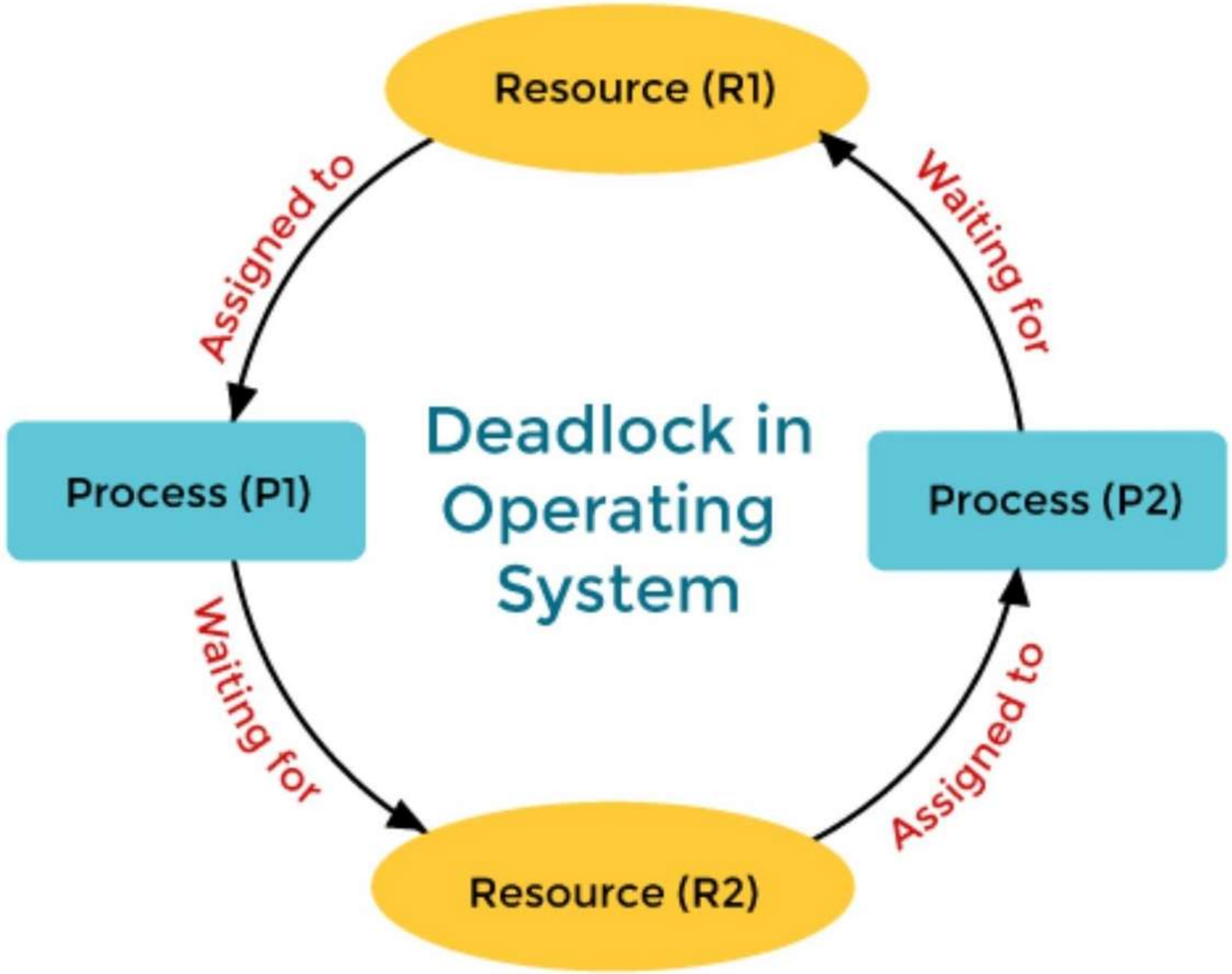


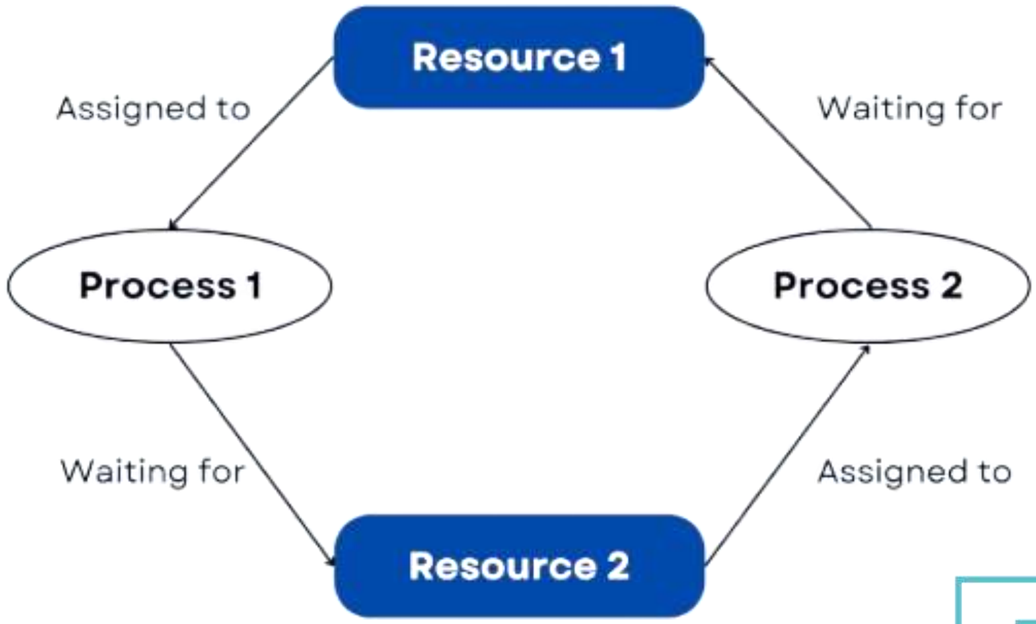
Operating Systems

SE 232

Sadiya Khanom Kanta
ID: 232-35-771
Siddhartho Sen
ID: 232-35-803

Monisha Debnath
ID: 232-35-241
Siddhartho Sen
ID: 232-35-803

What is Deadlock?

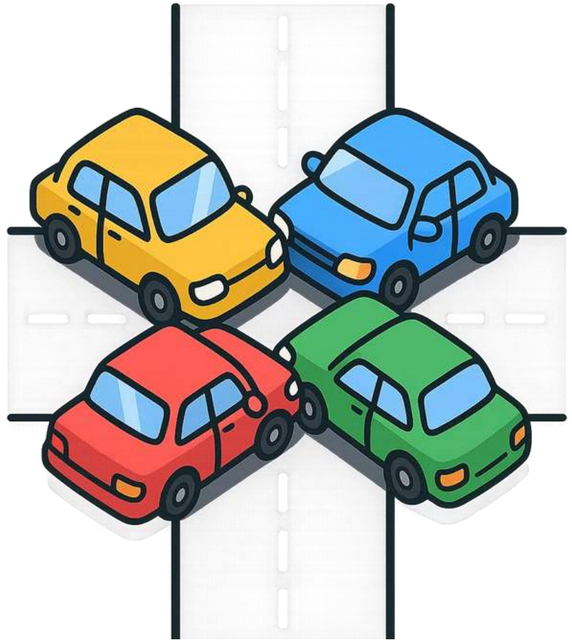


- Deadlock occurs when processes wait for each other’s resources.

- No process can proceed because each is waiting.

Real-Life Analogies of Deadlock

Traffic Gridlock: Each car blocks the next — no one can move.



Notebook & Pen: Two people holding what the other needs — both are stuck.

Conditions for Deadlock (Coffman's Conditions)

Conditions are:

1. **Mutual Exclusion**
2. **No Preemption**
3. **Hold and Wait**
4. **Circular Wait**

Mutual Exclusion

A resource can only be used by one process at a time, and no other process can access it until the first process releases it

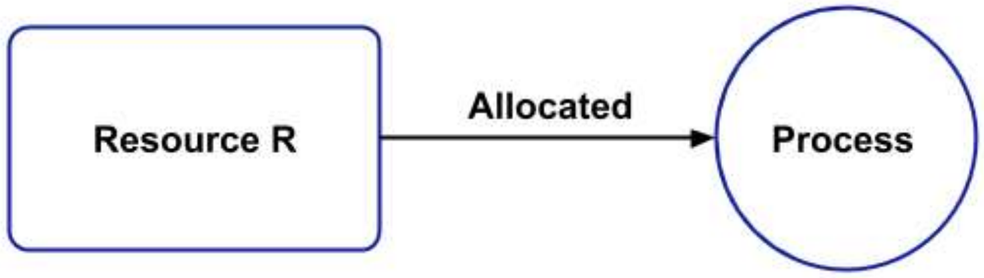
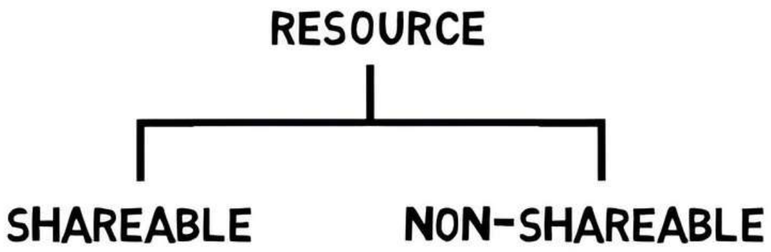
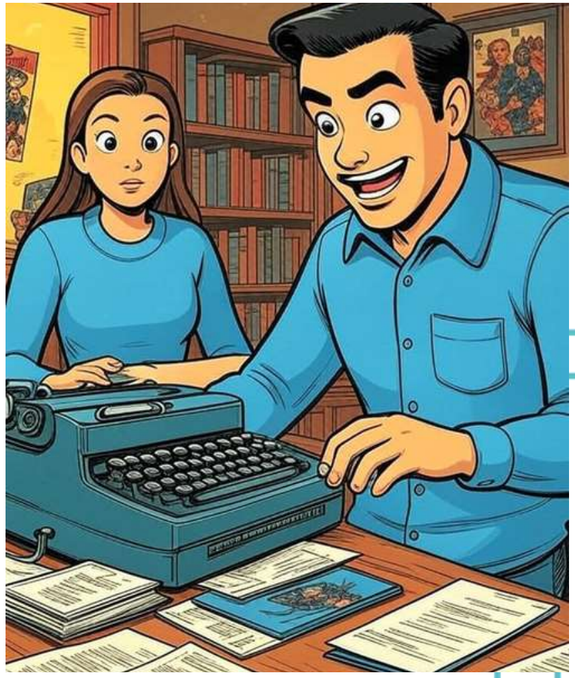


fig:Mutual Exclusion in Deadlock





Example: Printer

Hold and Wait

A process is holding at least one resource at a time and is waiting to acquire other resources held by some other process.

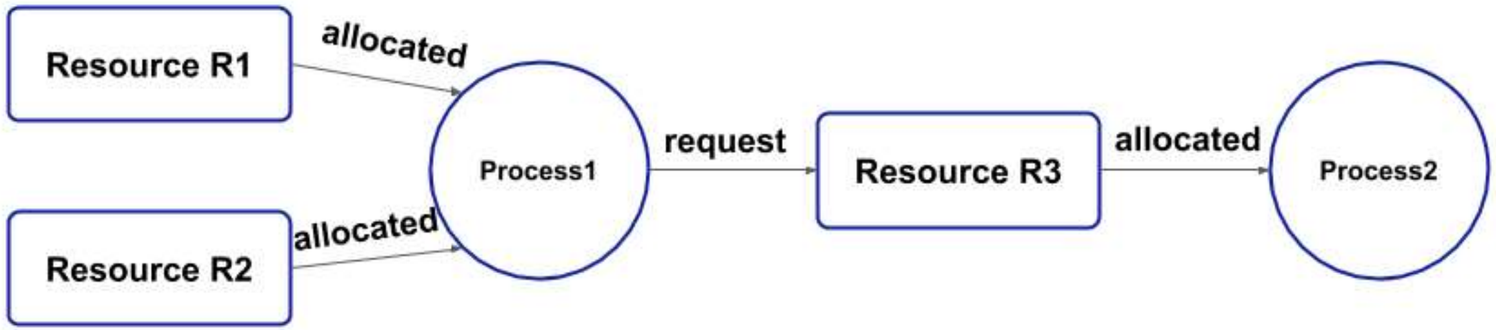


fig:Hold and Wait in Deadlock

Exampl e

No Preemption

Once a process has been granted a resource, it cannot be forcibly taken away from that process

preemption → force stopping a process

Example

Circular Wait

Circular wait happens when each process in a group is waiting for a resource that is held by the next process in the group, forming a circle.

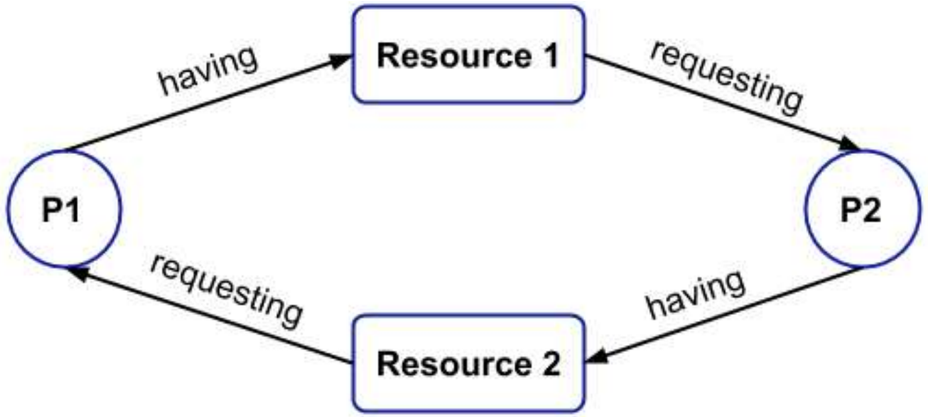


Fig:Circular Wait in Deadlock

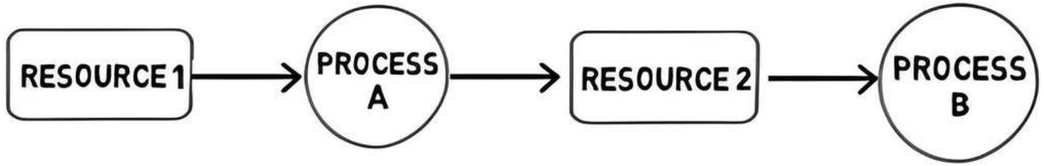


Fig:Not Following Circular Wait in Deadlock

Ways To Handle Deadlock

Prevention
blocks one of the four deadlock conditions

Avoidance
checks safety before granting resources

Detection and Recovery
detects and resolves deadlocks after they occur

Ignore
tolerates deadlocks if rare

Prevention

Avoidance

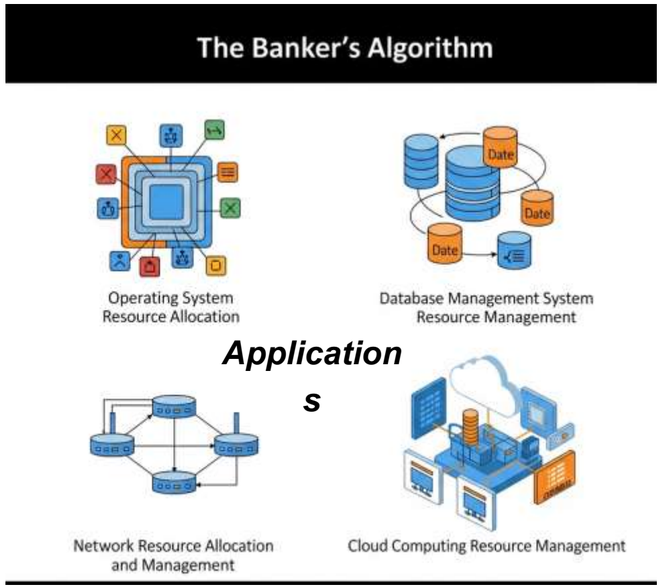
Detection and Recovery

Ignore

Banker’s Algorithm

Key Structures

- ***Allocation***
- ***Max***
- ***Available***
- ***Need***



Real-World Examples & Consequences

- ➤ *Deadlock occurs when a set of processes are blocked, each waiting for a resource held by another.*

- ➤ *It results in a standstill where no process can continue execution.*

- ➤ *Typically involves shared resources like memory, CPU cycles, or devices.*

- ➤ *Occurs in concurrent systems where processes compete for limited resources.*

Summary & Final Thoughts

- ***Deadlock occurs when a set of processes are blocked, each waiting for a resource held by another.***

- ***Understanding necessary conditions helps in effective prevention.***

- ***Banker's algorithm is a key strategy for avoidance.***

- ***Real-world consequences underline the importance of robust deadlock handling.***

THANK YOU