SNS COLLEGE OF TECHNOLOGY

An Autonomous Institution

Coimbatore-35

Department of Computer Science and Engineering

23CST206–OPERATING SYSTEMS AND VIRTUALIZATION

B.E- CSE /IV SEMESTER

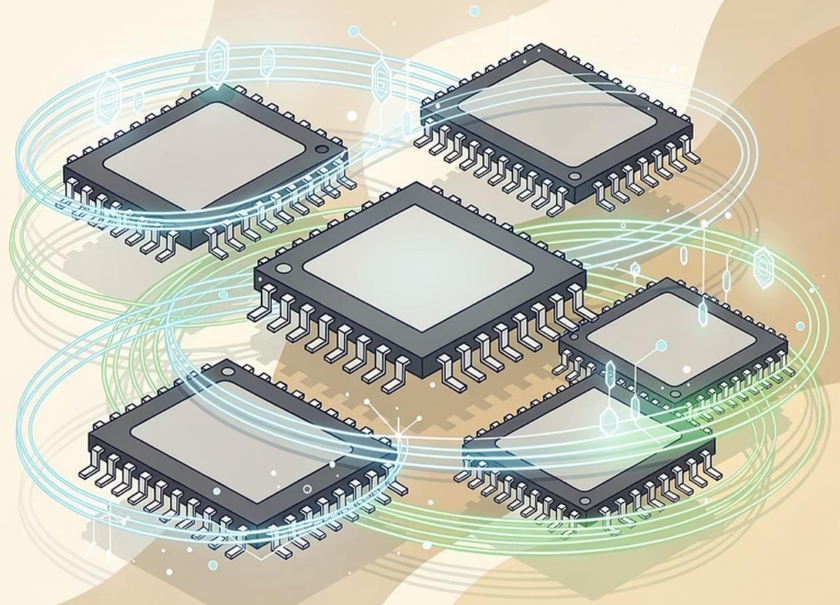
UNIT – III MEMORY MANAGEMENT

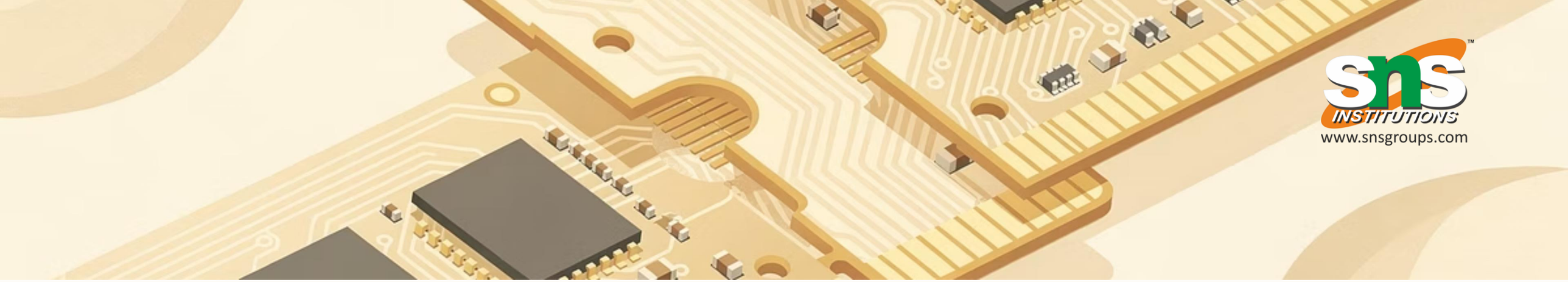
Topic 1: Main Memory - Swapping - Contiguous Memory Allocation

Main Memory, Swapping & Contiguous Allocation

OPERATING SYSTEMS

MEMORY MANAGEMENT





What is Main Memory?

Primary Memory

Internal memory CPU directly accesses

Active Storage

Stores OS, executing programs, active data

Temporary Workspace

CPU processes data only from RAM

Key Characteristics

RAM (Random Access Memory)

Direct access to any memory location

High Speed

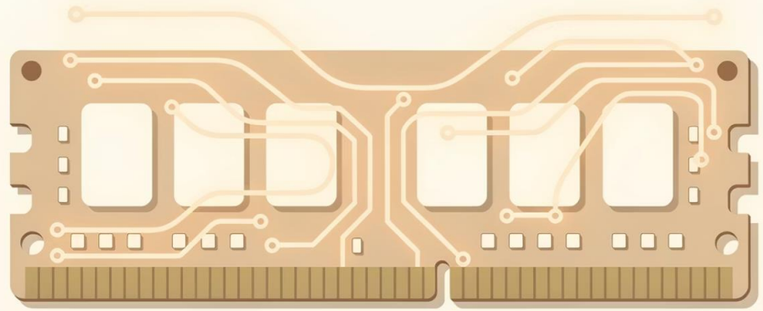
Faster than secondary storage

Volatile Nature

Data lost when power is off

Limited Size

Smaller than secondary memory



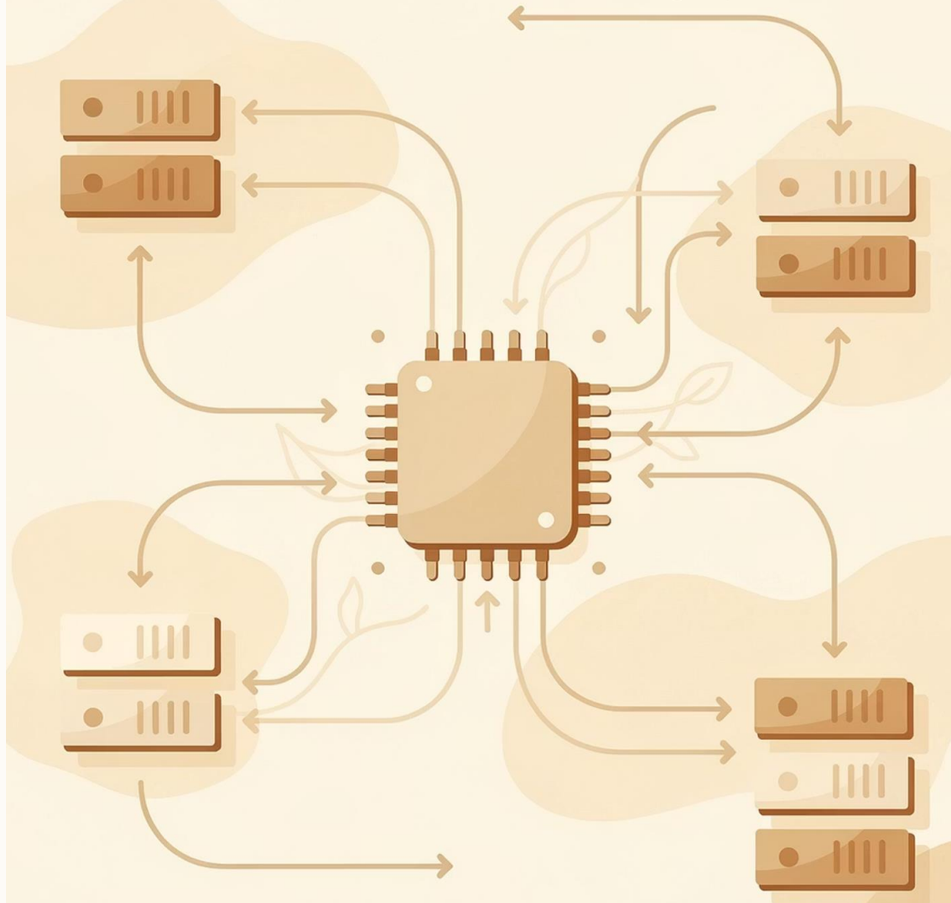
Role of Main Memory in OS

Main memory plays a central role in process execution

Process Storage

Memory Allocation

Fast Execution

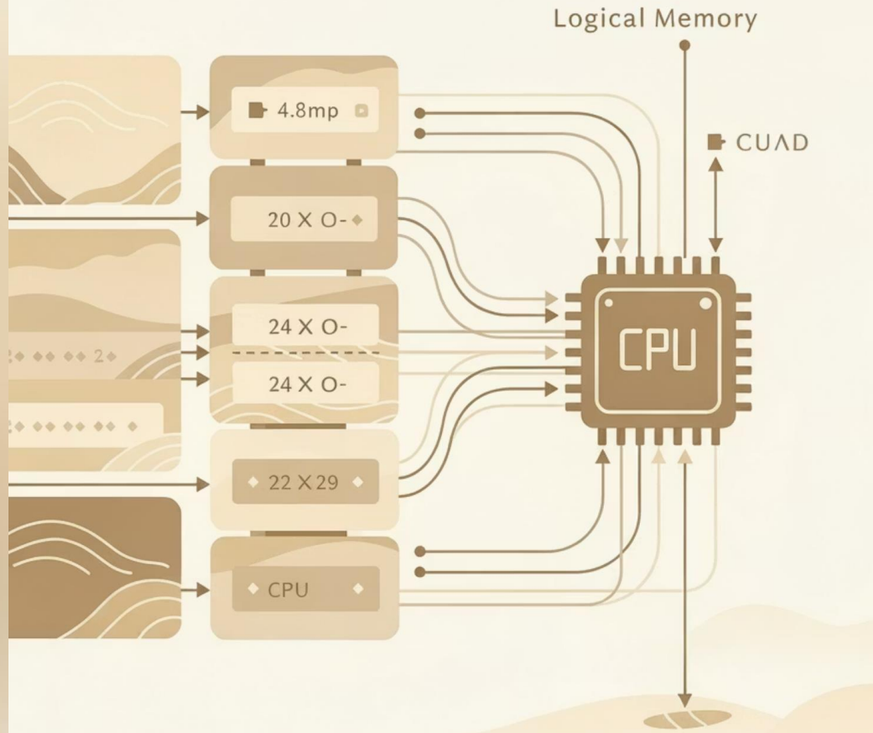


Logical Address (Virtual)

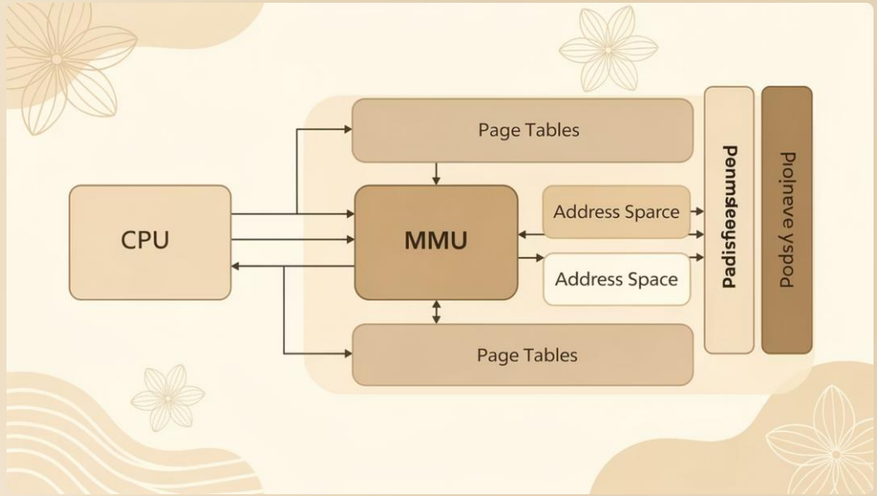
The **logical address**, also known as a virtual address, is generated by the CPU during program execution.

Program-Centric View

Key Advantages



Logical vs Physical Address



Logical Address

Virtual Address

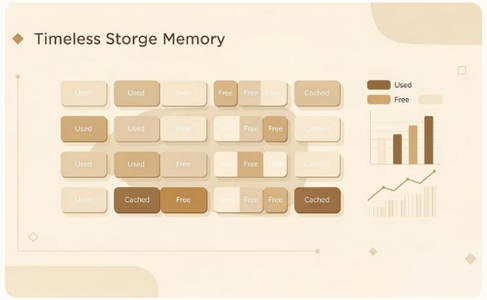
- Generated by CPU during execution
- Used by programs to refer to memory
- Each process has own address space

Physical Address

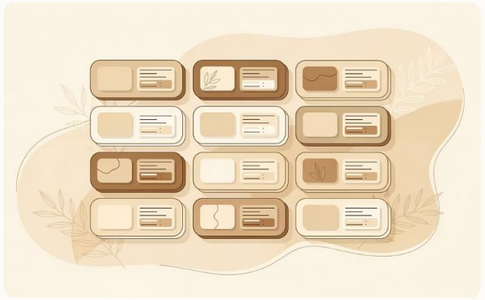
Actual Hardware Address

- Real address in main memory
- Managed by Memory Management Unit (MMU)
- Logical address converted before access

Memory Management Objectives

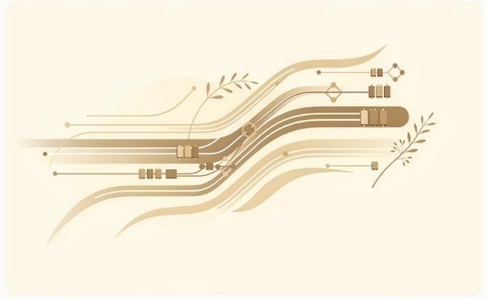


Efficient Use of Memory

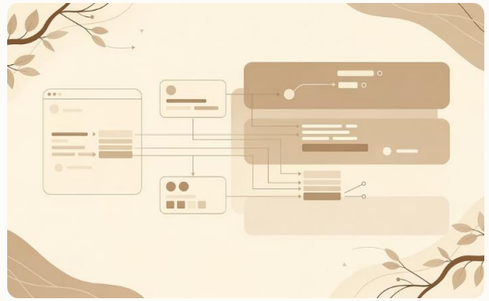


Avoid Memory Wastage

Protection & Security



Fast Memory Access

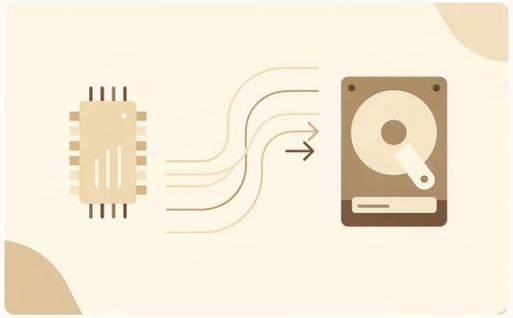


Support Multiprogramming

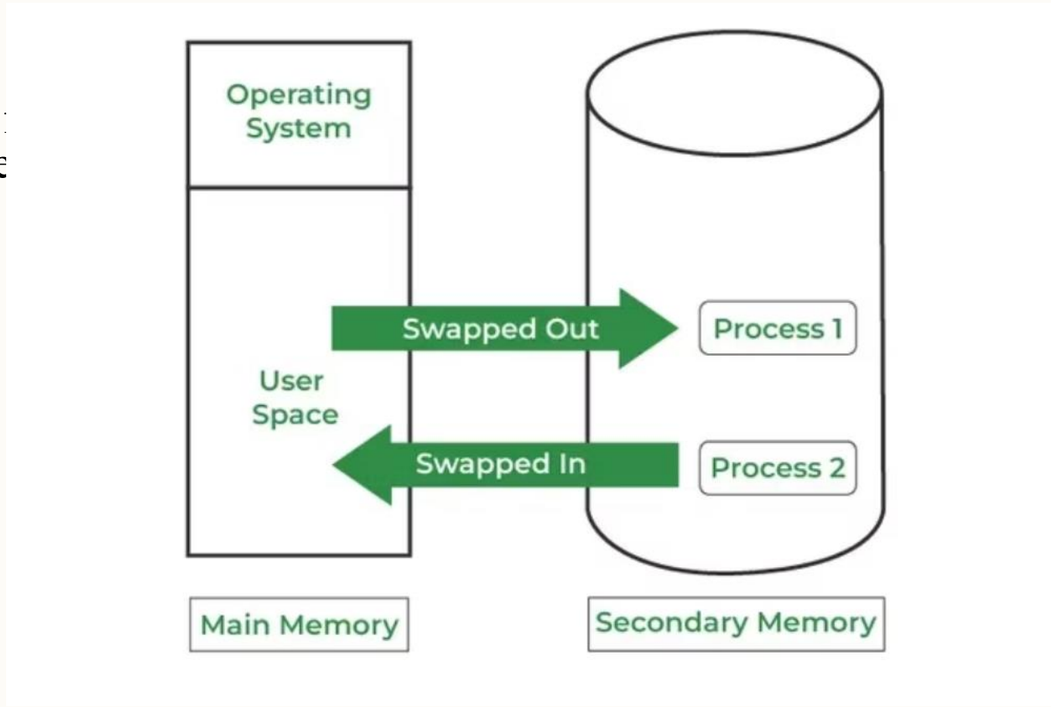
MEMORY TECHNIQUE

Swapping

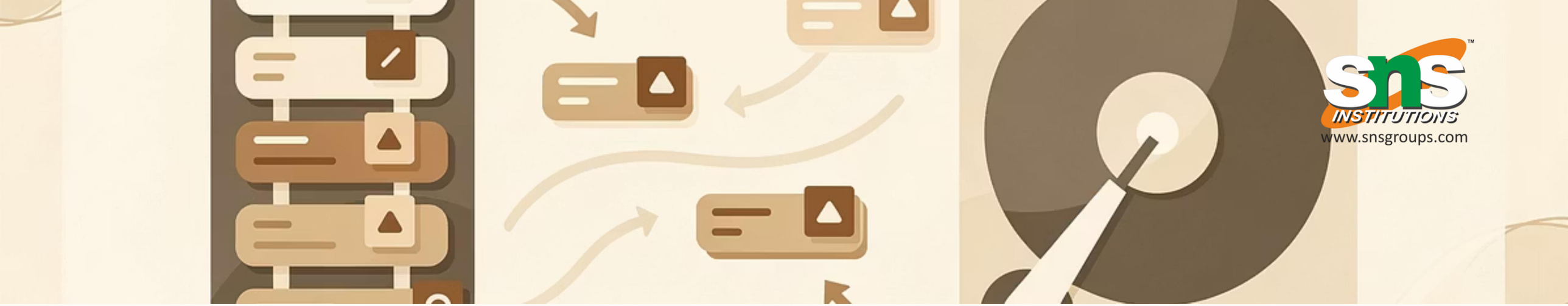
Swapping moves processes between main memory and secondary storage to manage limited memory space



Process temporarily moved
main memory to swap space



Allows multiple processes by swapping lower priority for higher priority



Why Swapping is Needed

Limited Memory

Main memory capacity is finite

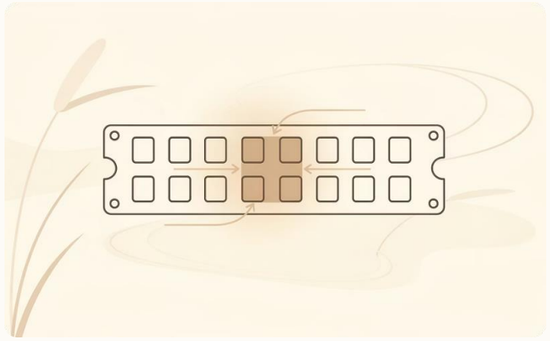
Multiprogramming

Allow more processes to run concurrently

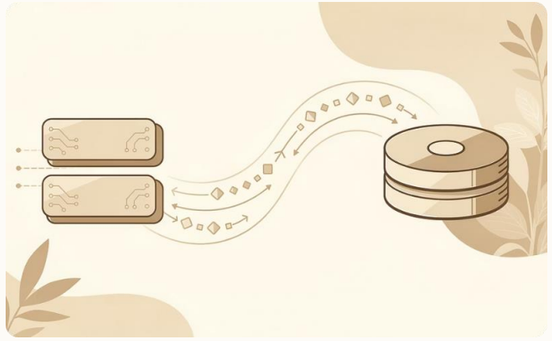
Priority Management

Free memory for higher-priority processes

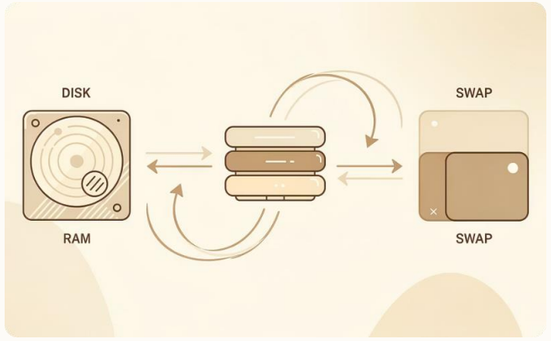
How Swapping Works



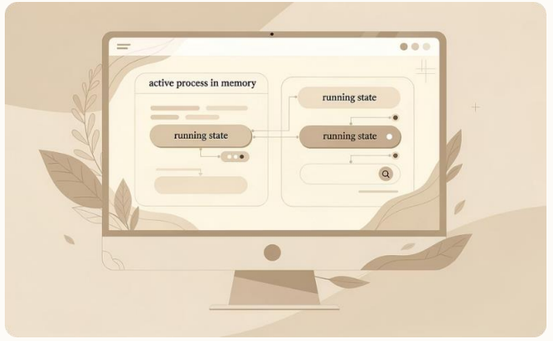
Select Process



Swap Out



Swap In



Resume

Components Involved

Main Memory

Secondary Storage

Medium-term Scheduler

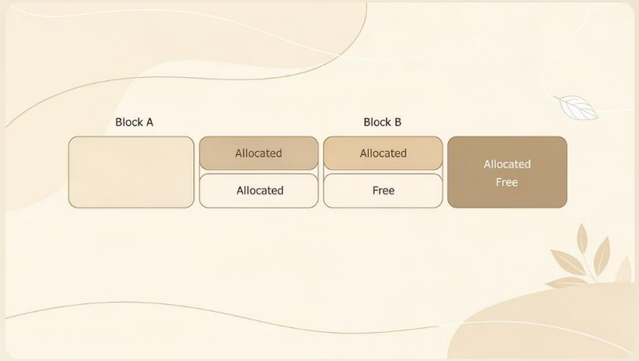
Swapping: Pros & Cons

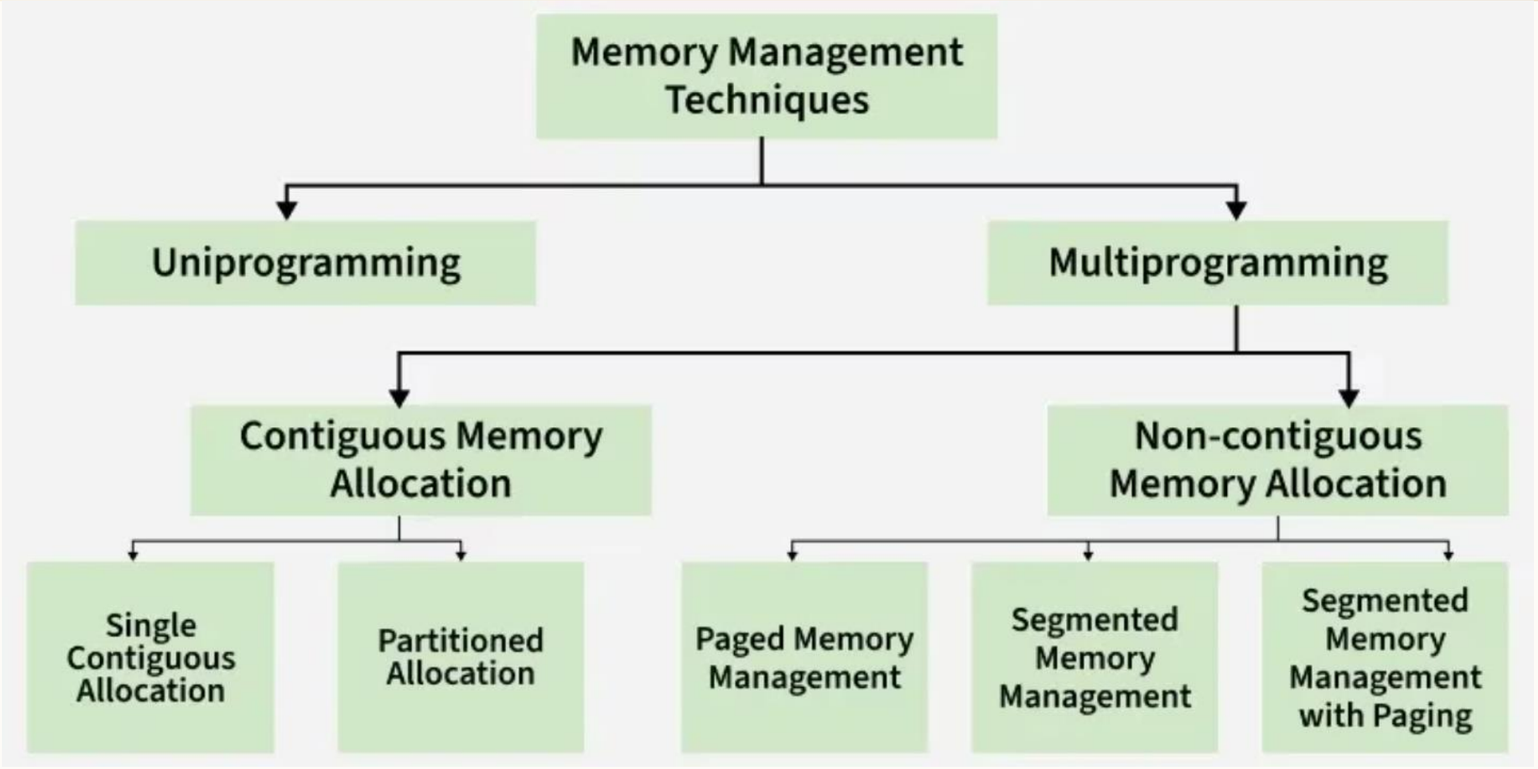
Advantages

- Increases multiprogramming degree
- Better memory utilization
- Allows execution of large number of processes

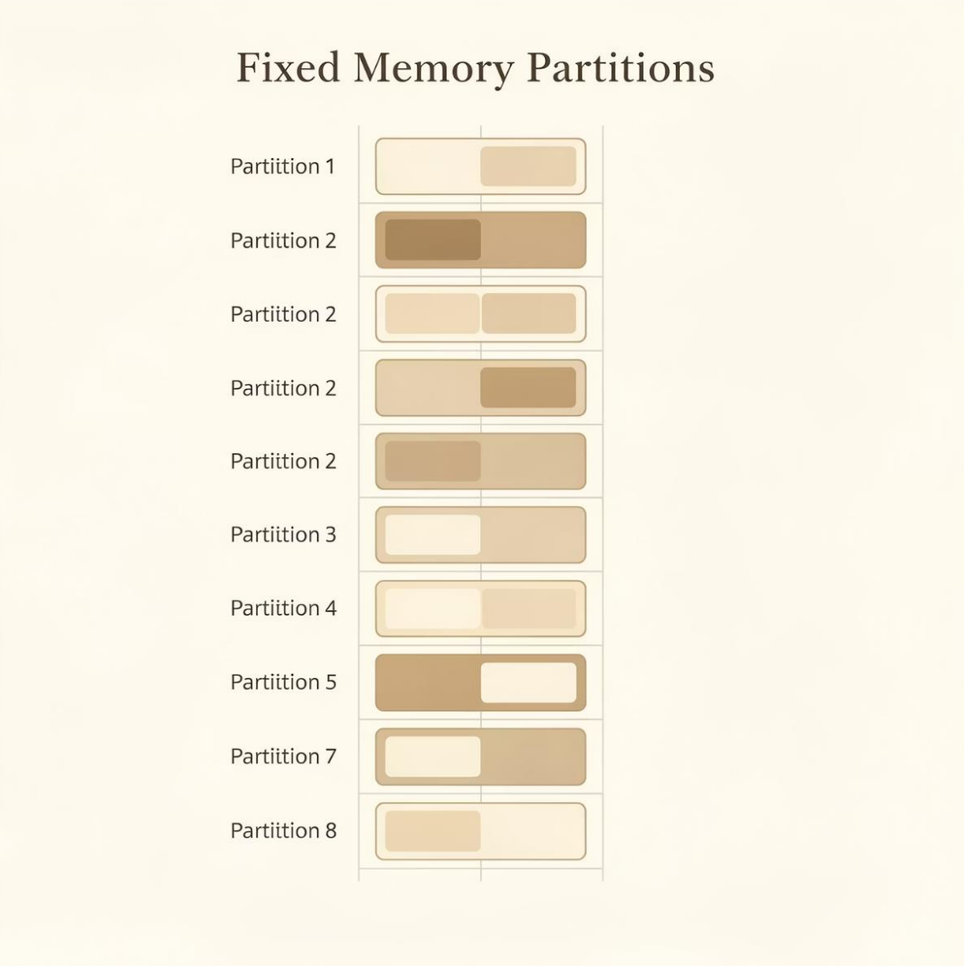
Disadvantages

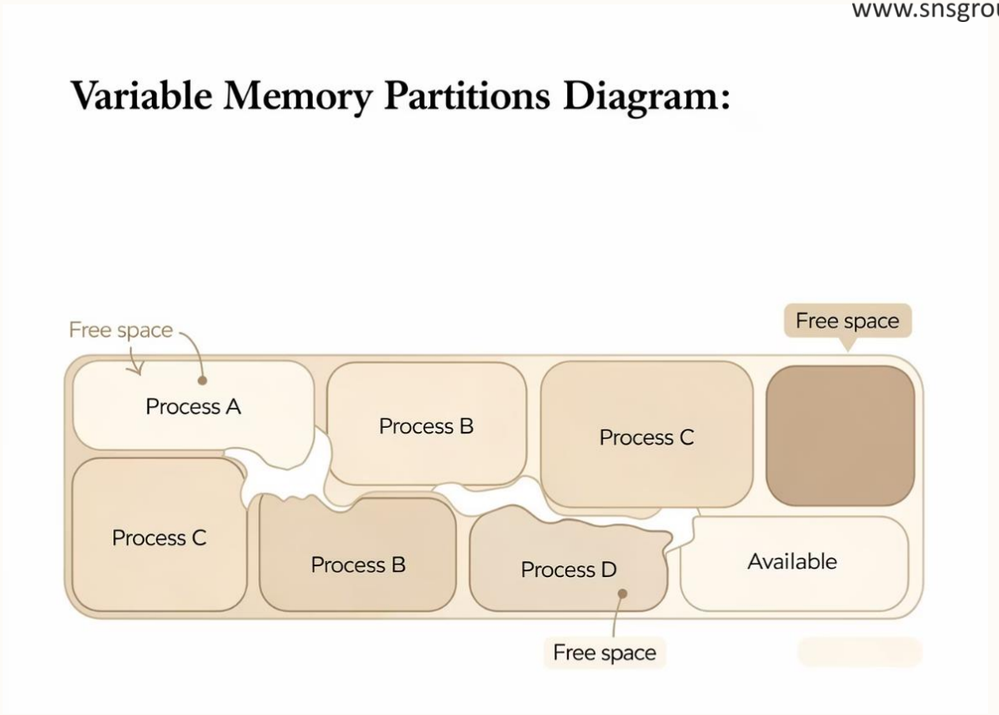
- High overhead due to disk I/O
- Slower execution
- Not suitable for real-time systems


Block A
Block B
Allocated
Allocated
Allocated
Free
Allocated
Free


Memory Management Techniques
Uniprogramming
Multiprogramming
Contiguous Memory Allocation
Non-contiguous Memory Allocation
Single Contiguous Allocation
Partitioned Allocation
Paged Memory Management
Segmented Memory Management
Segmented Memory Management with Paging

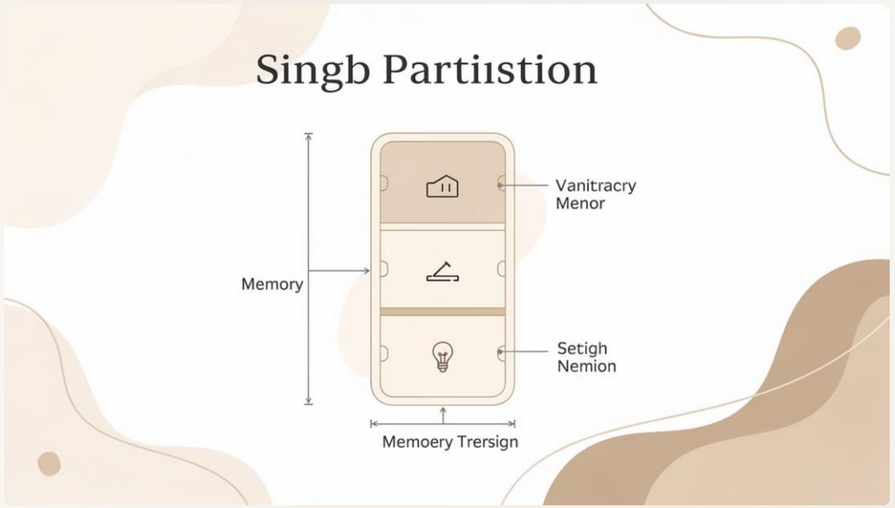
Allocation Schemes





Memory typically divied into two partitions: one for OS, one for user processes

Single Partition Allocation



Memory Division

One partition for OS, one for user process

One Process Only

Only one process loaded at a time

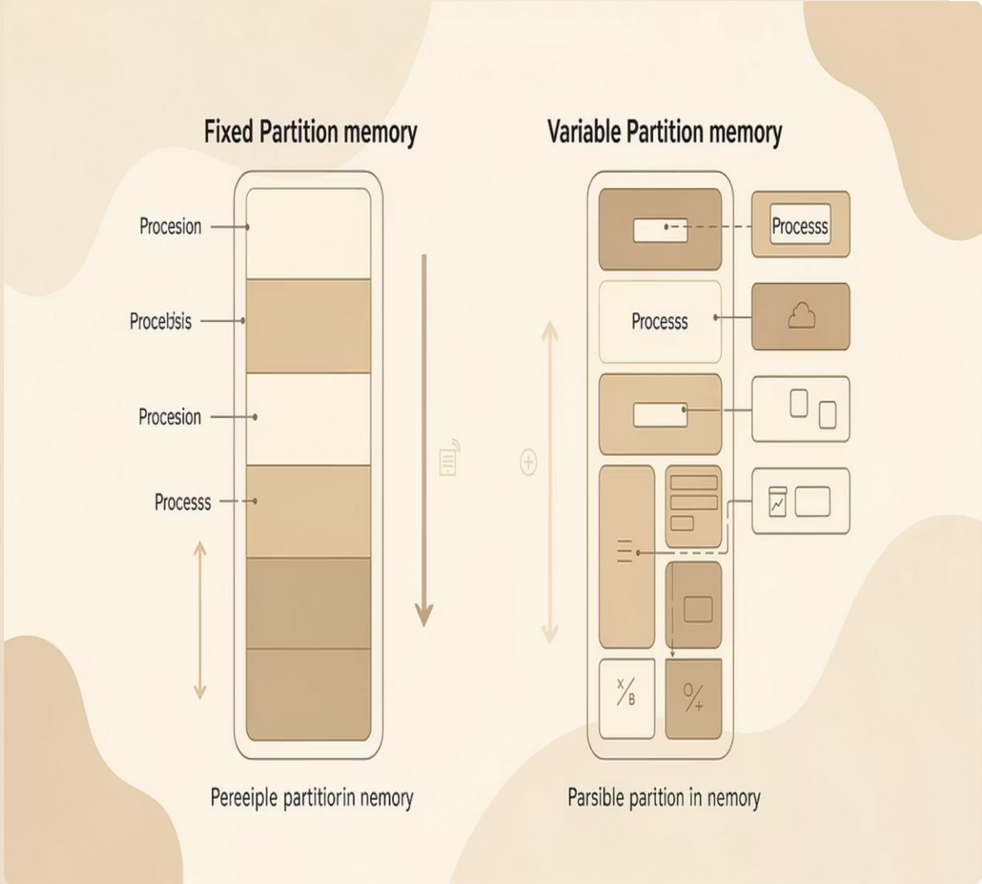
Simple Implementation

No memory sharing between processes

Drawbacks

- ✗ Poor CPU utilization
- ✗ No multiprogramming
- ✗ CPU idle during I/O

Multiple Partition Allocation



Fixed Partition

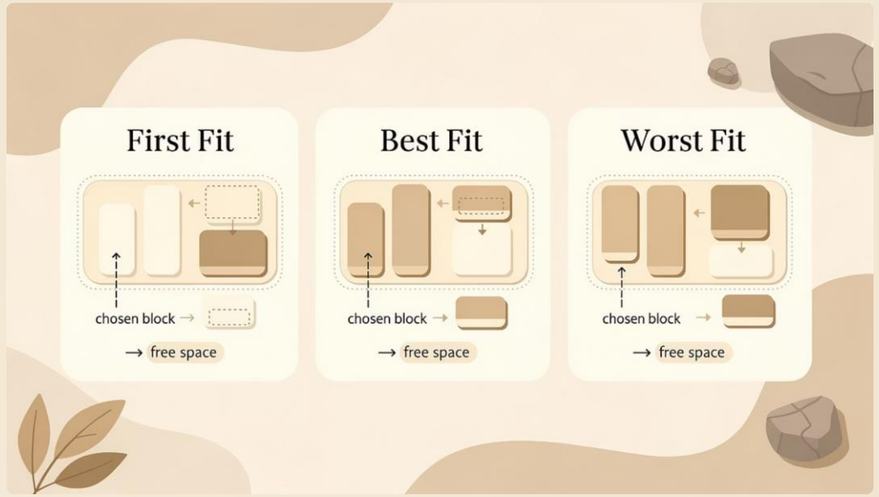
Problem: Internal Fragmentation – unused memory inside allocated partition

Variable Partition

Advantage: Better utilization, no internal fragmentation

Problem: External Fragmentation – free memory scattered in small blocks

Memory Allocation Strategies



1

First Fit

Allocate first block that is large enough

2

Best Fit

Allocate smallest block that fits the process

3

Worst Fit

Allocate largest available block

Fragmentation & Trade-offs

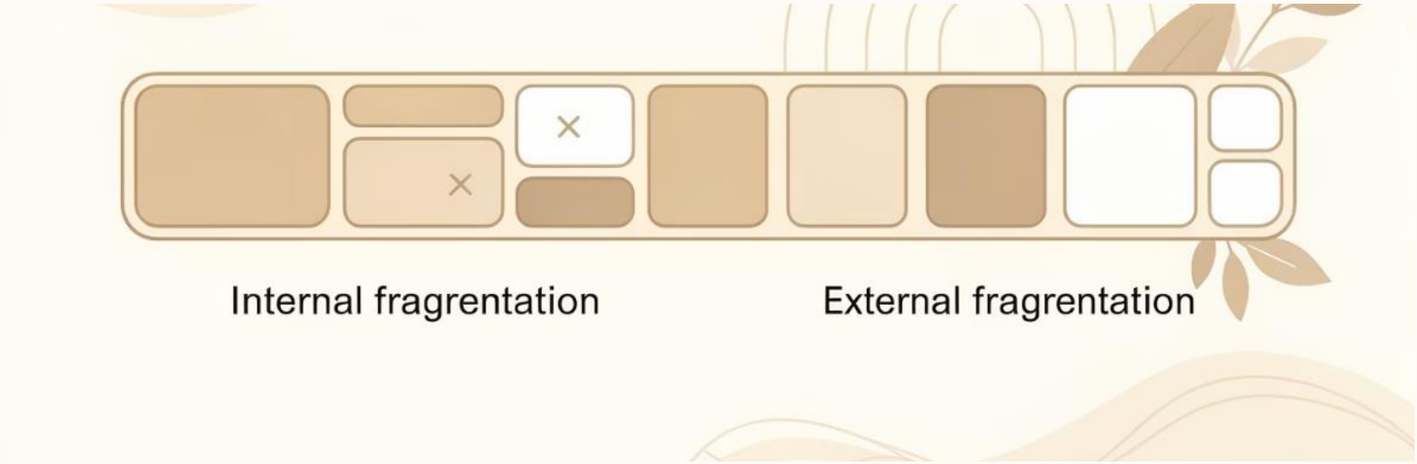
Fragmentation Types

Internal Fragmentation

Wasted memory inside allocated block

External Fragmentation

Free memory scattered in small blocks

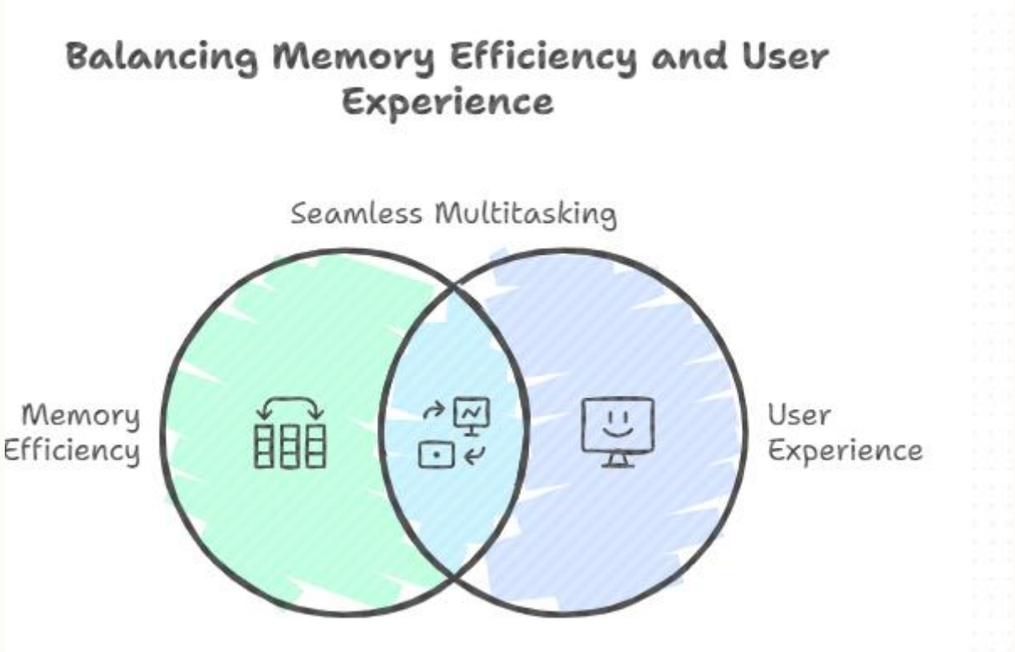


1. Problem Statement

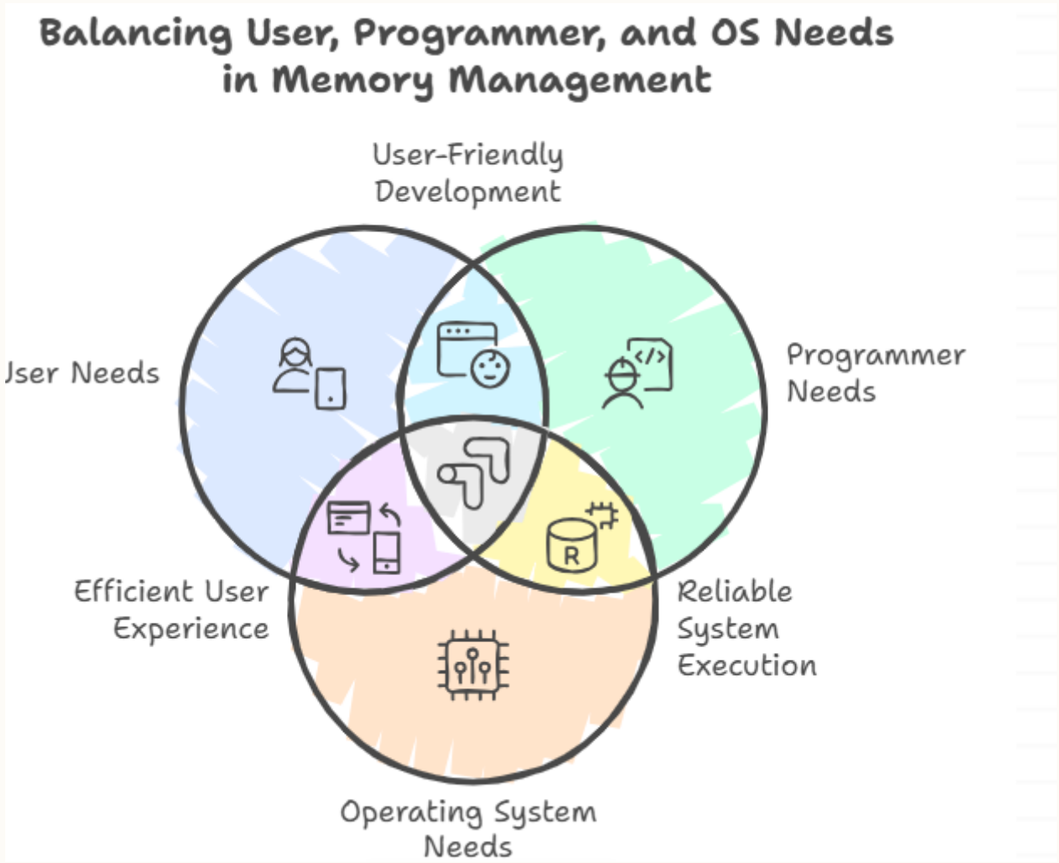
A computer system has **limited main memory**, but it must run **multiple programs simultaneously**.

Some programs are large, some are small, and users expect **fast response without crashes or delays**.

How can the operating system efficiently manage main memory, temporarily handle memory shortages, and allocate memory to processes without conflicts or wastage?



EMPATHIZE


Balancing User, Programmer, and OS Needs in Memory Management
User-Friendly Development
User Needs
Programmer Needs
Efficient User Experience
Reliable System Execution
Operating System Needs

DEFINE

Memory Management Goals

To solve the problem, the operating system must:

Manage Main Memory

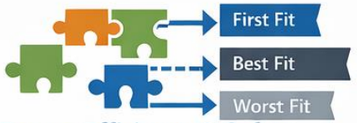
Store active processes in main memory.

Handle Memory Overflow

Move inactive processes to disk (Swapping).

Contiguous Memory Allocation

Allocate memory efficiently in one continuous block

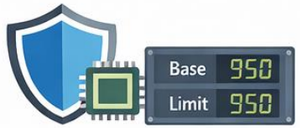


Ensure Efficiency & Safety

Protect data & minimize fragmentation.

Ensure:

- Fast access
- Data safety
- Minimal fragmentation
- High CPU utilization



Ensure Efficiency & Safety

Protect data & minimize fragmentation

IDEATE

Memory Management Solutions

Store Active Processes

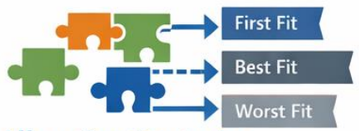
Store only active processes in main memory.

Swap Out Inactive Processes

Move inactive processes to disk.

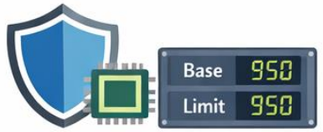
Contiguous Allocation

Allocate each process a continuous block.



Allocation Strategies

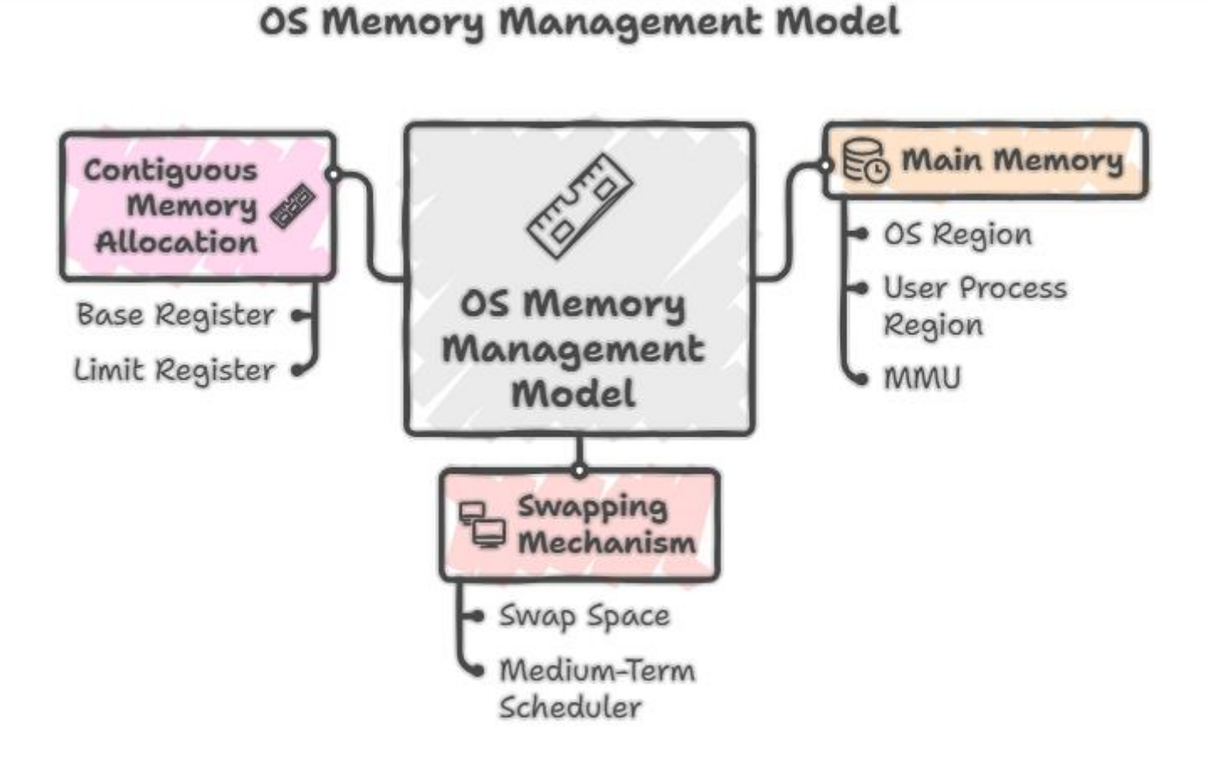
First Fit, Best Fit, Worst Fit



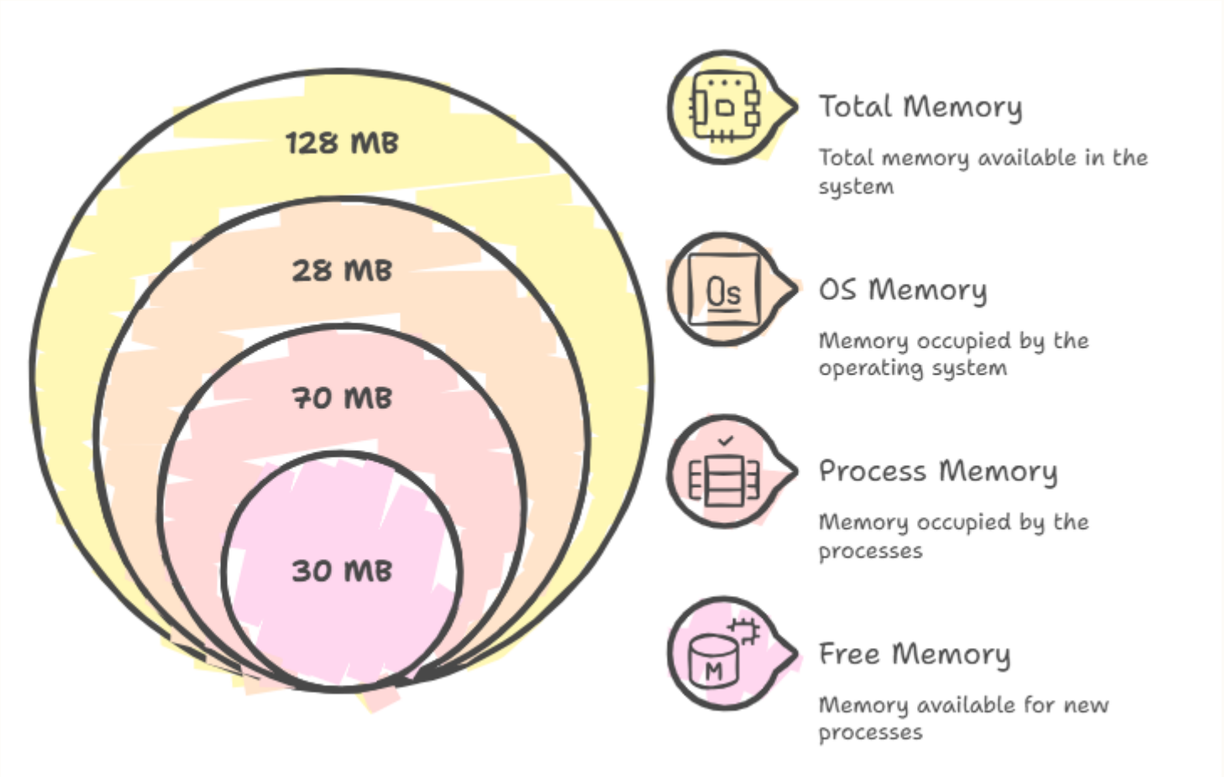
Memory Protection

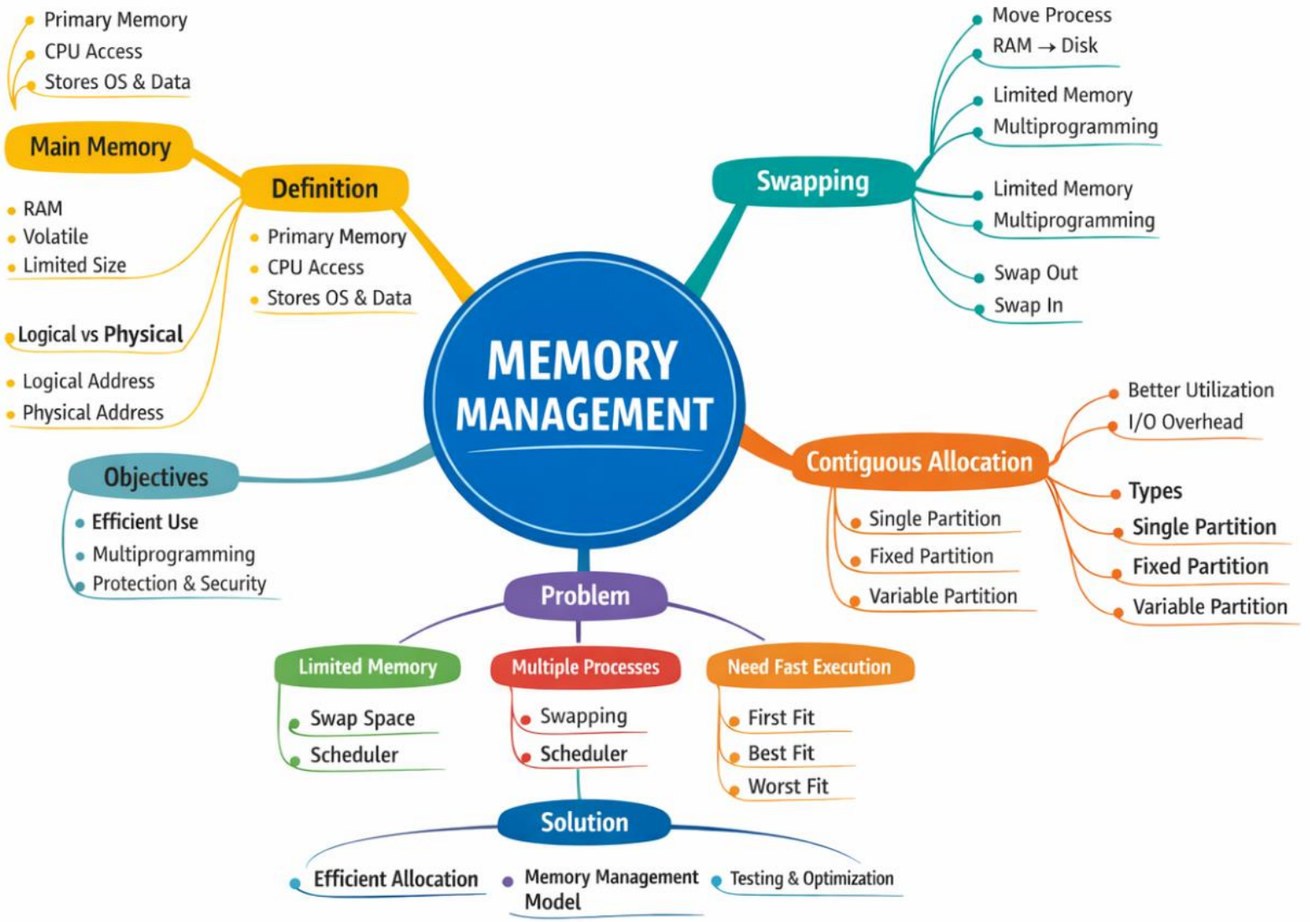
Use base & limit registers

5. PROTOTYPE


OS Memory Management Model
Contiguous Memory Allocation
Base Register
Limit Register
OS Memory Management Model
Main Memory
OS Region
User Process Region
MMU
Swapping Mechanism
Swap Space
Medium-Term Scheduler

TEST


128 MB
28 MB
70 MB
30 MB
Total Memory
Total memory available in the system
Os
OS Memory
Memory occupied by the operating system
Process Memory
Memory occupied by the processes
M
Free Memory
Memory available for new processes


MEMORY MANAGEMENT
Main Memory
Primary Memory
CPU Access
Stores OS & Data
Definition
Primary Memory
CPU Access
Stores OS & Data
RAM
Volatile
Limited Size
Logical vs Physical
Logical Address
Physical Address
Objectives
Efficient Use
Multiprogramming
Protection & Security
Swapping
Move Process
RAM → Disk
Limited Memory
Multiprogramming
Limited Memory
Multiprogramming
Swap Out
Swap In
Contiguous Allocation
Better Utilization
I/O Overhead
Types
Single Partition
Fixed Partition
Variable Partition
Single Partition
Fixed Partition
Variable Partition
Problem
Limited Memory
Swap Space
Scheduler
Multiple Processes
Swapping
Scheduler
Need Fast Execution
First Fit
Best Fit
Worst Fit
Solution
Efficient Allocation
Memory Management Model
Testing & Optimization