

# Abstract

A close-up of a text

Description automatically generated

# Objective

A close-up of a text

Description automatically generated

Contents

[Abstract 2](#_Toc141057507)

[Objective 2](#_Toc141057508)

[Memory Management 5](#_Toc141057509)

[Need for memory management. 5](#_Toc141057510)

[Types of Memory Management Techniques 6](#_Toc141057511)

[Contiguous memory management schemes 6](#_Toc141057512)

[Fixed Partitions 6](#_Toc141057513)

[Variable Partitions 8](#_Toc141057514)

[Non-Contiguous memory management schemes 10](#_Toc141057515)

[Segmentation 10](#_Toc141057516)

[Paging 11](#_Toc141057517)

[Virtual Memory 13](#_Toc141057518)

[How it functions 13](#_Toc141057519)

[Advantages 13](#_Toc141057520)

[Disadvantages 14](#_Toc141057521)

[Suitable situations 14](#_Toc141057522)

[Comparison 15](#_Toc141057523)

[Conclusion 16](#_Toc141057524)

[Reference 17](#_Toc141057525)

# Table of figures

[Figure 1 5](#_Toc141057538)

[Figure 2 6](#_Toc141057539)

[Figure 3 8](#_Toc141057540)

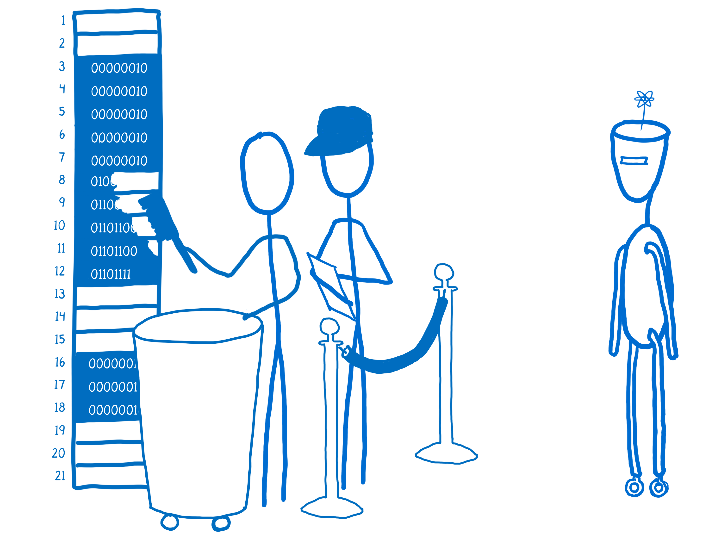
[Figure 4 10](#_Toc141057541)

[Figure 5 12](#_Toc141057542)

[Figure 6 13](#_Toc141057543)

# Memory Management

***Memory management*** is an essential component of modern operating systems, ensuring the optimal allocation and utilization of memory resources. It addresses issues with system performance, process isolation, and physical memory limitations. Operating systems can allocate memory to programs, optimize resource usage, enable virtual memory, and improve system performance by properly managing memory. Understanding multiple memory management strategies is crucial for choosing the best course of action in various situations.

  
Figure 1

## Need for memory management.

* **Effective memory distribution:** Control memory resources to distribute memory among processes and applications.
* **Process isolation:** Ensure that processes are kept separate from one another to avoid interference and ensure system stability.
* **Optimal utilization:** Increase resource utilization by increasing concurrent processes and reducing memory waste.
* **Support for virtual memory:** Provides access to memory that is not physically possible, enabling effective memory sharing and the execution of larger programs.
* **Handling fragmentation:**  Address memory fragmentation problems to maintain effective memory usage.
* **Performance of the system:** Speed for initial access and responsiveness are directly impacted by efficient memory management.

# Types of Memory Management Techniques

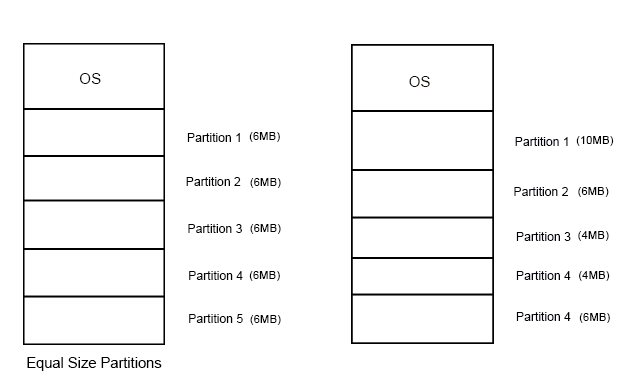
Memory management techniques are the operating systems' strategies and processes for efficiently allocating, monitoring, and utilizing memory resources. These methods make sure that programs and processes can access the memory they need to function. Memory management techniques are the operating systems' strategies and processes for efficiently allocating, monitoring, and utilizing memory resources. These methods make sure that programs and processes can access the memory they need to function.

## Contiguous memory management schemes

Contiguous memory management schemes are memory management approaches that assign memory to processes in a continuous, contiguous fashion. The memory blocks allotted to processes under these schemes must be sequential and close to one another.

### Fixed Partitions

Fixed Partition Allocation is a method of managing contiguous memory that divides the memory into fixed-size partitions or regions. Multiple processes may operate simultaneously in each partition, which is allocated to a single process.

   
Figure 2

#### How it functions

* At system startup or initialization, fixed-size memory divisions are created.
* Every partition has a fixed size that is normally determined by the needs of the operating system or the biggest process.
* Based on the processes' needs for memory, the partitions are assigned to them.
* Only partition-compatible processes can be loaded and run.

#### Advantages

* **Simpleness:** Fixed Partition Allocation is an easy to implement and comprehend memory management method.
* **Predictability:** Because of the set division sizes, memory allocation behavior is predictable.

#### Disadvantages

* **Internal fragmentation:** This can happen if a process does not fit completely inside of a partition, wasting memory space.
* **Limited flexibility:** The size of processes that can be supported is constrained by the fixed partition sizes, which could result in ineffective memory usage.

#### Suitable situations

* Fixed Partition Allocation works well in situations when processes' memory needs are well-known and generally consistent.
* In specialized or embedded systems where processes have particular memory needs, it may be appropriate.
* It can also be utilized when simplicity and predictability are the main priorities and multiprogramming is not necessary.

Overall, Fixed Partition Allocation is straightforward and provides process isolation, but it may also suffer from internal fragmentation and restricted flexibility. It works well in situations with predictable and stable memory requirements, like embedded devices or systems with specific functions.

### Variable Partitions

A memory management strategy utilized in contiguous memory management plans is variable partition allocation. To allow processes with differing memory requirements, the memory is partitioned into groups of varying sizes.

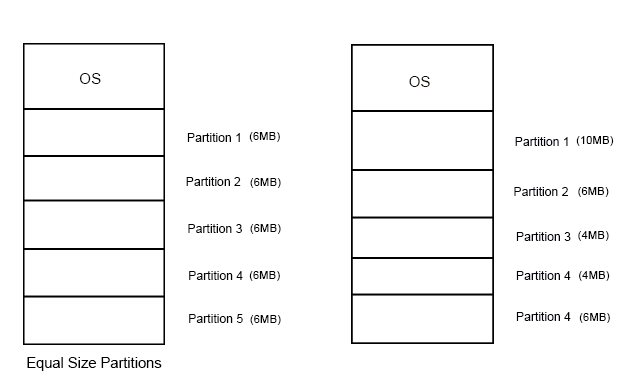


Figure 3

#### How it functions

* Depending on the needs of the process, different-sized memory partitions are created.
* Memory is filled with processes, and portions are then dynamically allocated to fit each process.
* When a process is finished, the related partition is deallocated, and the memory that was released is combined with neighboring partitions.

#### Advantages

* Efficient memory use by assigning partitions according to process requirements.
* Flexibility to allow for various-sized processes.

#### Disadvantages

* Both internal and external fragmentation is possible.
* Dynamic partition management procedures' overhead.

#### Suitable situations

* Settings with several programs where processes of different sizes coexist.
* Scenarios with changing memory needs.

In conclusion, variable partition allocation improves memory utilization and allows for a variety of process sizes. Although it provides effective usage and flexibility, it might be fragmented and result in extra costs. It is best suited for situations with changeable memory requirements and multi-programming environments.

## Non-Contiguous memory management schemes

Non-contiguous memory management schemes are memory management strategies that don't demand that processes be allocated contiguous memory blocks. These methods make use of a number of mechanisms to let processes receive memory allotments that are dispersed or fragmented across the system.

### Segmentation

A non-contiguous memory management strategy called segmentation divides the logical memory into segments of varying sizes, each of which represents a different stage of a process.

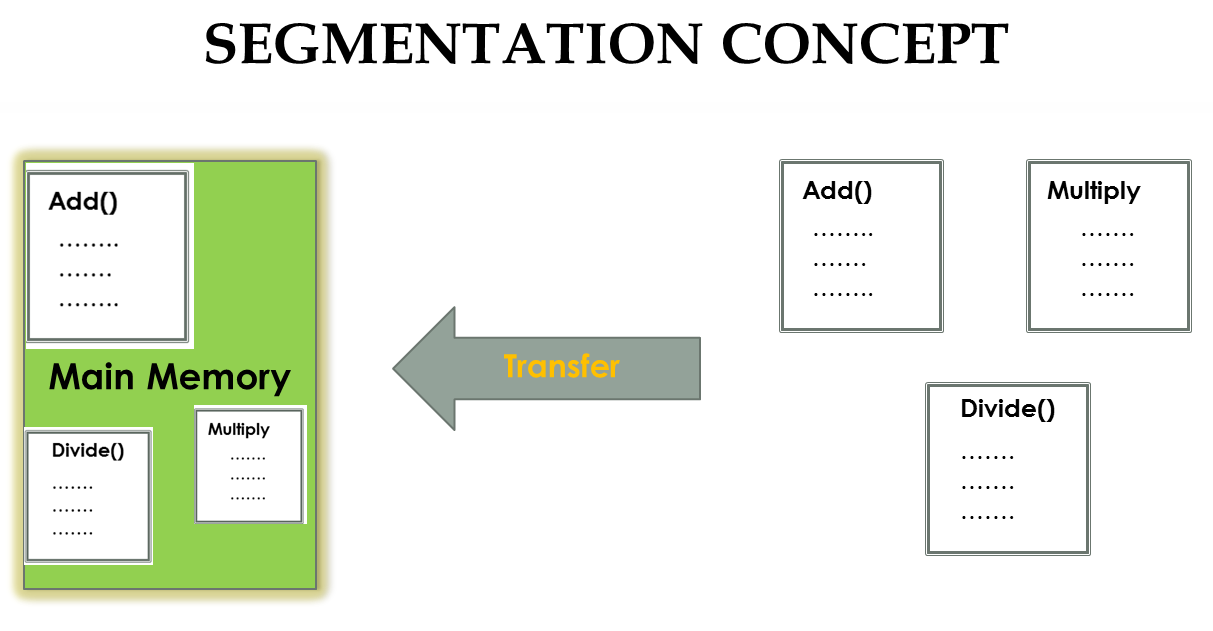


Figure 4

#### How it functions

* Each segment has a particular size and starting address and serves as a logical division of a process.
* Memory allocation is flexible since segments can be put in any free position in memory.
* The logical addresses of the process are mapped to the corresponding physical addresses using a segment table.

#### Advantages

* **logical organization**: Segmentation enables effective administration and access to various process areas.
* **Sharing and Protection**: By allowing processes to share segments, memory use is improved and restricted access is made possible.

#### Disadvantages

* **Fragmentation**: Segmentation can cause external fragmentation, which lowers memory effectiveness.
* **External Restrictions**: There may be restrictions due to the maximum number and size of segments.

#### Suitable situations

* **Variable Memory Requirements:** Segmentation is appropriate in situations where processes have variable memory requirements.
* **Dynamic Memory Usage:** Programs that dynamically create and deallocate memory benefit from segmentation.

Summary: Memory management may be more efficiently managed, shared, and protected when it is segmented. It works effectively in situations where memory requirements are varied and memory utilization is dynamic, although fragmentation and outside constraints may be an issue.

### Paging

The logical memory and physical memory are divided into fixed-sized blocks called pages by the non-contiguous memory management strategy known as paging. A contiguous block of memory is represented by each identically sized page.

#### How it functions

* A process's logical memory is split up into fixed-sized pages.
* A similar fixed-size division of the physical memory into pages-sized frames is also present.
* To efficiently access memory, a page table is utilized to map the logical pages to the physical frames.

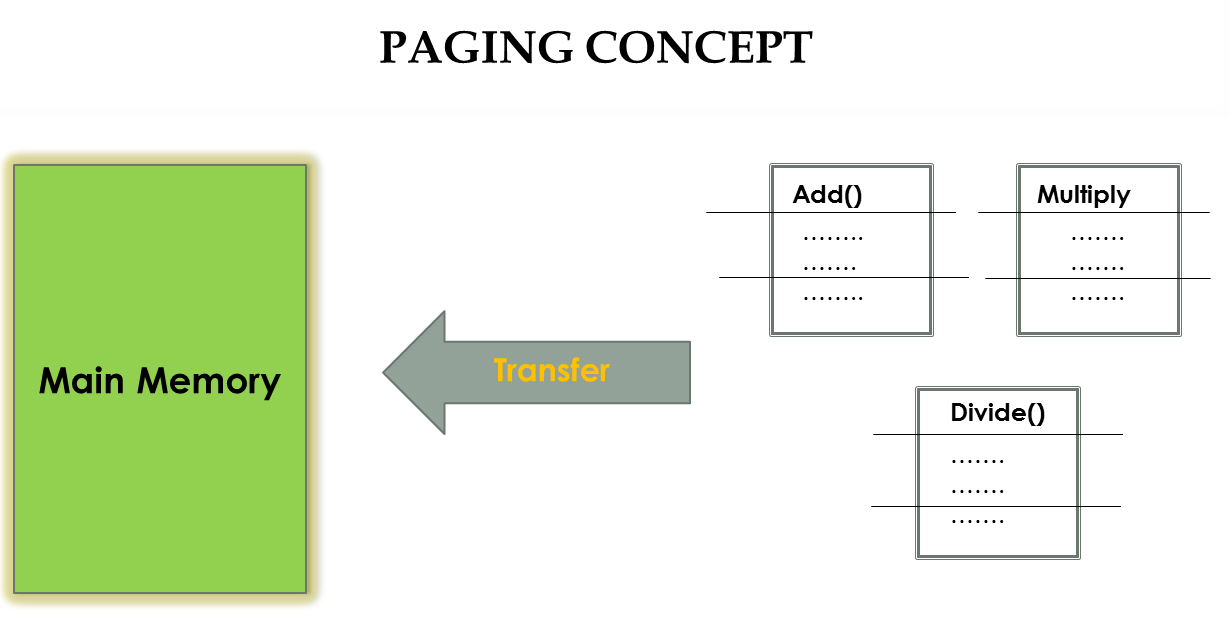


Figure 5

#### Advantages

* Memory was divided into equal-sized chunks, which simplified memory management.
* Non-contiguous allocation promotes effective memory usage.

#### Disadvantages

* Internal memory fragmentation as a result of partially used memory frames.
* Overhead due to page table lookups and memory mapping.

#### Suitable situations

* Systems having virtual memory and limited physical memory.
* Systems that require fixed-size memory allocation.

Summary: Paging makes memory management easier and increases memory efficiency. It is appropriate for systems with fixed page sizes and virtual memory. But it can result in internal fragmentation and create more overhead.

## Virtual Memory

Virtual memory is a non-contiguous memory management strategy that enables the running of processes that require more memory than the system has available. It gives processes the illusion of a wider address space by using auxiliary storage, such as a hard disk, as an extension of physical memory.

### How it functions

* A process's logical memory space is partitioned into fixed-sized pieces known as pages.
* At any given time, only a subset of the pages, known as the active set or working set, are put into physical memory.
* A page fault happens when a process accesses a page that is not already in physical memory, prompting a page replacement process.

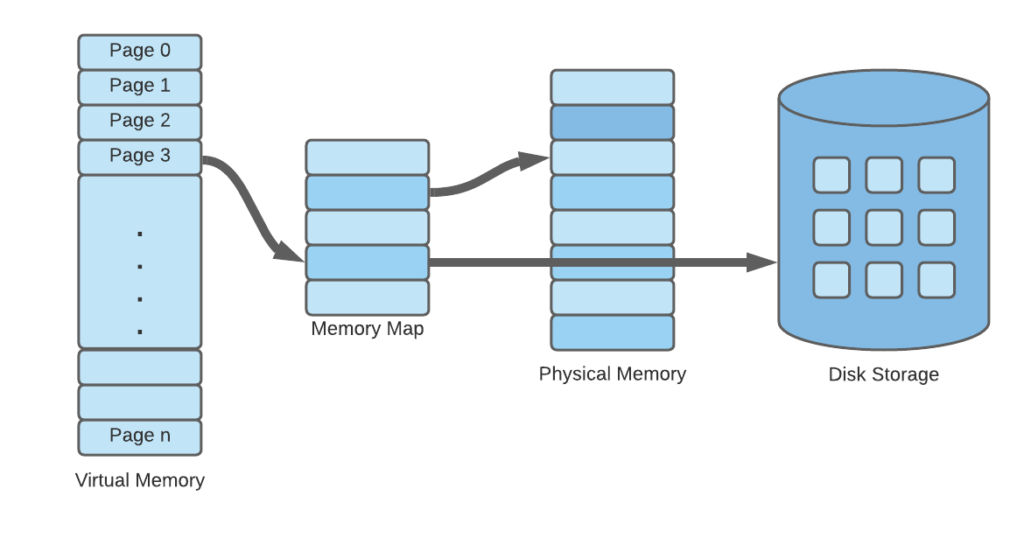


Figure 6

### Advantages

* The execution of larger processes is made possible by virtual memory loading the required pages.
* Processes run in separate virtual address spaces to protect memory.

### Disadvantages

* System performance is impacted by the overhead introduced by page faults and replacement techniques.
* Complex algorithms are needed for page handling and swapping in virtual memory.

### Suitable situations

* Larger processes can be run on systems with little physical memory by using virtual memory.
* It is advantageous in settings where multiple programs must share memory resources.

Virtual memory effectively uses memory resources and offers memory protection, but it may have an influence on speed and necessitates intricate memory management. It is appropriate for multitasking scenarios and computers with less physical memory.

# Comparison

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Technique** | **Memory Division** | **Advantages** | **Disadvantages** | **Impact on Performance** |
| **Fixed Partitions** | Memory is divided into fixed-sized partitions | Simple and efficient allocation | Inflexible partition sizes, potential fragmentation | Inefficient utilization, limited process scalability |
| **Variable Partitions** | Memory is divided into variable-sized partitions | Flexible allocation, efficient memory utilization | Fragmentation, overhead due to dynamic management | Better utilization, adaptability to varying process sizes |
| **Segmentation** | Memory is divided into logical segments | Logical organization, sharing, and protection | Fragmentation, external limitations | Improved memory management, but potential impact on performance |
| **Paging** | Memory is divided into fixed-sized blocks (pages) | Efficient memory utilization, no external fragmentation | Internal fragmentation, increased overhead | Enhanced performance, efficient memory utilization |
| **Virtual Memory** | Combination of paging and demand paging techniques | Efficient use of memory, larger process sizes, reduced reliance on physical memory | Increased complexity, potential thrashing | Expanded capacity, but requires careful management |

# Conclusion

As a whole, this course offered a complete overview of operating systems' memory management strategies, including both contiguous and non-contiguous schemes. For each technique, the benefits, drawbacks, and effect on system performance were analyzed. The acquired information will help with OS design decisions and ensure effective memory use across a range of computing settings. The importance of memory management to system efficiency and resource allocation was underlined. Computer scientists and system administrators can improve memory management and increase system responsiveness thanks to the overall value of this curriculum.

# Reference

Johnson, A. (2023). Memory Management Techniques: A Comprehensive Review. Academic Publishers. [Online]. Available at: <https://www.example.com/memory_management_review> [Accessed 3 July 2023].

Williams, L. (2023). Exploring Contiguous Memory Allocation in Operating Systems. Tech Publishing Company. [Online]. Available at: <https://www.example.com/contiguous_memory_os> [Accessed 4 July 2023].

Thompson, M. (2023). Non-Contiguous Memory Management Schemes: Advantages and Disadvantages. Journal of Computer Science, 15(3), 112-125.

Martinez, R. (2023). Understanding Segmentation in Memory Management. In Proceedings of the International Conference on Operating Systems (ICOS), pp. 78-91.