RAMA UNIVERSITY

www.ramauniversity.ac.in

FACULTY OF ENGINEERING & TECHNOLOGY

BCS-501 Operating System

Lecturer-14

Manisha Verma

Assistant Professor

Computer Science & Engineering

Deadlocks

- **System Model**
- **Deadlock Characterization**
- **Resource-Allocation Graph**

The Deadlock Problem

- 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.

P_1 and P_2 each hold one tape drive and each needs another one.

Example

semaphores *A* and *B*, initialized to 1

P_0	P_1
wait (A);	*wait(B)*
wait (B);	*wait(A)*

- To develop a description of deadlocks, which prevent sets of concurrent processes from completing their tasks

- To present a number of different methods for preventing or avoiding deadlocks in a computer system

System Model

System consists of resources
Resource types $R_1, R_2, \ldots, R_m$

CPU cycles, memory space, I/O devices
Each resource type R_i has W_i instances.

Each process utilizes a resource as follows:
- **request**
- **use**
- **release**

Deadlock Characterization

Deadlock can arise if four conditions hold simultaneously.

Mutual exclusion: only one process at a time can use a resource.
Hold and wait: a process holding at least one resource is waiting to acquire additional resources held by other processes.
No preemption: a resource can be released only voluntarily by the process holding it, after that process has completed its task.
Circular wait: there exists a set {P0, P1, …, P0} of waiting processes such that P0 is waiting for a resource that is held by P1, P1 is waiting for a resource that is held by
P2, …, Pn–1 is waiting for a resource that is held by
Pn, and P0 is waiting for a resource that is held by P0.

Resource-Allocation Graph

- A set of vertices V and a set of edges E.

- V is partitioned into two types:

  P = {P1, P2, …, Pn}, the set consisting of all the processes in the system.

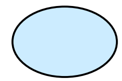
  R = {R1, R2, …, Rm}, the set consisting of all resource types in the system.

request edge – directed edge P1 $\rightarrow$ Rj

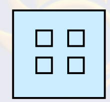
assignment edge – directed edge Rj $\rightarrow$ Pi

Resource-Allocation Graph (Cont.)

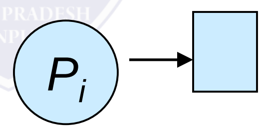
Process



Resource Type with 4 instances

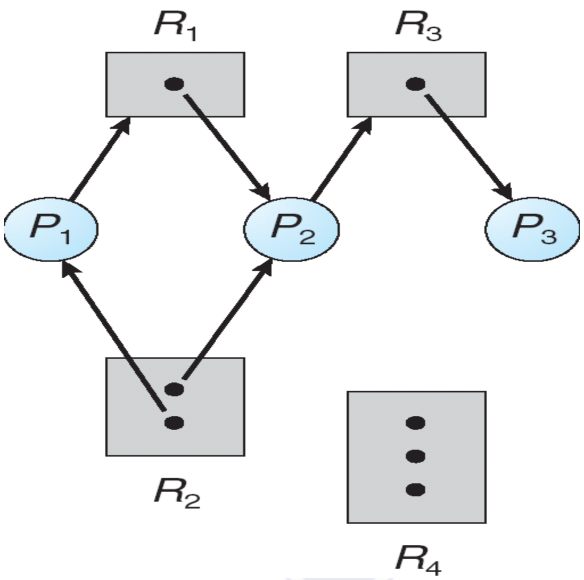


Pi requests instance of Rj



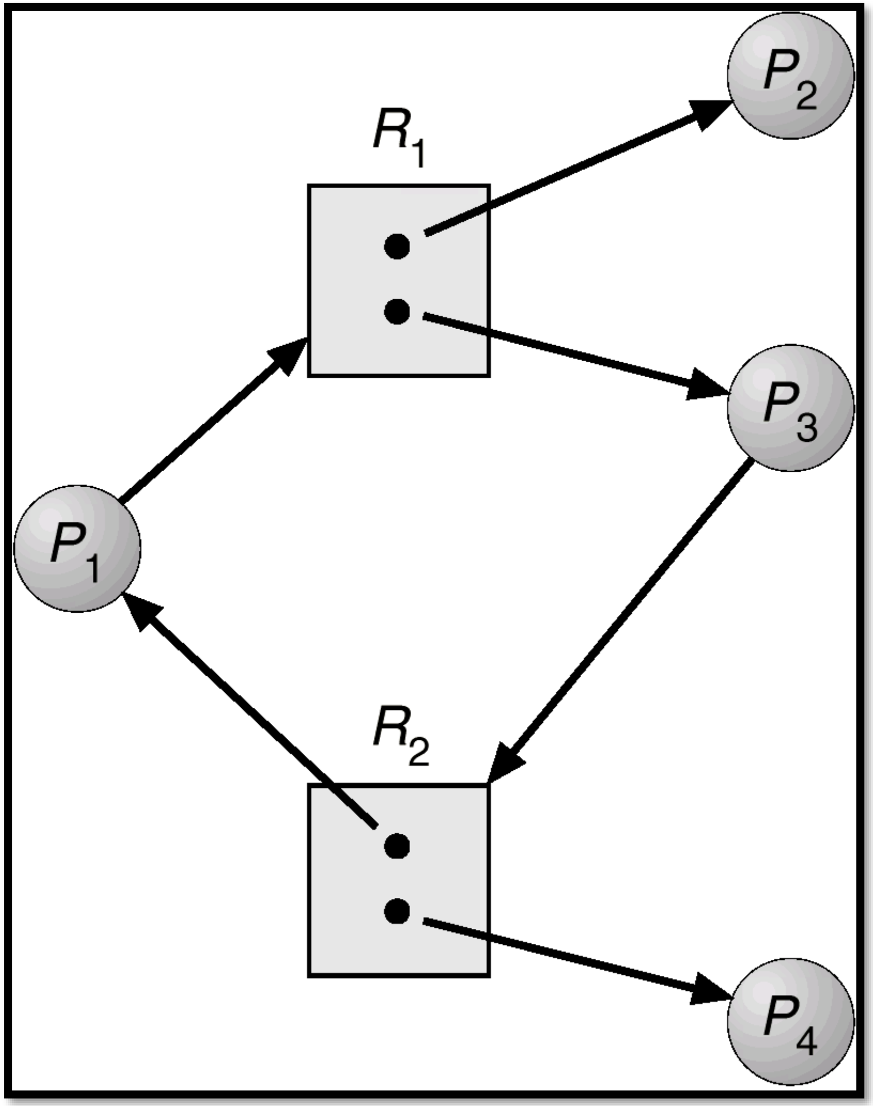
Pi is holding an instance of Rj

Example

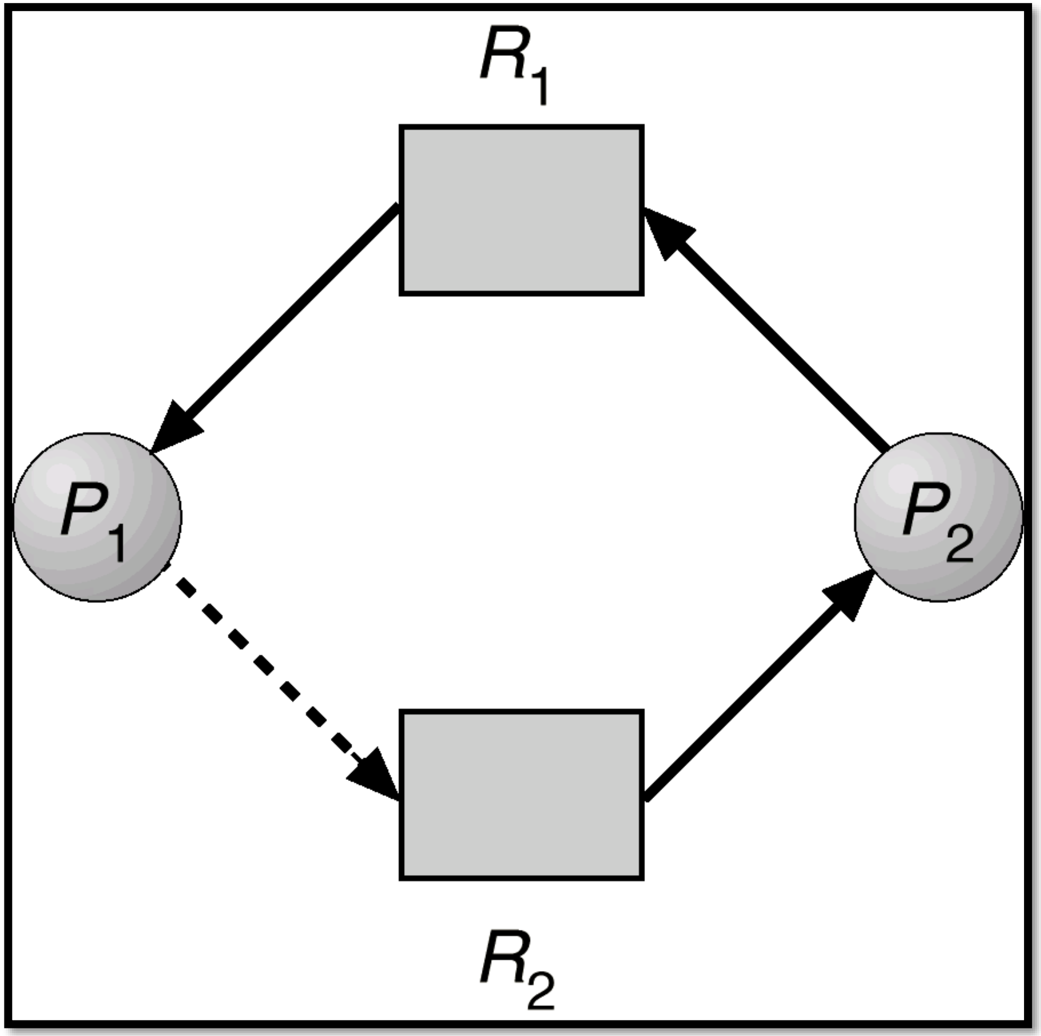


If graph contains no cycles ⇒ no deadlock
If graph contains a cycle ⇒
- if only one instance per resource type, then deadlock
- if several instances per resource type, possibility of deadlock

Resource Allocation Graph With A Cycle But No Deadlock



Unsafe State In Resource-Allocation Graph



Methods for Handling Deadlocks

•Ensure that the system will never enter a deadlock state:
- ➢Deadlock prevention
- ➢Deadlock avoidence

•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

MCQ

For effective operating system, when to check for deadlock?

A. every time a resource request is made
B. at fixed time intervals
C. both (a) and (b)
D. none of the mentioned

A problem encountered in multitasking when a process is perpetually denied necessary resources is called:

A. deadlock
B. starvation
C. inversion
D. aging

Which one of the following is a visual (mathematical) way to determine the deadlock occurrence?

A. resource allocation graph
B. starvation graph
C. inversion graph
D. none of the mentioned

To avoid deadlock:

A. there must be a fixed number of resources to allocate
B. resource allocation must be done only once
C. all deadlocked processes must be aborted
D. inversion technique can be used

The number of resources requested by a process :

A. must always be less than the total number of resources available in the system
B. must always be equal to the total number of resources available in the system
C. must not exceed the total number of resources available in the system
D. must exceed the total number of resources available in the system