**VASAVI COLLEGE OF ENGINEERING**

**Ibrahimbagh, Hyderabad-500 031**

Department of Information Technology

**HAND OUT FOR B.E 1II/IV Students: 2017-18**

Sub: **Operating Systems**

**Name of the faculty: J.Suneetha Section: IT**

**---------------------------------------------------------------------------------------------------------------------**

**UNIT-III**

Inter Process Communication**, Process Synchronization** - Peterson's Solution, Bakery Algorithm, Semaphores, Critical Section, Monitors. Classical problems of synchronization. **Deadlocks:**  Deadlock prevention, deadlock avoidance and Deadlock Detection and Recovery - Bankers Algorithm.

**Inter Process Communication:**

A mechanism through which data is shared among the process in the system is referred to as Inter-process communication. Multiple processes communicate with each other to share data and resources.

There are two fundamental models of Inter-Process communication that are commonly used, these are:

1. Shared Memory Model

2. Message Passing Model

**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.

Race Conditions

Situations like this where processes access the same data concurrently and the outcome of execution depends on the particular order in which the access takes place is called a race condition.

#### Critical Section Problem

A Critical Section is a code segment that accesses shared variables and has to be executed as an atomic action.

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

1. Mutual Exclusion
2. Progress
3. Bounded Waiting

**Peterson's Solution:**

Two process solution:

Peterson’s Solution is a classical software based solution to the critical section problem.

**Bakery Algorithm:**

Goal – Solve the CS problem for n processes

Approach– Customers take numbers →lowest number gets service next (here service means entry to the CS).

#### Semaphores

Managing concurrent processes by using the value of a simple integer variable to synchronize the progress of interacting processes. This integer variable is called semaphore. So it is basically a synchronizing tool and is accessed only through two low standard atomic operations, wait and signal designated by P() and V() respectively.

**Monitor:**

A monitor type presents a set of programmer defined operations which can provide mutual exclusion within the monitor

**Classical problems of synchronization**

Producer-Consumer problem

Readers-Writers problem

Dining Philosophers problem

**Deadlocks:**

A set of blocked processes each holding a resource and waiting to acquire a resource held by another process in the set.

**Deadlocks Prevention:**

Do not allow one of the four conditions to occur

Mutual exclusion

Hold and wait

No preemption

Circular wait:

**Deadlock Avoidance:**

If we have prior knowledge of how resources will be requested, it's possible to

determine if we are entering an "unsafe" state.

### Deadlock Detection

If deadlocks are not avoided, then another approach is to detect when they have occurred and recover somehow.

**Recovery from Deadlock:**

Process Termination

 Abort all deadlocked processes.

Abort one process at a time until the deadlock cycle is eliminated.