CS & IT ENGINEERING

Operating System

Process Synchronization

Lecture -5

By- Vishvadeep Gothi sir

Recap of Previous Lecture

Topic Semaphore

Topic Producer-Consumer Problem

Topic Reader-Writer Problem

Topics to be Covered

Topic Dining Philosopher Problem

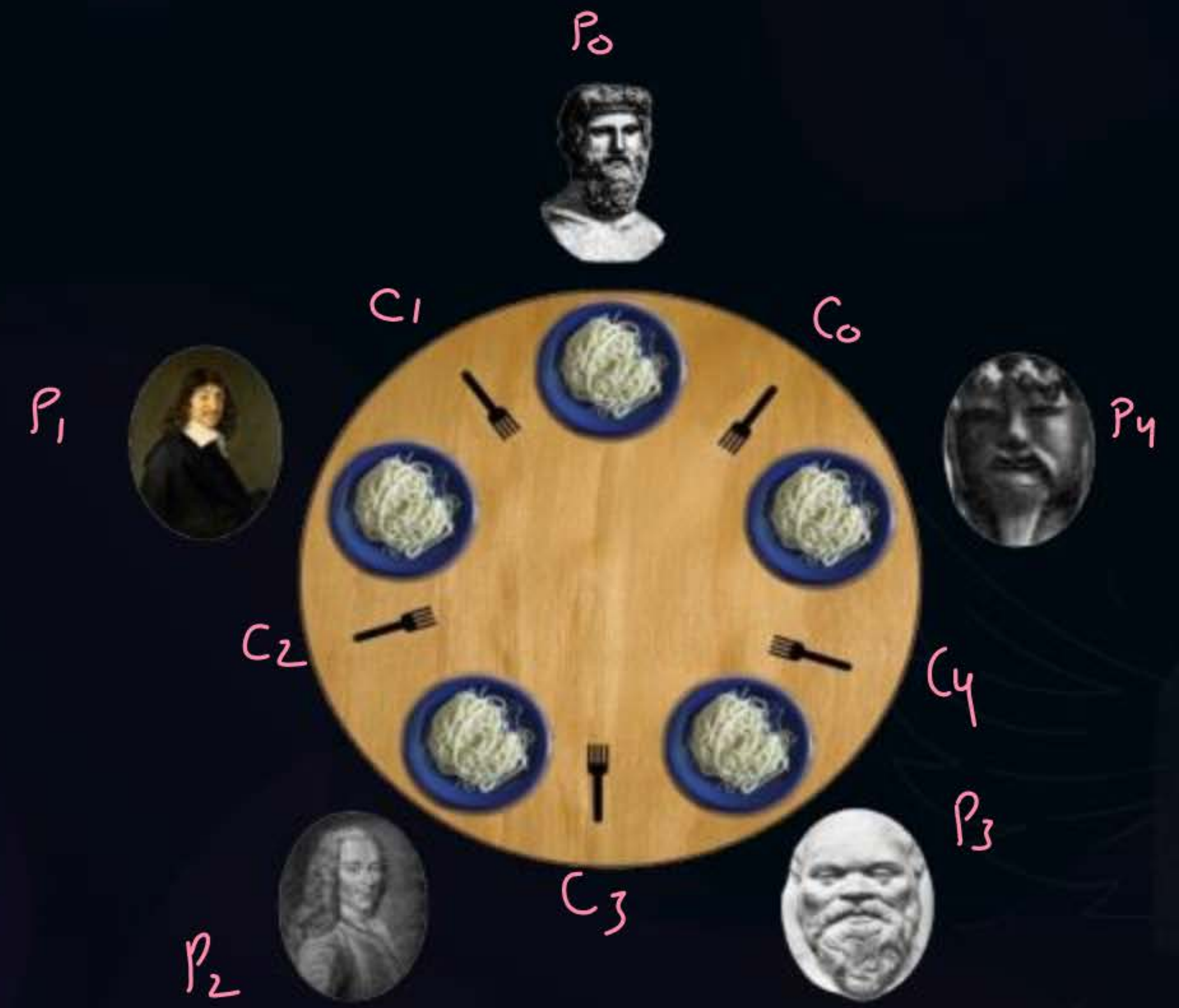
Topic Questions on Synchronization

Topic Deadlock

Topic : Dining Philosopher Problem

- K philosophers seated around a circular table
- There is one chopstick between each philosopher
- A philosopher may eat if he can pick up the two chopsticks adjacent to him
- One chopstick may be picked up by any one of its adjacent followers but not both

Topic : Dining Philosopher Problem Solution



Topic : Dining Philosopher Problem Solution

Binary semaphore array of size k ⇒ $ch[k] = \{1, 1, 1, 1, 1\}$

Process P_i

```
wait (ch [i])
wait (ch [(i+1) mod k])

// eating

signal (ch [i])
signal (ch [(i+1) mod k])
```

This solution suffers from deadlock

Topic : Dining Philosopher Problem Solution

Some of the ways to avoid deadlock are as follows –

1. There should be at most (k–1) philosophers on the table

Topic : Dining Philosopher Problem Solution

Some of the ways to avoid deadlock are as follows –

1. There should be at most (k–1) philosophers on the table
2. A philosopher should only be allowed to pick their chopstick if both are available at the same time

Topic : Dining Philosopher Problem Solution

Some of the ways to avoid deadlock are as follows –

1. There should be at most (k–1) philosophers on the table
2. A philosopher should only be allowed to pick their chopstick if both are available at the same time
3. One philosopher should pick the left chopstick first and then right chopstick next; while all others will pick the right one first then left one

Topic : Characteristics of Semaphores

- Used to provide mutual exclusion
- Used to control access to resources
- Solution using semaphore can lead to have deadlock
- Solution using semaphore can lead to have starvation
- Solution using semaphore can be busy waiting solutions
- Semaphores may lead to a priority inversion → a low priority process can get c.s. before a high priority process.
- Semaphores are machine-independent

Topic : Operations on Resources

3 Operations on resources:

→ H/w or S/w

1. Request → granted or reject or wait
2. Use
3. Release

Topic : Deadlock

If two or more processes are waiting for such an event which is never going to occur

	Holds	waits
P0	keyboard	HDD
P1	HDD	Printer
P2	Printer	keyboard

Topic : Deadlock

Topic : Necessary Conditions for Deadlock

Deadlock can occur only when all following conditions are satisfied:

1. Mutual Exclusion ⇒ for resource access
2. Hold & Wait ⇒ each deadlocked process must hold atleast a resource and must wait for atleast a resource.
3. No-preemption ⇒ no preemption of resources
4. Circular Wait ⇒

Topic : Resource Allocation Graph

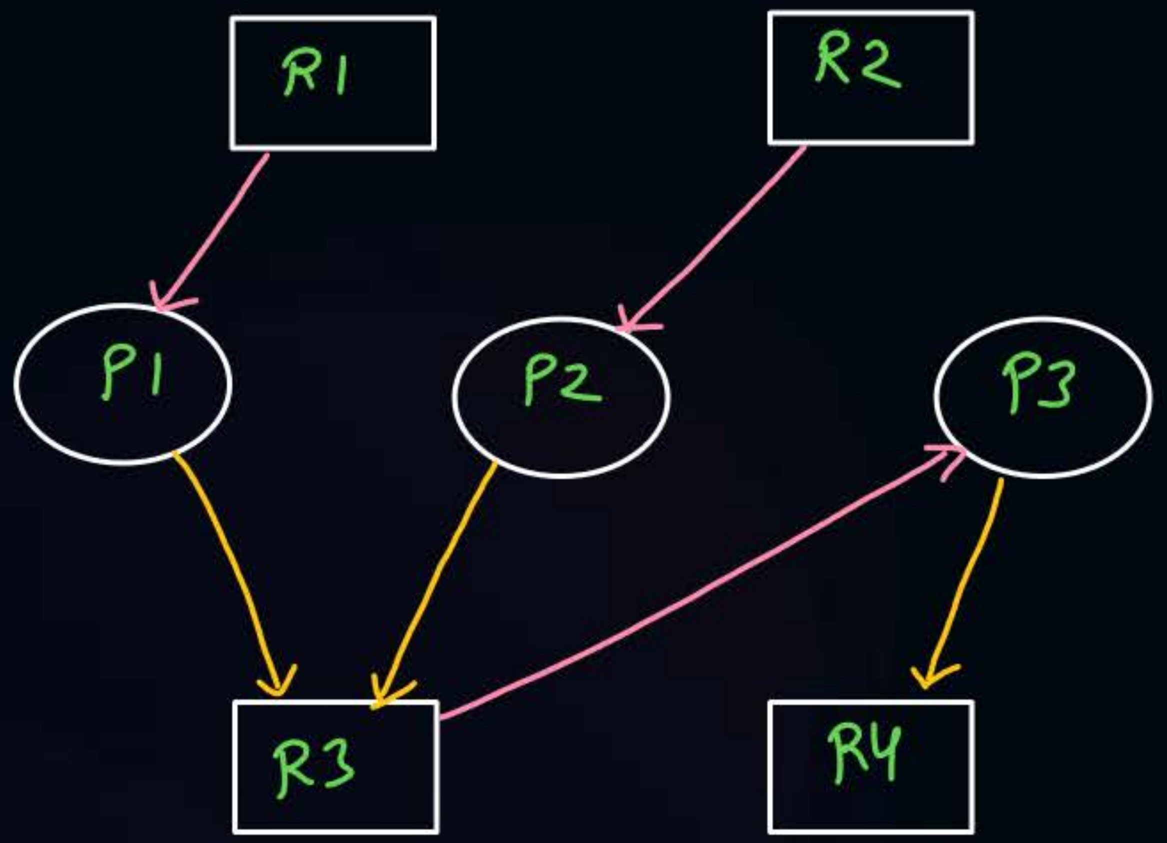
(directed graph)

It shows which resource is allocated to which process and which process has requested which resource.

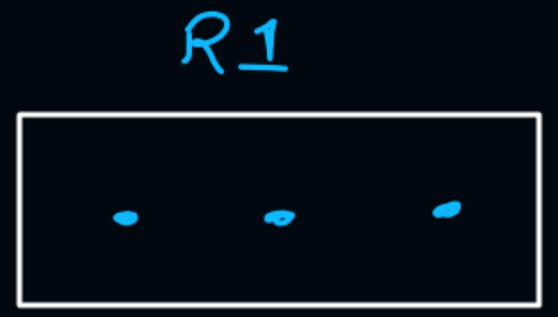
Vertices
- → Process ⬭
- → Resource □

Edges
- → Allocation Edge ⇒ from resource instance to process
- → Request Edge ⇒ Process to resource

Topic : Resource Allocation Graph

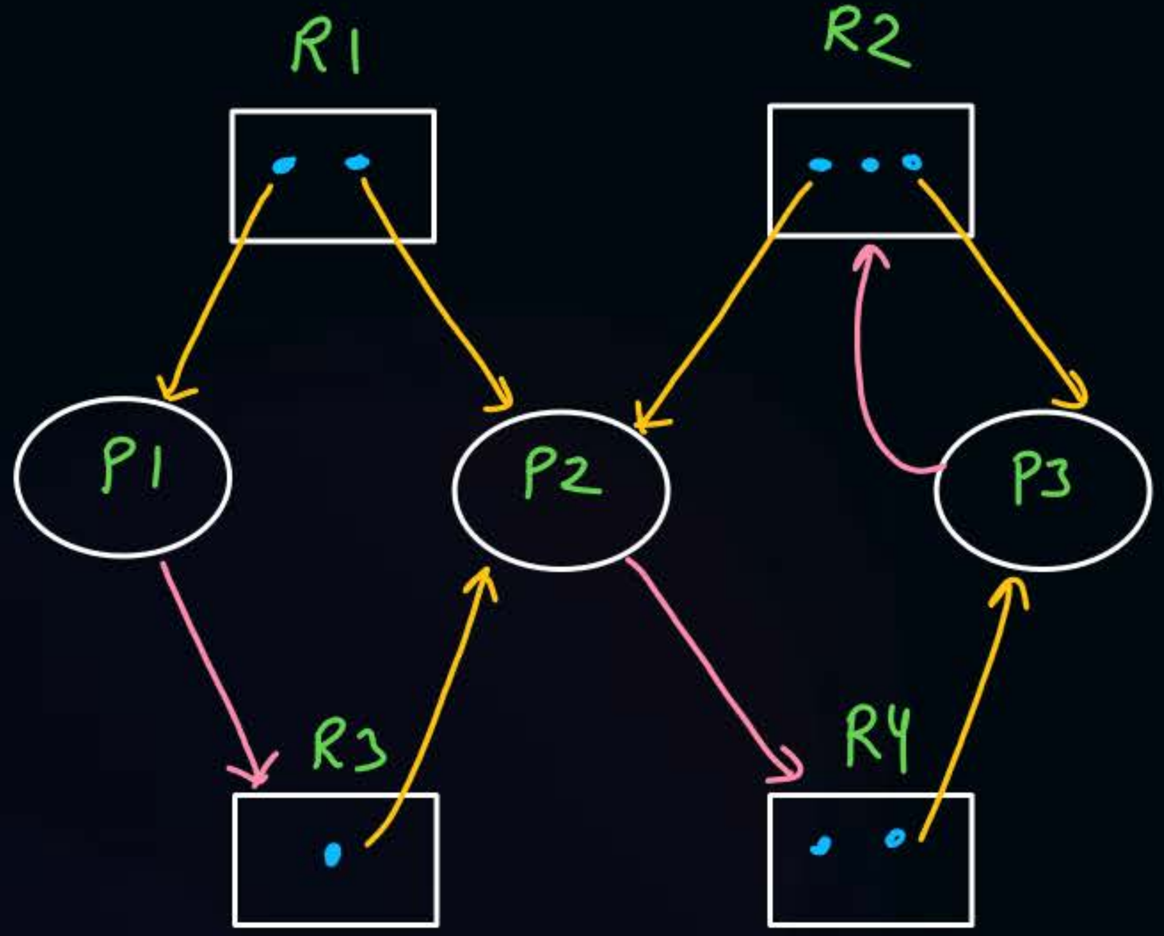


If resources have multiple instances



It shows there are 3 instances of resource R1.

Topic : Resource Allocation Graph



Topic : Recovery From Deadlock

1. Make Sure that deadlock never occur
    - Prevent the system from deadlock or avoid deadlock
2. Allow deadlock, detect and recover
3. Pretend that there is no any deadlock

2 mins Summary

Topic Dining Philosopher Problem

Topic Questions on Synchronization

Topic Deadlock

Happy Learning

THANK - YOU