

# 5.1 Basic Memory Management - Partitioning, Fixed and Variable, Free Space management Techniques - Bitmap, Linked List.

---

## 5.1 Basic Memory Management

---

Memory management is a fundamental part of an operating system that handles the allocation, management, and deallocation of memory spaces to processes. Let's explore its concepts like partitioning, fixed and variable partitioning, and free space management techniques such as bitmaps and linked lists.



## Memory Partitioning

---

Partitioning is the technique used to divide the main memory (RAM) into different sections for better memory allocation and management.

### Types of Partitioning:

---

#### 1. Fixed Partitioning:

- ◆ **Definition:** Divides memory into fixed-sized partitions.
- ◆ **How it Works:** Each partition can hold only one process. A new process is allocated to the next available partition that fits its size.
- ◆ **Example:** A 256MB memory is divided into four fixed partitions of 64MB each.
- ◆ **Pros:** Simple to implement, fast allocation.
- ◆ **Cons:** Causes internal fragmentation (unused space inside a partition).

#### 2. Variable Partitioning:

- ◆ **Definition:** Divides memory into variable-sized partitions based on process requirements.
- ◆ **How it Works:** The OS allocates exactly the required memory to a process, which may change dynamically.
- ◆ **Example:** Process A needs 50MB, so it is allocated 50MB; Process B needs 30MB and gets 30MB.
- ◆ **Pros:** Reduces internal fragmentation.
- ◆ **Cons:** Can cause external fragmentation (scattered free memory blocks).

# Free Space Management Techniques

---

To efficiently utilize memory, operating systems use different techniques to manage free space. Two common techniques are **bitmaps** and **linked lists**.

## Bitmap:

---

- ◆ **Definition:** A method that uses a bit array to represent free and occupied memory blocks.
- ◆ **How it Works:** Each bit in the bitmap represents a memory block. '0' means the block is free, and '1' means it is occupied.
- ◆ **Example:** If the memory has 8 blocks, the bitmap "11001100" indicates that blocks 1, 2, 5, and 6 are occupied.
- ◆ **Pros:** Fast search for free blocks.
- ◆ **Cons:** Memory overhead increases with larger memory sizes.

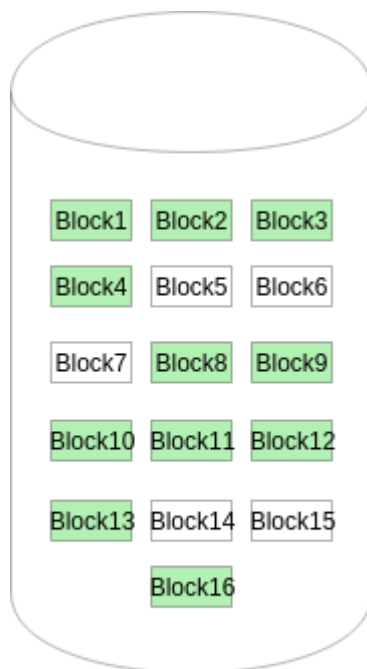


Figure - 1

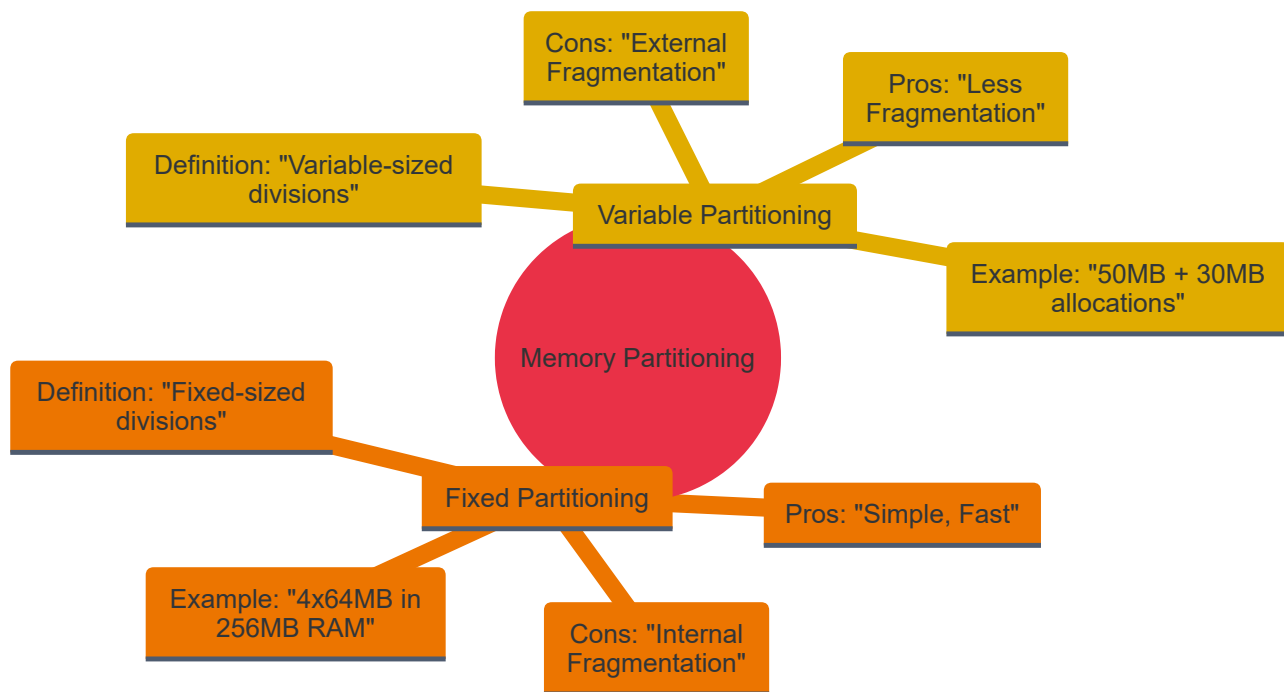
## Linked List:

---

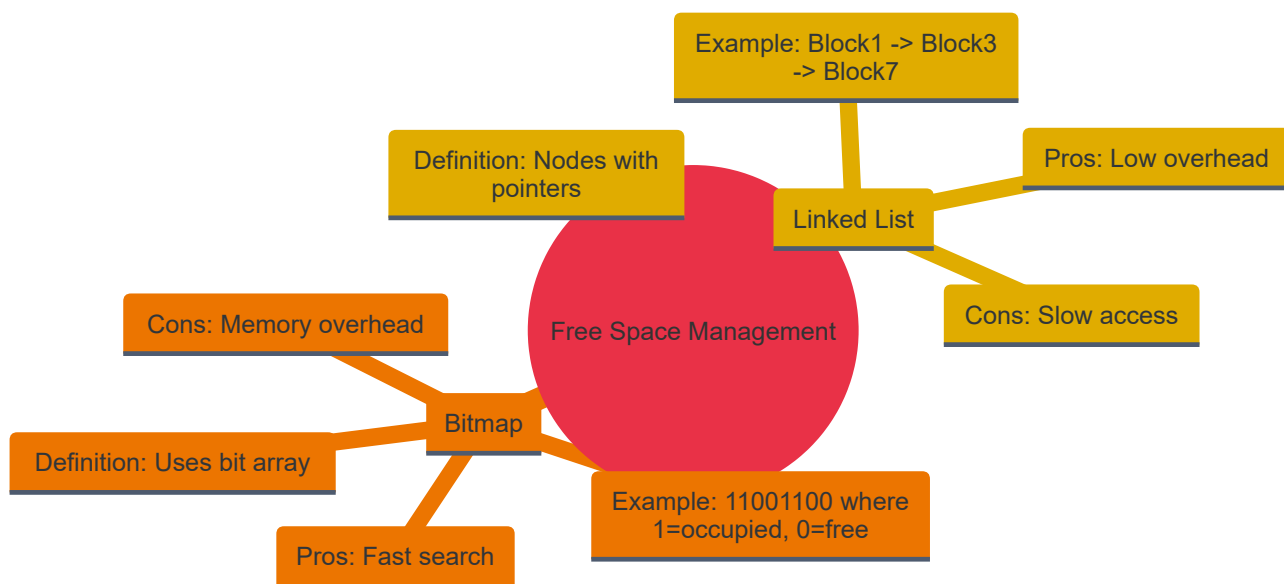
- ◆ **Definition:** A method that uses a linked list where each node represents a block or group of free blocks.
- ◆ **How it Works:** Each node contains pointers to the next block or set of free blocks. The list is traversed to find available memory.
- ◆ **Example:** A list like "Block1 -> Block3 -> Block7" indicates Blocks 1, 3, and 7 are free.
- ◆ **Pros:** Low memory overhead, flexible for dynamic allocation.
- ◆ **Cons:** Slower access time due to list traversal.

# Diagrams

## 1. Memory Partitioning Types:



## 2. Free Space Management Techniques:



## 3. Linked List Representation:

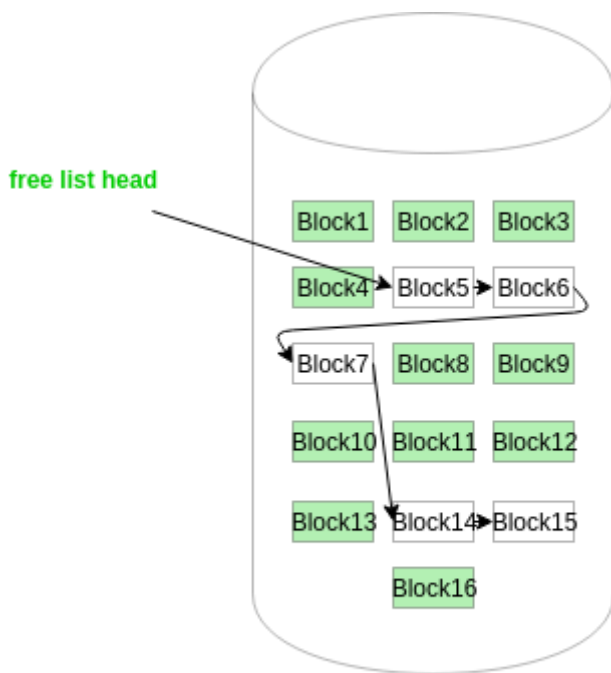


Figure - 2

## Explanation of Diagrams:

- 1. Memory Partitioning Diagram:** Visualizes the two types of memory partitioning — Fixed and Variable.
- 2. Free Space Management Diagram:** Represents the methods — Bitmap and Linked List, explaining their workings, pros, and cons.
- 3. Linked List Graph:** Shows a linked list structure of free blocks in memory, demonstrating dynamic allocation.



## Table Summary: Memory Management Techniques

Type	Definition	How It Works	Example	Pros	Cons
Fixed Partitioning	Fixed-size memory divisions	Each partition holds a single process	4 partitions of 64MB each in 256MB RAM	Simple, Fast	Internal Fragmentation
Variable Partitioning	Variable-size memory divisions	Allocates memory as needed	Allocating 50MB + 30MB	Less Internal Fragmentation	External Fragmentation
Bitmap	Uses a bit array to represent	'0' for free, '1' for	Bitmap: 11001100	Fast search	Memory overhead

Type	Definition	How It Works	Example	Pros	Cons
	memory blocks	occupied blocks			
Linked List	Uses nodes with pointers representing free memory blocks	Nodes point to next free memory block	Block1 -> Block3 -> Block7	Low memory overhead	Slower access, list traversal



## Conclusion:

- ◆ **Partitioning Techniques:** Help manage how memory is allocated and organized, with different methods to balance simplicity, speed, and fragmentation.
- ◆ **Free Space Management:** Efficient memory use is ensured through techniques like Bitmap and Linked List, each with unique advantages and disadvantages based on speed, memory overhead, and allocation flexibility.