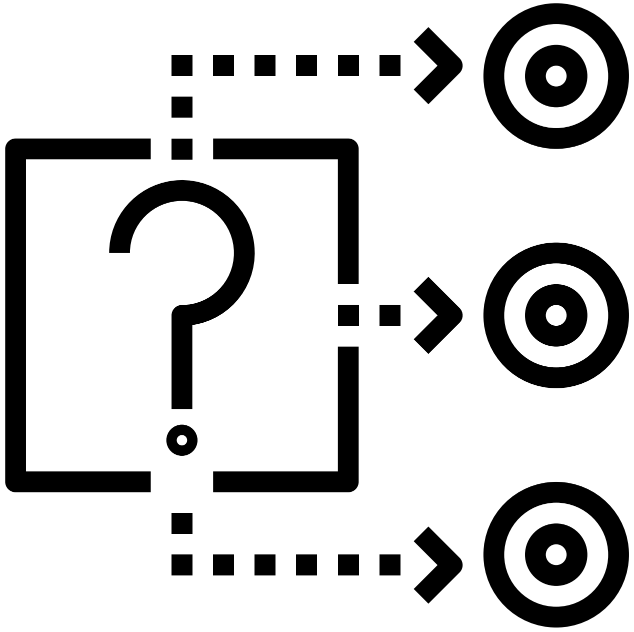
DEADLOCK AND STARVATION

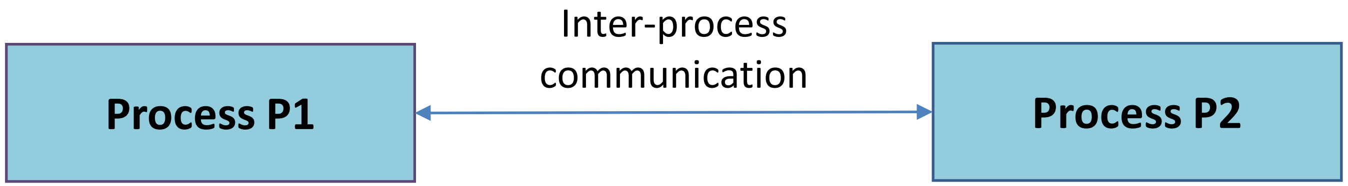
DCS4103 Operating System



Inter-Process Communication

Introduction

Allows processes to communicate with each other and letting another process know that some event has occurred or the transferring of data from one process to another



Advantages

- ❑ Helps to speedup modularity
- ❑ Computational
- ❑ Privilege separation
- ❑ Convenience
- ❑ Communication

Synchronization

信号量
Semaphore

- Variable that controls the access to a common resource by multiple processes
  - Binary
  - Counting

互斥
Mutual Exclusion

- Only one process thread can enter the critical section at a time

障碍
Barrier

- Does not allow individual process to process until all the process reach it

Spinlock

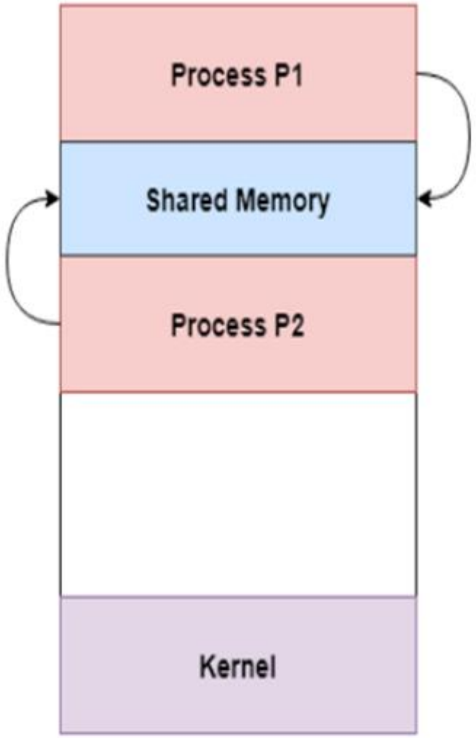
- Acquire the spinlock waits or stays in a loop while checking that the lock is available or not
- Busy waiting
  - The process is active, but does not perform any functional operation

执行
Implementation

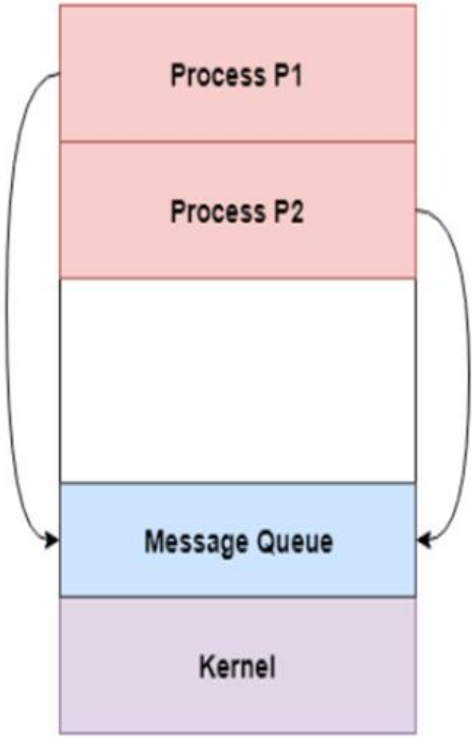
Pipe	Message Passing	Direct Communication	Indirect Communication
• Unidirectional • Moved in only a single direction at a time • POSIX systems • Windows OS	• Allows processes to synchronize and communicate with each other without restoring the shared variables	• Create or establish link between two communication processes • Only one link can exist	• Only exist or be established when processes share a common mailbox • Share multiple communication links

Implementation

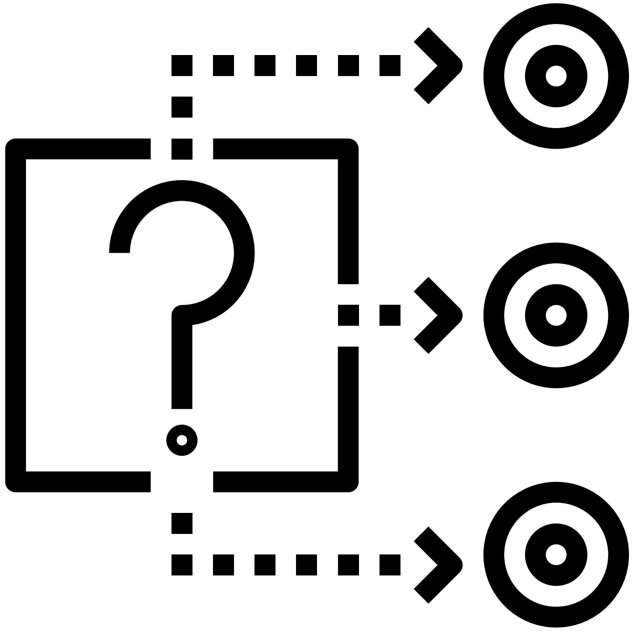
Shared Memory	Message Queue
• Which can be used or accessed by multiple processes simultaneously • Processes can communicate with each other	• Allows different messages to read and write the data • The message are stored or stay in the queues unless their recipients retrieve them



Shared Memory



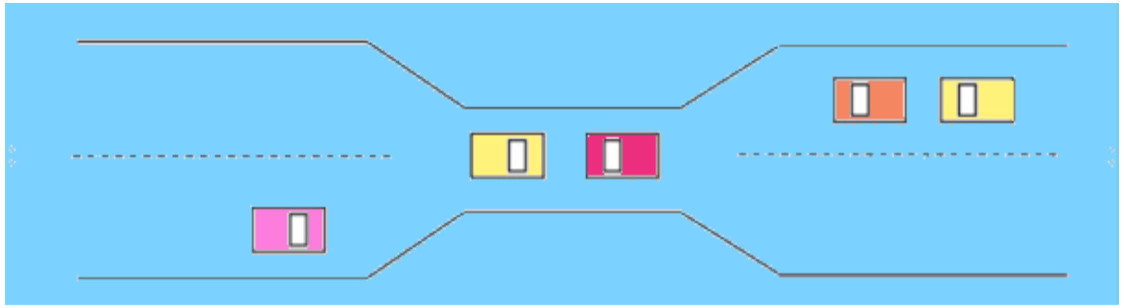
Message Queue

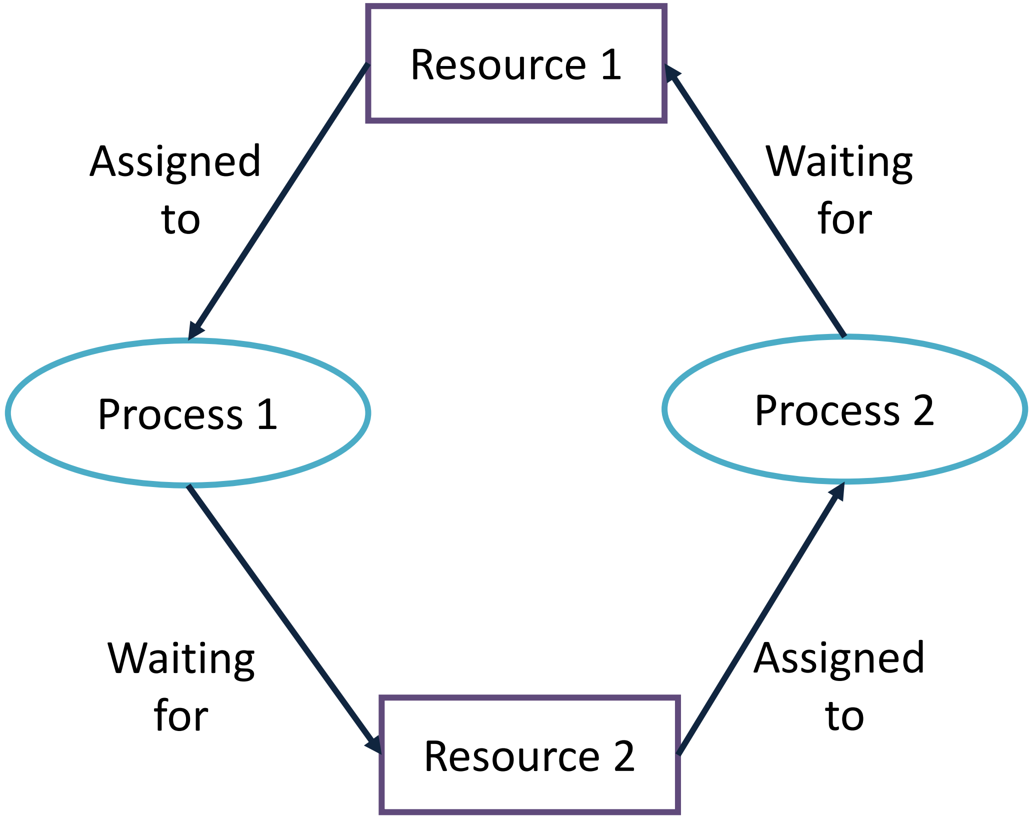


Deadlock

Introduction

- A situation that occurs in OS when any process enters a waiting state
- Another waiting process holding the demanded resource
- Examples
  - Traffic
  - Bridges





Principles / Causes

Mutual Exclusion

- Two process cannot use the same resource at the same time

Hold and Wait

- A process waits for some resources while holding another resource at the same time.

No preemption

- The process which once scheduled will be executed till the completion.
- No other process can be scheduled by the scheduler meanwhile.

Circular Wait

- All the processes must be waiting for the resources in a cyclic manner so that the last process is waiting for the resource which is being held by the first process.

Strategies

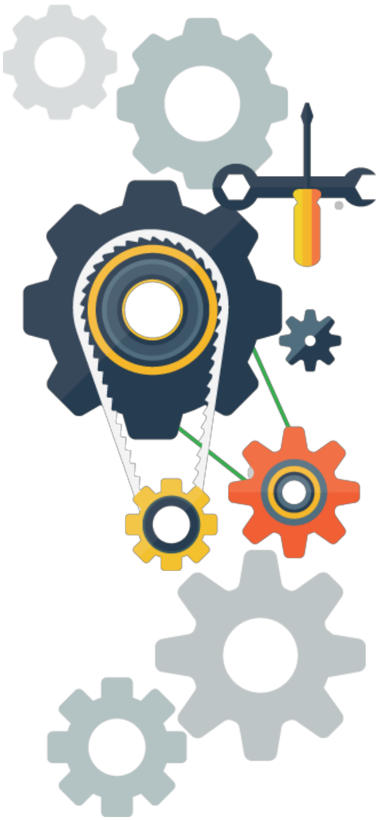
Strategies

Deadlock Ignorance

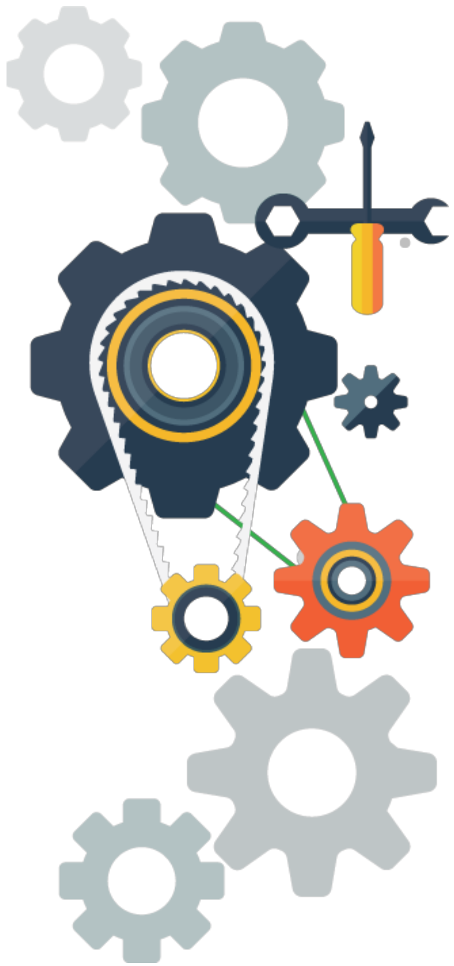
- ❑ The most widely used
- ❑ Assumes that deadlock never occurs
- ❑ For a single end user system
- ❑ Simply restart the computer
  - Windows
  - Linux

Deadlock Prevention

- ❑ To prevent a deadlock before it can occur
- ❑ Happens only when Mutual Exclusion, hold and wait, No preemption and circular wait holds simultaneously.
- ❑ The system checks every transaction before executing it



Strategies



Deadlock Avoidance

- The simplest and most useful
- Declare the maximum number of resources a process can request to complete its execution.

Avoidance Algorithm

- Resource-allocation graph
  - Pictorial representation of the state of a system

Banker's Algorithm

- If the customer's request does not cause the bank to leave a safe state, the cash will be allocated, otherwise the customer must wait until some other customer deposits enough
- To test for safely simulating the allocation for determining the maximum amount available for all resources
- Checks for all the possible activities before determining whether allocation should be continued or not.

Banker's Algorithm

Characteristics

- Keep many resources that satisfy the requirement of at least one client
- Whenever a process gets all its resources, it needs to return them in a restricted period.
- When a process requests a resource, it needs to wait
- The system has a limited number of resources
- Advance feature for max resource allocation

Disadvantages

- Does not allow the process to change its Maximum need while processing
- It allows all requests to be granted in restricted time, but one year is a fixed period for that.
- All processes must know and state their maximum resource needs in advance.

Starvation

Introduction

- ❑ Happens when a low priority program requests a system resource but cannot run
- ❑ Higher priority program has been employing that resource for a long time
- ❑ Prevent the lower priority process from obtaining the requested resource.
  - ▪ Priority Scheduling Algorithm

Causes

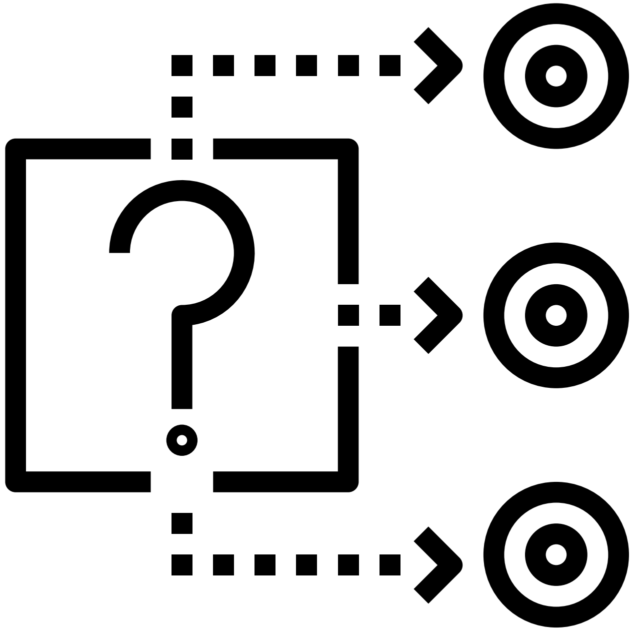
- ❑ Lack of resources
- ❑ Faulty resource allocation decisions
- ❑ Higher priority operations monopolize the processor

Solutions

Aging	Manager	Avoidance
• The priority of a process increases the longer it waits.	• Independently distributes resources properly	• Random process selection

Deadlock vs Starvation

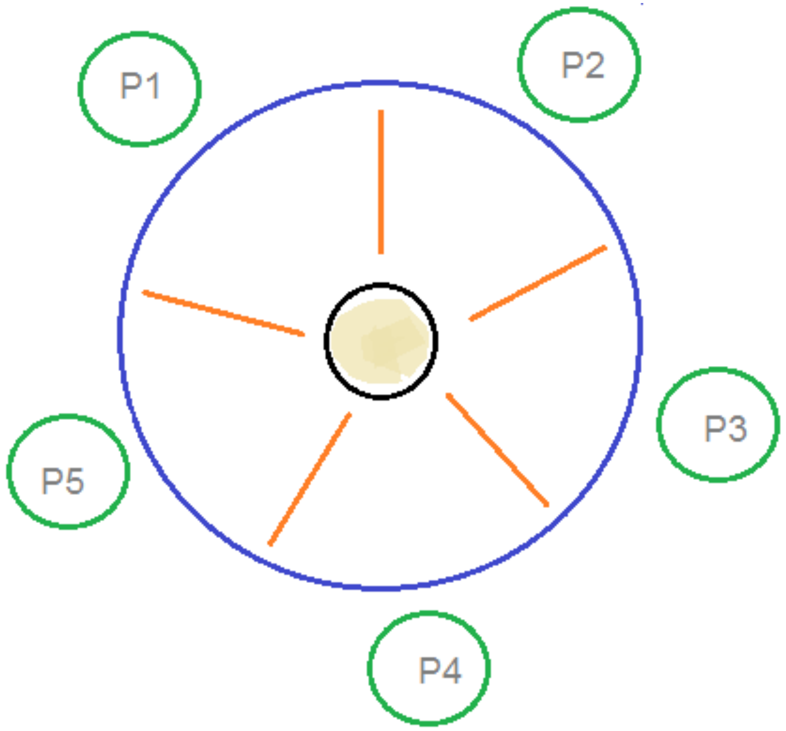
Features	Deadlock	Starvation
Definition	When every process holds a resource and waits for another process to hold another resource.	When a low priority program requests a system resource but cannot run because a higher priority program has been employing that resource for a long time.
Basic	Occurs when no process can proceed and becomes blocked.	Occurs when low priority procedures are blocked while high priority operations proceed.
Other names	Circular wait.	Lived Lock
Resources	Other processes block requested resources while a process is deadlocked.	High-priority processes continue to use the requested resources.
Arising Condition	Mutual exclusion's occurrence, Hold and wait, No preemption, and Circular wait all happen simultaneously.	Uncontrolled resource management, enforcement of priorities.
Prevention	Avoidance	Aging



Dining Philosophers Problem

Introduction

- Evaluate situations where there is a need of allocating multiple resources to multiple processes.
- When a philosopher wants to eat, he uses two chopsticks
  - one from their left
  - one from their right
- When a philosopher wants to think, he keeps down both chopsticks at their original place.



Semaphore

- Wait for the chopstick at the left and picks up that chopstick.
- Then wait for the right chopstick to be available, and then picks it too.
- After eating, put both the chopsticks down.
- Deadlock
  - Two neighboring philosophers want to eat at the same time

Solution

- One-to-one relationship of user-level thread to the At most four philosophers on the table
- An even philosopher should pick the right chopstick and then the left chopstick while an odd philosopher should pick the left chopstick and then the right chopstick.
- Only be allowed to pick chopstick if both are available at the same time.

Revision Questions

1. List approaches to implement inter-process communication.
2. Describe the cause deadlock to arise
3. Explain starvation and the causes.

Revision Questions

Consider the following set of processes:

Process	Allocation			Max			Work		
	R1	R2	R3	R1	R2	R3	R1	R2	R3
P_0	1	1	2	4	3	3	2	1	0
P_1	2	1	2	3	2	2			
P_2	4	0	1	9	0	2			
P_3	0	2	0	7	5	3			
P_4	1	1	2	1	1	2			

Compute whether the system is safe or not using the Banker's algorithm. Determine the sequence if it is safe.

Revision Questions

Consider the following set of processes:

Process	Allocation				Max				Work			
	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4
P_0	1	2	1	0	3	3	2	3	6	4	6	4
P_1	2	1	2	0	3	3	2	2				
P_2	1	0	1	2	2	1	3	3				
P_3	1	0	0	1	3	2	2	2				

Compute whether the system is safe or not using the Banker’s algorithm. Determine the sequence if it is safe.