

SHREE ADARSHA SECONDARY SCHOOL

**A Report**

**For**

**Operating System**

**On**

**“Memory Hierarchy, Memory Management functions, Static &dynamic Partitioning, Internal& External fragmentation and the concept of virtual Memory. “**



**Submitted by:**

**Name:**

**Class:**

**Roll No:**

**Date:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Pradip khatiwoda**

**Submitted to:**

Contents

[Memory Management in Operating Systems 1](#_Toc161084372)

[1. Introduction 1](#_Toc161084373)

[2. Memory hierarchy 1](#_Toc161084374)

[3. Memory Management Functions 2](#_Toc161084375)

[4. Static and Dynamic Partitioning 2](#_Toc161084376)

[5. Internal and External Fragmentation 4](#_Toc161084377)

[**Internal Fragmentation:** 5](#_Toc161084378)

[**External Fragmentation**: 5](#_Toc161084379)

[6. Concept of Virtual Memory 5](#_Toc161084380)

[7. Comparison of Memory Management Techniques 6](#_Toc161084381)

[7. Conclusion 7](#_Toc161084382)

# Memory Management in Operating Systems

**Memory Management in Computer Systems**

## Memory Management in OSIntroduction

Figure 1:Memory management

Memory management is a crucial aspect of modern computer systems, playing a vital role in the efficient utilization of memory resources. It involves a set of techniques and strategies aimed at optimizing the use of available memory, ensuring smooth operation of software applications and system processes. This report explores various aspects of memory management, including memory hierarchy, memory management functions, partitioning strategies, fragmentation issues, and the concept of virtual memory.

2. Memory hierarchy

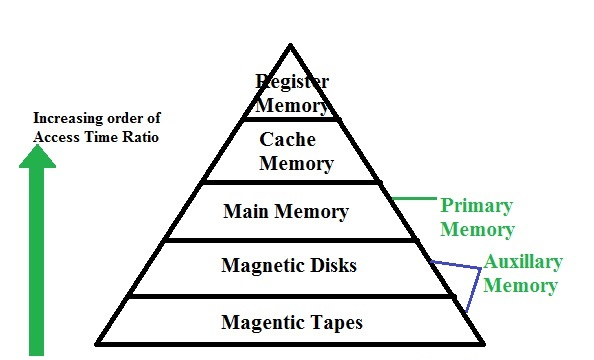
The memory hierarchy in a computer system consists of multiple levels of memory with varying speeds, capacities, and costs. At the top of the hierarchy are registers, small storage units directly accessible by the CPU for storing frequently accessed data and instructions. Next are the cache memories, which serve as intermediate storage between registers and main memory (RAM), offering faster access times but limited capacity. Main memory, or RAM, holds program data and instructions that are actively used by the CPU, while secondary storage, such as hard disk drives (HDDs) or solid-state drives (SSDs), provides long-term storage for data and programs. Understanding the memory hierarchy is essential for designing efficient memory management systems that balance speed, capacity, and cost. The memory hierarchy is a fundamental concept in computer architecture, consisting multiple levels that enable efficient data processing and storage.

Figure 2:Memory hierarchy

3. Memory Management Functions

Memory management functions are responsible for allocating and deallocating memory resources to processes, as well as managing memory fragmentation. Memory allocation involves assigning memory blocks to processes based on their size and memory requirements. Different allocation strategies, such as contiguous and non-contiguous allocation, offer trade-offs between memory utilization and fragmentation. Memory deallocation, on the other hand, involves releasing memory blocks that are no longer in use, allowing them to be reused by other processes. However, improper memory deallocation can lead to fragmentation, both internally within allocated memory blocks and externally in the overall memory space. This section will explore various memory management functions and strategies for optimizing memory usage while minimizing fragmentation.

4. Static and Dynamic Partitioning

Memory partitioning techniques are used to divide the available memory space into smaller segments or partitions, which are then allocated to processes. These techniques provide a structured approach to managing memory resources efficiently.

**Static Partitioning**: Static partitioning involves dividing the available memory into fixed-sized partitions during system initialization. Each partition has a predetermined size, and once created, it remains unchanged throughout the execution of the system. Processes are allocated to partitions based on their size, and each partition can accommodate only one process at a time.

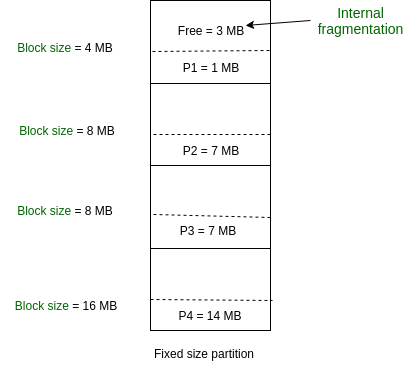


Figure 3: Internal fragmentation

Advantages of Static Partitioning:

1. Simplicity: Static partitioning is straightforward to implement and manage since partition sizes are fixed.
2. Predictability: System behavior is predictable as partition sizes do not change during runtime.

Disadvantages of Static Partitioning:

1. Wastage of Memory: Fixed partition sizes may lead to inefficient use of memory, especially when processes have varying memory requirements.
2. Limited Flexibility: Static partitioning does not adapt well to changing memory demands, leading to potential underutilization or fragmentation of memory.

**Dynamic Partitioning**: Dynamic partitioning, also known as variable partitioning, allows the size of memory partitions to vary dynamically based on the memory requirements of processes. Memory is divided into variable-sized partitions, and each partition is allocated to a process based on its size. When a process requests memory, the operating system dynamically allocates a partition of suitable size from the available free memory.

Advantages of Dynamic Partitioning:

1. Flexibility: Dynamic partitioning adapts to the varying memory requirements of processes, leading to efficient utilization of memory.
2. Reduced Fragmentation: Variable-sized partitions help reduce internal fragmentation by allocating memory based on process requirements.

Disadvantages of Dynamic Partitioning:

1. Overhead: Managing variable-sized partitions incurs overhead in terms of memory management and allocation algorithms.
2. Fragmentation: Although dynamic partitioning reduces internal fragmentation, it may still lead to external fragmentation as memory blocks are allocated and deallocated over time.

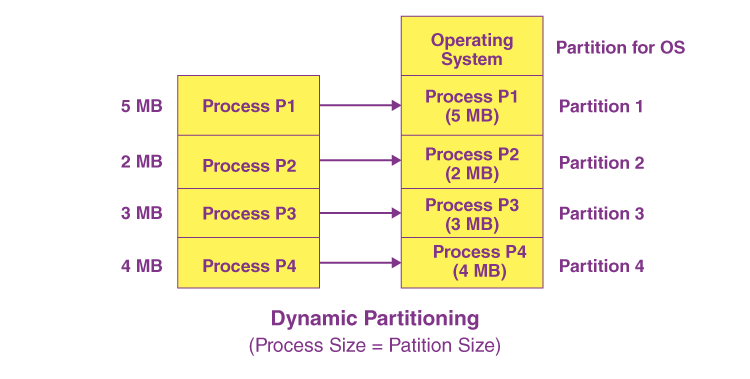


Figure 4:Dynamic Partitioning

5. Internal and External Fragmentation

Fragmentation is a common issue in memory management that can lead to inefficient use of memory resources. Internal and external fragmentation are two types of fragmentation that occur in memory allocation and deallocation processes.

### **Internal Fragmentation:**

Figure 5:Internal fragmentation

Internal fragmentation occurs when allocated memory blocks have unused space within them because they are larger than the requested memory size. This means that some memory within the allocated block remains unused, contributing to inefficiency. For example, if a process requires 70 bytes of memory but is allocated a fixed-size block of 100 bytes, the remaining 30 bytes are wasted, resulting in internal fragmentation.

### **External Fragmentation**:

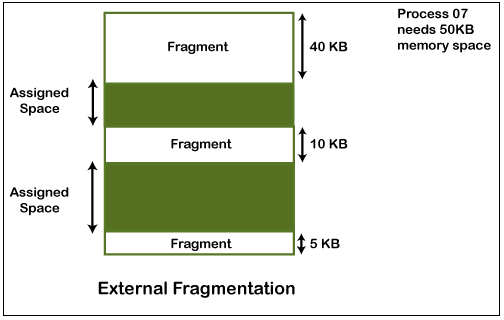
External fragmentation arises when there is enough total free memory available to satisfy a memory request, but this free memory is not contiguous. This fragmented nature of free memory makes it challenging to allocate contiguous blocks of memory for processes, despite there being sufficient total free memory. External fragmentation occurs due to the allocation and deallocation of variable-sized memory blocks over time, leaving gaps or fragmented free memory scattered throughout the memory space.

Figure 6: External fragmentation

6. Concept of Virtual Memory

Virtual memory is a memory management technique that extends the available physical memory by using disk space as an extension. It allows processes to use more memory than physically available by swapping data between main memory and secondary storage. Demand paging is a virtual memory technique that loads only necessary pages into main memory, swapping others to disk as needed. Page replacement algorithms, such as least recently used (LRU) and first-in, first-out (FIFO), determine which pages to swap out of memory when space is needed. While virtual memory offers advantages such as increased memory capacity and support for multitasking, it also introduces overhead due to disk I/O operations and page faults.

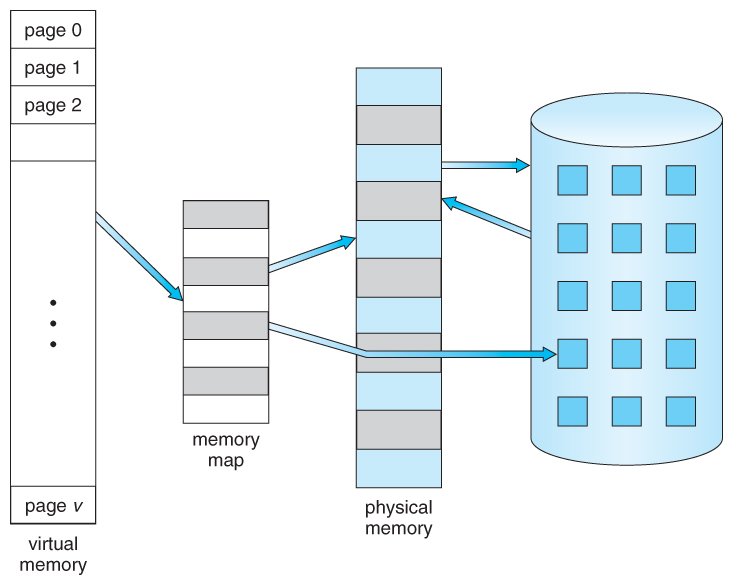


Figure 7: Concept of virtual memory

7. Comparison of Memory Management Techniques

| **Technique** | **Description** | **Advantages** | **Disadvantages** |
| --- | --- | --- | --- |
| Static Partitioning | Fixed-size partitions allocated to processes | Simple implementation | Wasteful for variable-sized processes |
| Dynamic Partitioning | Memory allocated dynamically as needed | Efficient use of memory | Fragmentation issues |
| Virtual Memory | Illusion of larger memory space using secondary storage | Allows for larger processes | Performance overhead, page faults |

7. Conclusion

Efficient memory management is crucial for optimizing system performance and resource utilization in computer systems. This report has provided a comprehensive overview of memory management, covering memory hierarchy, memory management functions, partitioning techniques, fragmentation, and virtual memory. By understanding these concepts and techniques, system designers and developers can design efficient memory management systems that meet the performance and scalability requirements of modern computing environments.

**8. References**