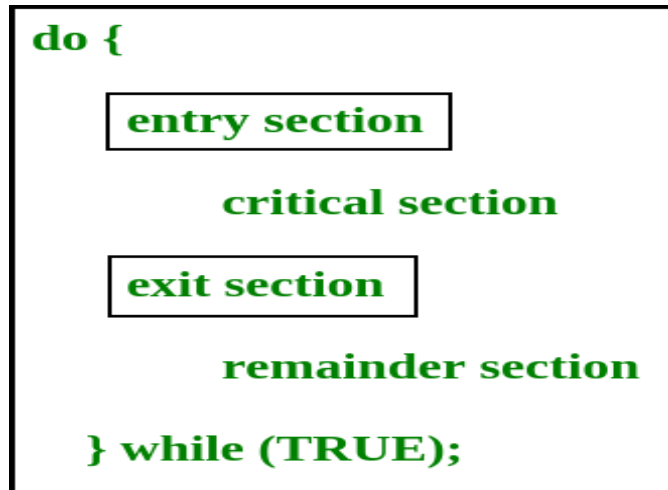
On the basis of synchronization, processes are categorized as one of the following two types:

- **Independent Process** : Execution of one process does not affect the execution of other processes.
- **Cooperating Process** : Execution of one process affects the execution of other processes.

Process synchronization problem arises in the case of cooperating process as resources are shared in cooperating processes. Concurrent access to shared data may result in data inconsistency. Maintaining data consistency requires mechanisms to ensure the orderly execution of cooperating processes.

Critical section problem

Each process has a segment of code, called a critical section which the process may be changing common variables, updating a table, writing a file, and so on. Thus the critical section is the part of a program where the shared variables can be accessed. The important feature of the system is that, when one process is executing in its critical section, no other process is to be allowed to execute in its critical section. That is, no two processes are executing in their critical sections at the same time. The critical-section problem is to design a protocol that the processes can use to cooperate. Each process must request permission to enter its critical section. The section of code implementing this request is the entry section. The critical section may be followed by an exit section. The exit section handles the exit from the critical section. The remaining code (i.e. other than shared variables/data) is the remainder section.



A solution to the critical-section problem must satisfy the following three requirements:

1. Mutual exclusion - If process P_i is executing in its critical section, then no other processes can be executing in their critical sections.
 2. Progress - If no process is executing in its critical section and some processes wish to enter their critical sections, then only those processes that are not executing in their remainder sections can participate in deciding which will enter its critical section next, and this selection cannot be postponed indefinitely.
 3. Bounded waiting - There exists a bound, or limit, on the number of times that other processes are allowed to enter their critical sections after a process has made a request to enter its critical section and before that request is granted.
-