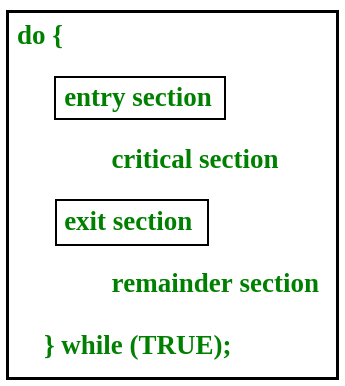
**Unit – III Process Synchronization**

* **Process Synchronization:**

**Process Synchronization** is the task of coordinating the execution of processes in a way that no two processes can have access to the same shared data and resources.

It is specially needed in a multi-process system when multiple processes are running together, and more than one processes try to gain access to the same shared resource or data at the same time. This can lead to the inconsistency of shared data. So the change made by one process not necessarily reflected when other processes accessed the same shared data. To avoid this type of inconsistency of data, the processes need to be synchronized with each other

****

There, are four essential elements of the critical section:

* **Entry Section:** It is part of the process which decides the entry of a particular process.
* **Critical Section:** This part allows one process to enter and modify the shared variable.
* **Exit Section:** Exit section allows the other process that are waiting in the Entry Section, to enter into the Critical Sections. It also checks that a process that finished its execution should be removed through this Section.
* **Remainder Section:**Allother parts of the Code, which is not in Critical, Entry, and Exit Section, are known as the Remainder Section.
* **Critical Section:**

A critical section is a segment of code which can be accessed by a signal process at a specific point of time. The section consists of shared data resources that required to be accessed by other processes.

* The entry to the critical section is handled by the wait() function, and it is represented as P().
* The exit from a critical section is controlled by the signal() function, represented as V().

In the critical section, only a single process can be executed. Other processes, waiting to execute their critical section, need to wait until the current process completes its execution.

Critical section problem must satisfy three requirements:

* **Mutual Exclusion** : If a process is executing in its critical section, then no other process is allowed to execute in the critical section.
* **Progress** : If no process is executing in the critical section and other processes are waiting outside the critical section, then only those processes that are not executing in their remainder section can participate in deciding which will enter in the critical section next, and the selection can not be postponed indefinitely.
* **Bounded Waiting** : A bound must exist 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.