# Operating system Class- BCA IV Semester

**Dr. Vijay Kant Sharma**

**Assistant professor ,**

Department Of Computer Application

Jagatpur P.G. College, Varanasi

Affiliated to Mahatma Gandhi Kashi vidhyapith Varanasi

Email-mzp.vijay@gmail.com

**OUTLINE-**

**UNIT :- III**

**Operating System**

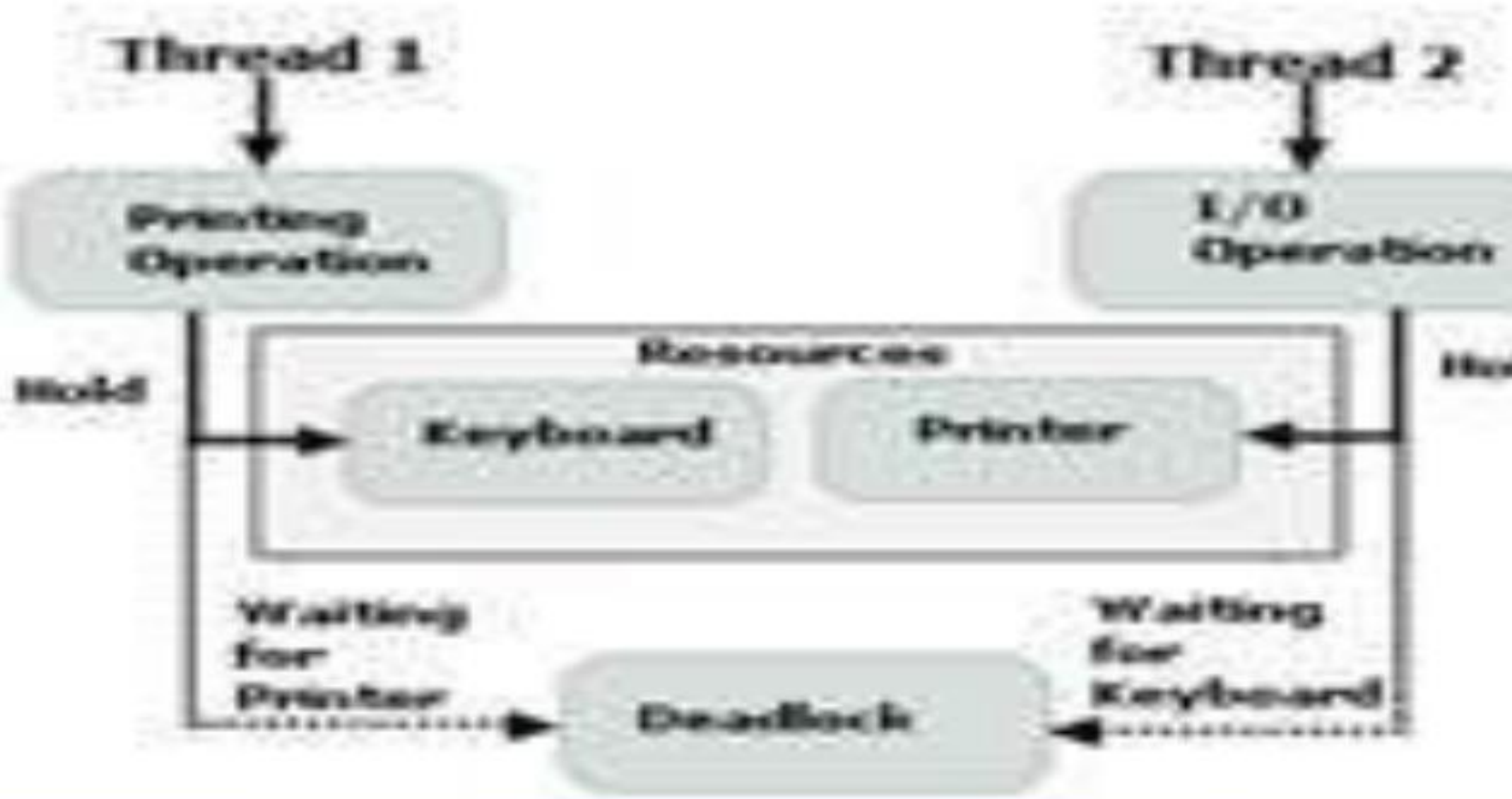
**Deadlock concept, Condition**

**Avoidance ,prevention**

**Detection and Recovery**

# DEADLOCK

A process request the resources, the resources are not available at that time, so the process enter into the waiting state. The requesting resources are held by another waiting process, both are in waiting state, this situation is said to be


Thread 1
Thread 2
Printing Operation
I/O Operation
Resources
Hold
Keyboard
Printer
Waiting for Printer
Waiting for Keyboard
Deadlock

# Conditions for Deadlock

A Deadlocked system must satisfied the following 4 conditions. These are :

1. MUTUAL EXCLUSION
2. HOLD AND WAIT
3. NO PREEMPTION
4. CIRCULAR WAIT

# Deadlock Avoidence

Avoid actions that may lead to a deadlock.

Think of it as a state machine moving from 1 state to another as each instruction is executed.

We can avoid the situattion of deadlocked by:

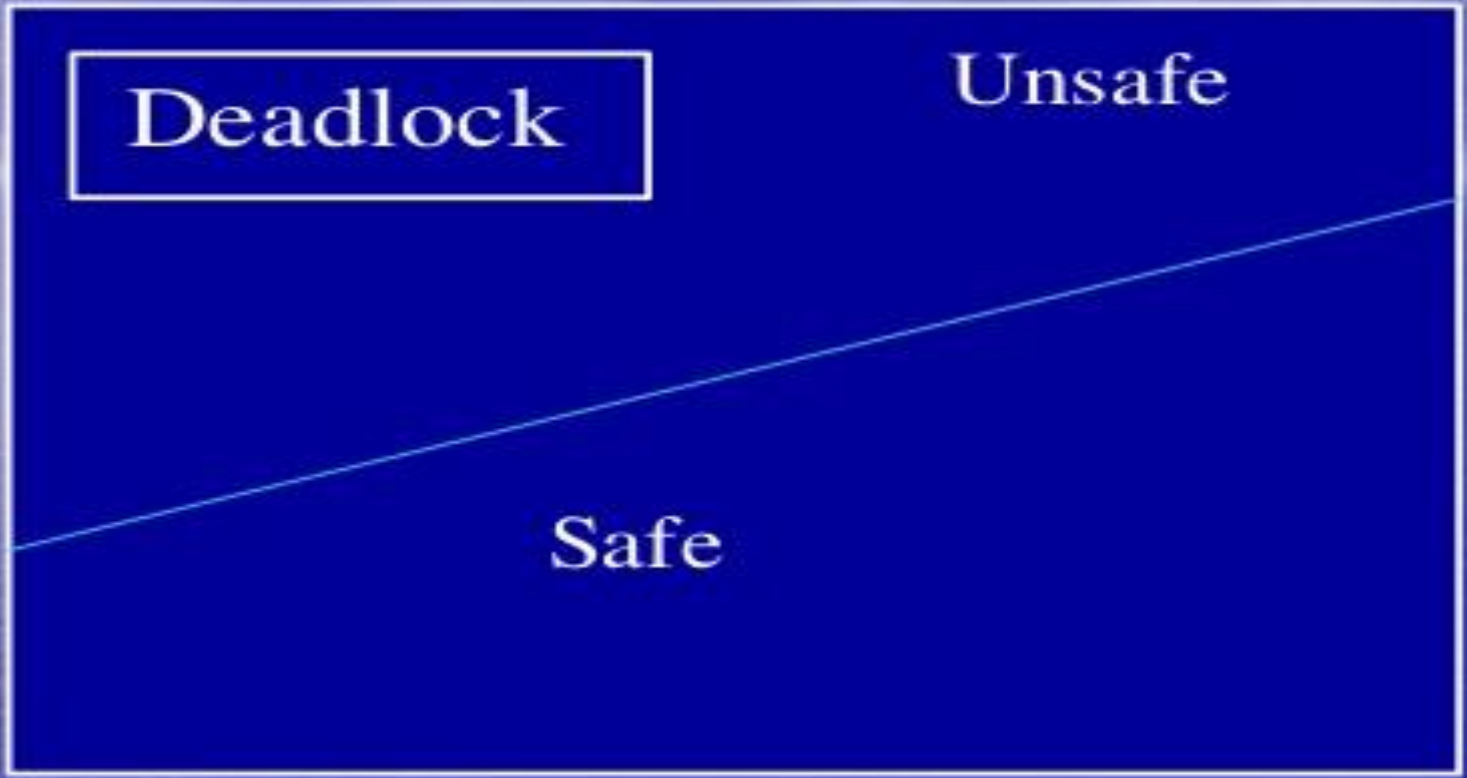
1. Safe State
2. Banker's Algorithm
3. Resource Allocation Graph

# Safe State

Safe state is one where :

It is not a deadlocked state There is some sequence by which all requests can be satisfied.

To avoid deadlocks, we try to make only those transitions that will take you from one safe state to another. We avoid transitions to unsafe state (a state that is not deadlocked, and is not safe)



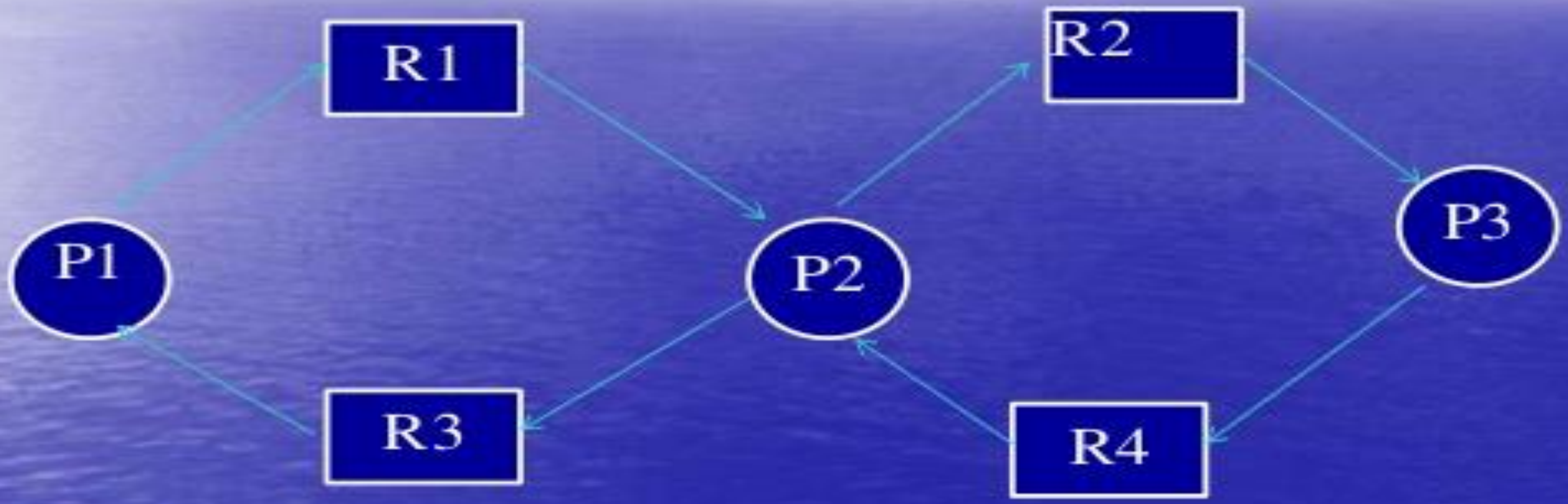

Safe unsafe and deadlock state spaces

# Banker's Algorithm

When a request is made, check to see if after the request is satisfied, there is a (atleast one!) sequence of moves that can satisfy all the requests ie. the new state is safe. If so, satisfy the request, else make the request wait.

# Resource Allocation Graph

If we have a resource allocation system with only one inatance of each process, a varient of the resource allocation graph can be used for deadlock avoidence.




Resource Allocation Graph

# Deadlock Prevention

*Difference from avoidance is that here, the system itself is build in such a way that there are no deadlocks.*

*Make sure atleast one of the 4 deadlock conditions is never satisfied.*

*This may however be even more conservative than deadlock avoidance strategy.*

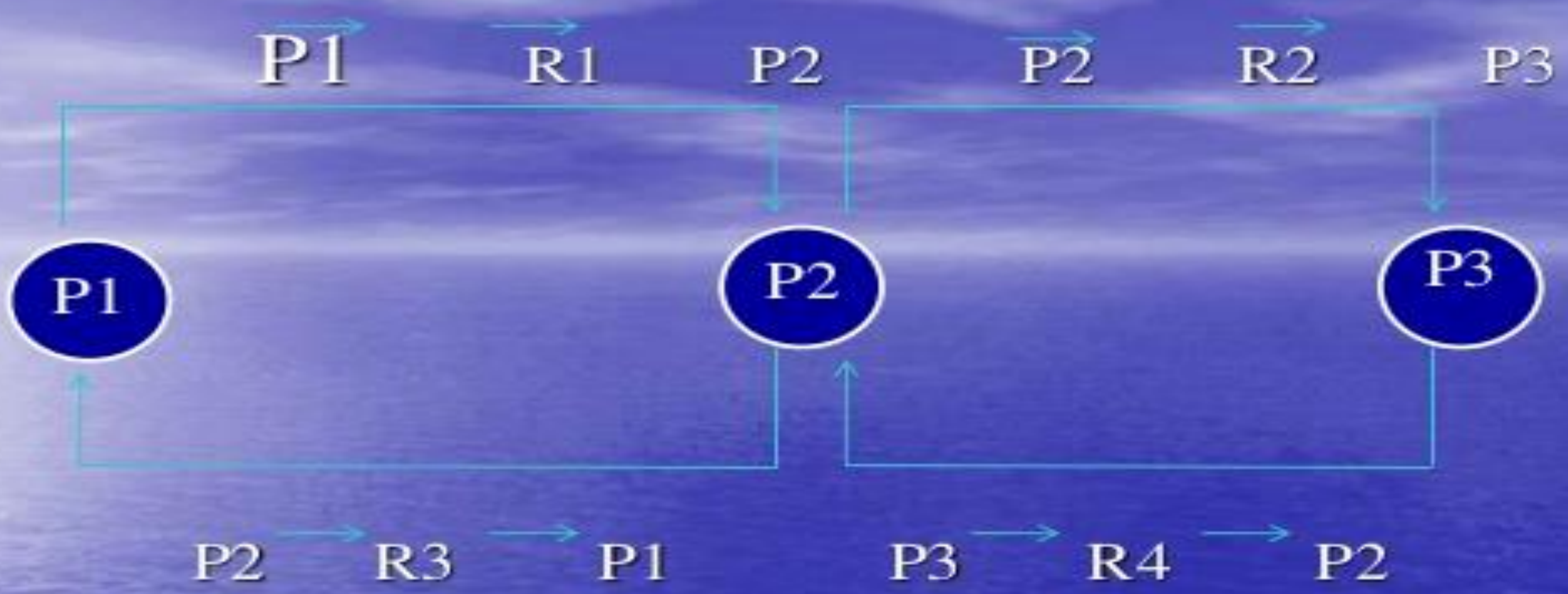

MUTUAL EXCLUSION: We can deny this condition by simple protocol i.e., "convert the all non-sharable resources to sharable resources".
HOLD AND WAIT: We can deny this condition with following two protocols :
"A process request the resources only when the process has none".
• "Each process to request and be allocated all its resources before it begins execution".
NO PREEMPTION :To ensure that this condition does not hold, we use the following protocol : we preempt the desired resources from the waiting process and allocate them to the requesting process.
CIRCULAR WAIT: We ensure that circular wait must not happened if we apply a simple solution, i.e., numbering all the resources types and each process request resources in an increasing order of enumeration

# Deadlock Detection

Detection mechanism of deadlocks for single instance of resource type is different. We can detect the dead locks using wait for graph for single instance resource type and detect using detection algorithm for multiple instances of resource type.

1. SINGLE INSTANCE OF RESOURCE TYPE :

Single instance of resource type means, the system consisting of only one resource for one type. We can detect this type of deadlocks with the help of wait for graph.




<u>Wait for graph</u>

A system is in deadlock state , if and only if the wait for graph contains cycles. So we can detect the deadlocks with cycles. In the figure there is 2 cycles one is P1 to P2 to P1, second one P2 to P3 to P2 so the system consisting of deadlocks.

# 2. SEVERAL INSTANCE OF RESOURCE TYPE

The wait for graph is not applicable to several instance of resource type. So we need another method for this type, that is “deadlock detection algorithm”. This algorithm looks like ‘Banker’s algorithm” and it employees several data structures that are similar to those used in the Banker’s algorithm.

# Deadlock Recovery

Once deadlock has been detected, some strategy is needed for recovery. The various approaches of recovering from deadlock are:

- PROCESS TERMINATION
- RESOURCE PREEMPTION

# PROCESS TERMINATION

"Process termination" it is one method to recover from deadlock. We uses 2 methods for process termination, these are:

a) ABORT ALL DEADLOCKED PROCESS : It means release all the processes in the deadlocked state, and start the allocation from the starting point. It is a great expensive method.

b) ABORT ONE BY ONE PROCESS UNTIL THE DEADLOCK CYCLE IS ELIMINATED : In this method first abort the one of the processes in the deadlocked state, and allocated the resources to some other process in the deadlock state then check whether the deadlock breaked

# RESOURCE PREEMPTION

To eliminate deadlocks using resource preemption, preempt some resources from processes and give these resources to other processes until the deadlock cycle is broken.

There are 3 methods to eliminate the deadlocks using resource preemption. These are :

a) SELECTING A VICTIM : Select a victim resource from the deadlock state, and preempt that one.

b) ROLLBACK : If a resource from a process is preempted, what should be done with that process. The process must be roll backed to some safe state and restart it from that state.

c) STARVATION : It must be guaranteed that

# Questions:-

- **What is deadlock explain it?**
- **What is the necessary and sufficient condition for deadlock?**
- **How the deadlock be prevented?**
- **How we avoid the deadlock occurrences?**

**Reference Books:**

1. Silberschatz and Galvin, "Operating System Concepts", Person, 5th Ed. 2001

2. Madnick E., Donovan J., "Operating Systems:, Tata McGraw Hill,2001

3. Tanenbaum, "Operating Systems", PHI, 4th Edition, 2000

4. Dietel, "Operating Systems", TMH.

**Declaration**

The content is exclusively meant for academic purpose and for enhancement teaching and learning.Any other use of economic/commercial purpose is strictly prohibited. The users of the content shall not distribute ,disseminate or share it with anyone else and its use is restricted to advancement of individual knowledge.The information provided in this e-content is authentic and best as per my knowledge.

**Dr.Vijay kant Sharma**

**Assistant professor**

**Department of computer Application**

**jagatpur P.G. College Varanasi Affiliated to Mahatma Gandhi Kashi Vidyapith Varanasi**

# THANK YOU