THIRUTHANGAL NADAR COLLEGE
knowledge is power

1

THIRUTHANGAL NADAR COLLEGE

(Belongs to the Chennaivazh Thiruthangal Hindu Nadar Uravinmurai Dharma Fund)

Selavayal, Chennai-51.

A Self-Financing Co-educational College of Arts & Science

Affiliated to the University of Madras

Accredited with 'B' Grade by NAAC

An ISO 9001: 2015 Certified Institution

NAME OF THE DEPARMENT: BCA(SHIFT-I)

SUBJECT : OPERATING SYSTEM

TOPIC : SYNCHRONIZATION,DEADLOCK& RECOVERY

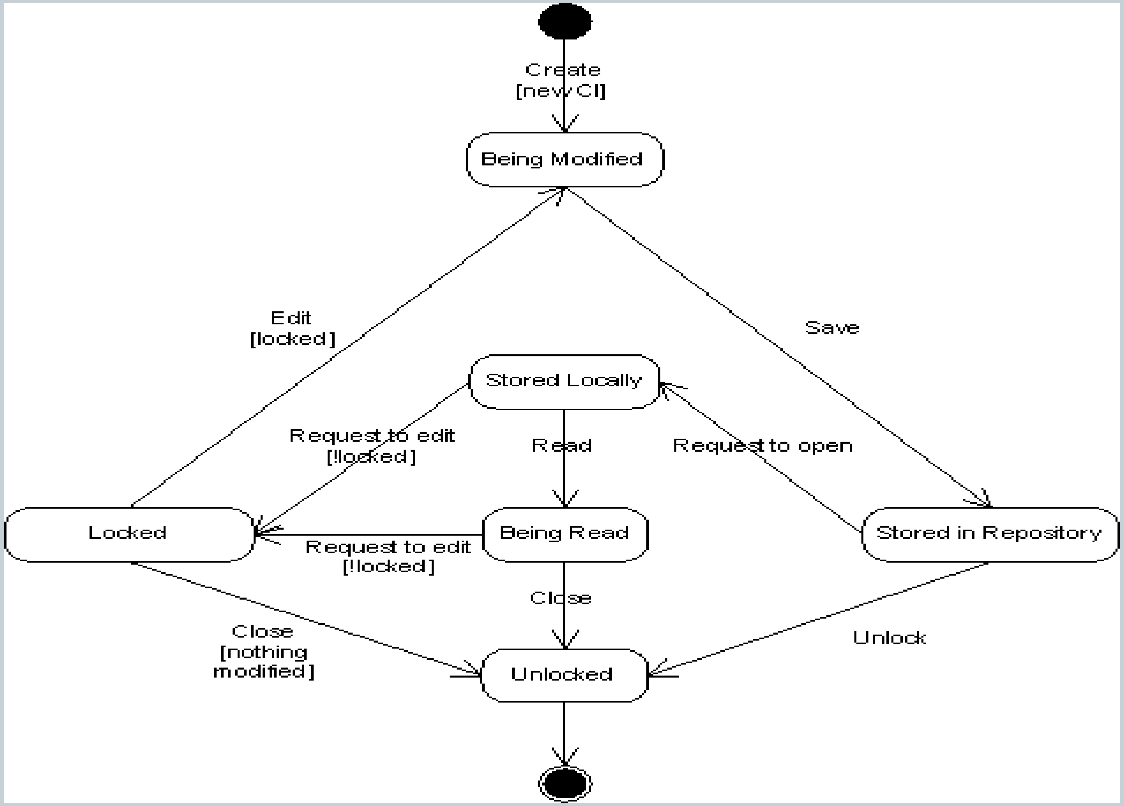
STAFF NAME :MR.K.SOMASUNDARAM

Contents

- Process Synchronization:
- Critical-Section problem
- Synchronization Hardware
- Semaphores
- Classic Problems of Synchronization
- Critical Region
- Deadlock Characterization
- Methods for handling Deadlocks
- Prevention, Avoidance, and Detection of Deadlock
- Recovery from deadlock

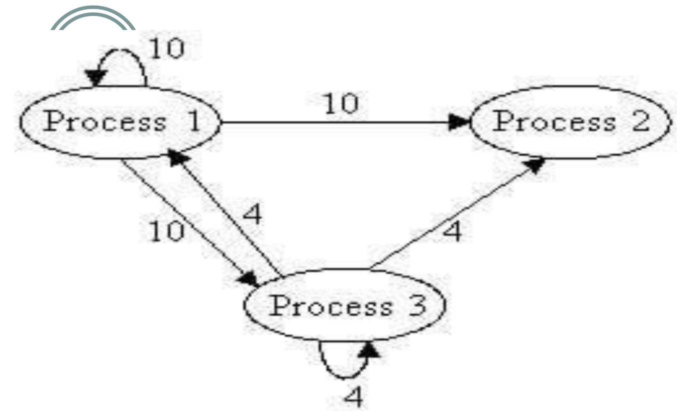
Process Synchronization

✓Process Synchronization means sharing system resources by processes in a such a way that, Concurrent access to shared data is handled thereby minimizing the chance of inconsistent data



Critical Section Problem

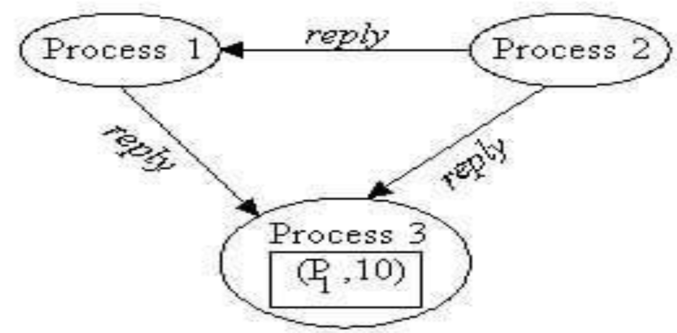
Processes 1 and 3 both want to enter the critical section, so they send request messages



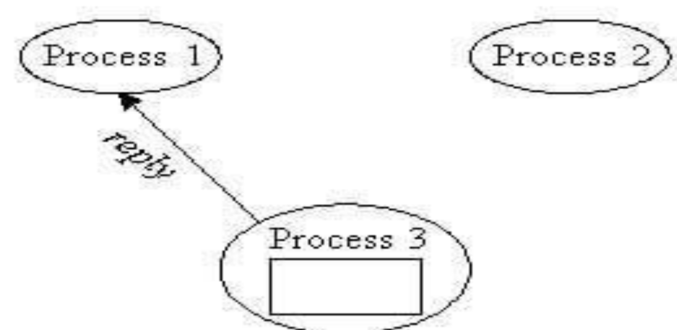
Processes 1 sends reply to Process 3

Process 2 replies to both Process 1 and 3

Process 3 queues Process 1's request.



Process 3 exits the critical section, dequeues Process 1's request, and sends a reply to Process 1

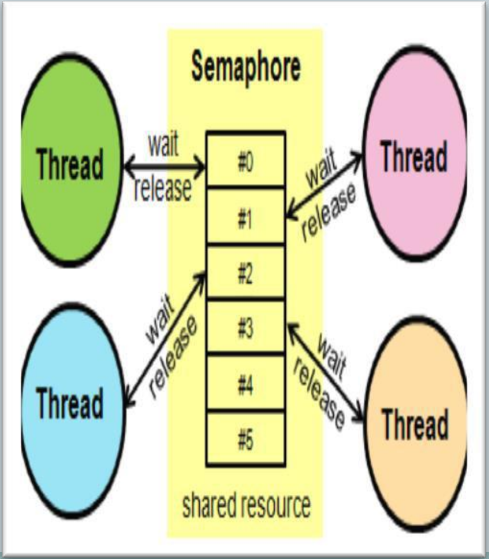


Semaphores

✓A semaphore is a variable or abstract data type used to control access to a common resource by multiple processes in a concurrent system such as a multiprogramming operating system.

✓concurrent accesses to shared resources can lead to unexpected or erroneous behavior, so parts of the program where the shared resource is accessed are protected. This protected **section** is the **critical section** or **critical** region.

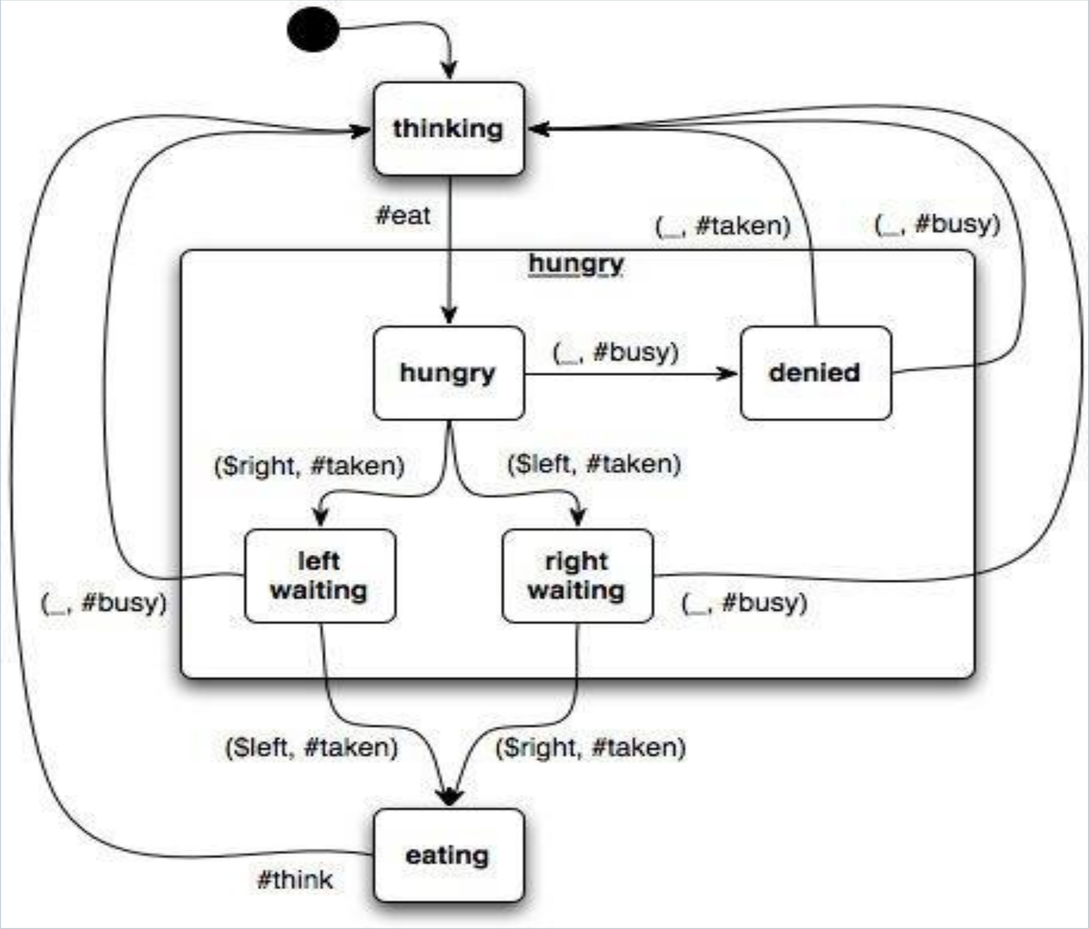
Semaphores



Critical Region

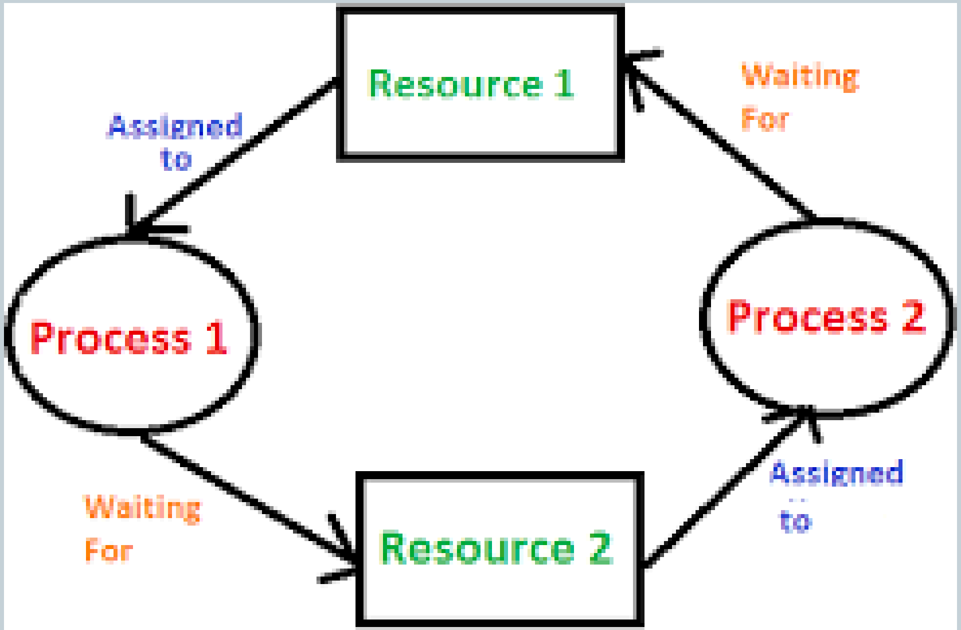
Classification Problem of Synchronization

✓ Dining Philosophers problem is an example of synchronization problem



Deadlock

- A set of blocked processes each holding a resource and waiting to acquire a resource held by another process in the set.
- Example
  - System has 2 tape drives
  - P1 and P2 each hold one tape drive and each needs another one.



https://www.youtube.com/watch?v=8dc6zz6fkDA

Deadlock Characterization

Deadlock can arise if four conditions hold simultaneously.

1. **Mutual exclusion**: at least one process must be held in a non-sharable mode.

2. **Hold and wait:** there must be a process holding one resource and waiting for another.

3. **No preemption:** resources cannot be preempted.

4. **Circular wait:** There must exist a set {p_0, p_1,.....p_n} of waiting processes such that p_0 is waiting for a resource which is held by p_1, p_1 is waiting for a resource which is held by p_2,..., p_{n-1} is waiting for a resource which is held by p_n and p_n is waiting for a resource which is held by p_0.

Methods for Handling Deadlock

- Ensure that the system will *never* enter a deadlock state.
- Allow the system to enter a deadlock state and then recover.
- Ignore the problem and pretend that deadlocks never occur in the system; used by most operating systems, including UNIX.

Prevention, Avoidance and Detection of Deadlock

- **Deadlock Prevention.**
  Disallow one of the four necessary conditions for deadlock.

- **Deadlock Avoidance.**
  Do not grant a resource request if this allocation have the potential to lead to a deadlock.

- **Deadlock Detection.**
  Always grant resource request when possible. Periodically check for deadlocks. If a deadlock exists, recover from it.

- **Ignore the problem...**
  Makes sense if the likelihood is very low.

Recovery from Deadlock

- **Process Termination**
  - Abort all deadlocked processes:
    - Fast
    - A lot of process work is lost.
  - Abort one deadlocked process at a time and check for deadlocks again:
    - More work to resolve a deadlock.
    - Better in terms of process work.
    - What is a good order to abort processes?
- **Resource Preemption**
  - what is a good way to select a victim
  - How can we rollback and then recover from preemption?
  - How can we protect from starvation