## Critical Section

→ It is a part of the program where a shared resource is used. If multiple process enters it together, data may get computed or lost

→ Solution
- → Mutal Exclusion
- → Progress
- → Bounded waiting

## Monitor

→ A monitor is a tool in programming that helps different parts of a program take turns using a shared resource without interfering with each other

## Peterson's soln

→ Mutual Exclusion is comforted as at any time only one process can access the critical section

→ Process that is outside the critical section is unable to block other processes from entering.

→ Every process gets a fair chance to enter the critical section

## Authentication

→ It is the process of recognizing a user's identity. It is the mechanism of associating an incoming request with a set of identifying credentials

## Semaphore

→ Counting
→ Binary

→ It is an int variable, shared among multiple processes. The main aim of using a semaphore is process synchronization and access control for a common resource in a concurrent Environment

## Mutual - exclusion

→ It is a property of process synchronization which stats that "no two processes can exist in the critical section at any given point of time".

## Deadlock

→ It is a situation that occurs in OS when any process enters a waiting state becoz another waiting process is holding the demanded resource.

### condition of deadlock

- → Mutual Exclusion
- → Hold & wait
- → No preemption
- → Circular wait

## Recovery Technique

- → Process termination
- → Deadlock ignorance
- → Deadlock prevention
- → Deadlock avoidance
- → Deadlock detection & recovery

## ~~Recovery Technique~~

## → Segmentation

→ It is a memory management technique in which the memory is divided into the various size parts. Each part is known as segment which can be allocated to a process

| Monitor | Semaphore |
|---|---|
| → We can use cond^n variable only in this | → we can't use |
| → wait always block the caller | → wait doesn't always block the caller |
| → The monitor are comparised of the shared variable and the procedure which operate the shared variable | → The semaphore's value means the no of shared resources that are present in the system |
| → Cond^n variable are present in the monitor | → Cond^n variable are not present in the semaphore |

## Race cond^n

→ It occurs when multiple processes or thread attempt to access and modify shared data simultaneously, leading to unpredictable outcome

## RAG (Resource Allocation Graph)

→ It is a directed graph which can be used to represent the state of a system in the form of picture.

## Throughput

→ It is the measure of the amount of work a system can complete in a given period of time.

$$T = \frac{\text{Total Task completed}}{\text{Total time taken}}$$

## Starvation

→ It happens if a method is indefinitely delayed. This can emerge once a method need a further resource for execution that isn't assigned

→ It is the problem that occur when low priority process get jammed for an unspecified time as the high priority process keep executing