Memory in Operating System

Team Rinaldi

John Mark A. Gomez (Leader) | John Patrick Gasang (Rapporteur)

3-ECE

CPE355

Sir Ruiz

Table of Contents

Abstract

An operating system, or "OS," is software that controls the execution of application programs and communicates between the computer hardware and other applications. It performs the basic tasks like handling Input and output "I/O", controlling peripherals, process management, file management and memory management.

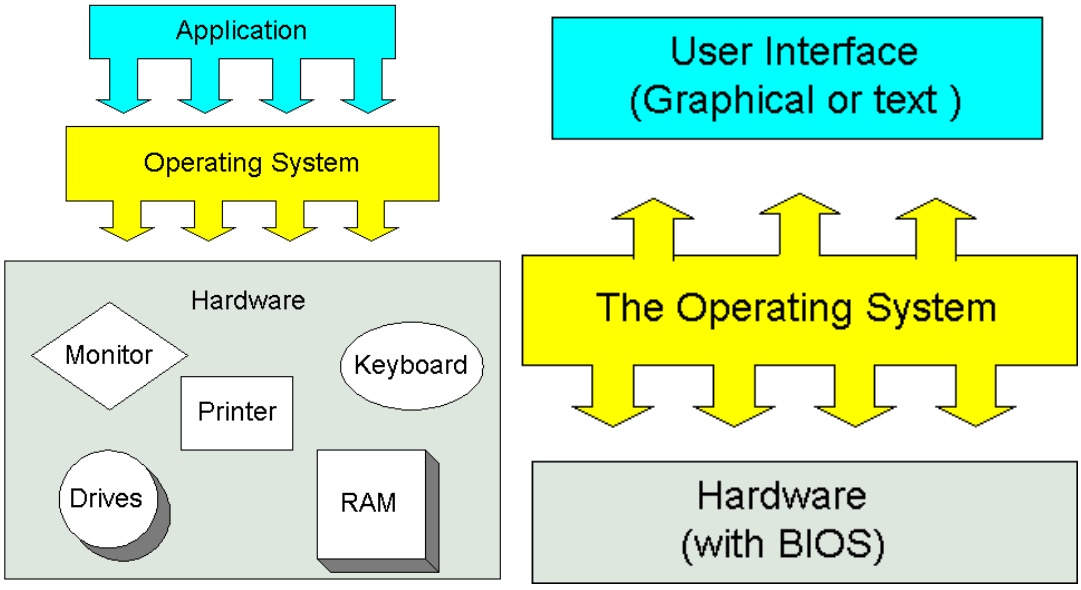
One of the main functions of the Operating System is memory management, which is the process of controlling and coordinating computer memory to optimize the system's overall performance, which can also refer to as the management of Primary Memory or Main memory. Main memory is providing a fast storage that can be accessed by the CPU. Memory management helps the OS in allocating the main memory space to the processes and their data at the time of execution.which utilize the memory space. And it can execute multiple processes at the same time in the OS.

Memory management is a crucial function in an operating system that uses various techniques to help manage the primary memory or main memory. Memory management techniques such as segmentation in which the memory is divided into unequal variable-sized blocks, known as the segment. Paging, similarly to segmentation, the memory is divided but into equal fixed-sized blocks called pages. Lastly, swapping is a temporary exchange of a process between the main memory and secondary storage to make available memory for other processes.

The importance of memory management in an operating system is that it directly affects the execution time of the process, which depends on the availability of data in the main memory. For that reason, memory management must execute to which the essential data is always present in the main memory and must ensure the accuracy, availability, and consistency of the data imported to be effective.

I. Operating System

An Operating System (OS) is the most important system of the computer that interacts with the user and hardware. Some popular or common computer OS include Windows, Mac OS X developed by apple inc., and the open source OS is linux and nowadays we can also see an OS on our device like our smartphones that is operated by android or ios. They have different Graphical User Interface (GUI) but there are 3 main functions of OS 1st is to manage the computer's resources. Examples of these are central processing units (CPU), memory, disk drives, and other computer peripherals such as mouse, keyboard and printers. The 2nd is to establish a user interface, and the 3rd is to execute and provide services for applications software.

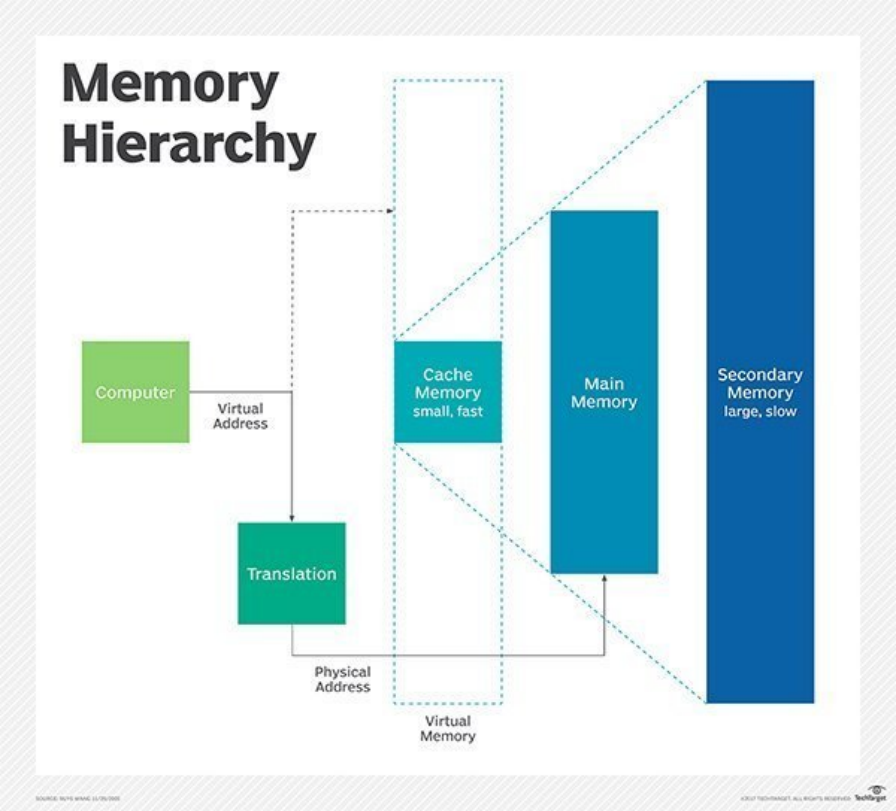


II. Memory in Operating System

Like human memory the functions of memory in Operating systems are used to acquire, store, retain, and later retrieve information. A portion of the OS is in main memory.

The main memory has two partitions which are low and high memory in low memory reside the operating system. Main memory is a physical memory that is the internal memory of the computer. The main memory is also known as Random-Access Memory (RAM). In every program that is running and every file you access it must be copied from storage device into the main memory.

The larger amount of main memory it means you can run or access multiple programs at the same time. The difference between Hard Drive and RAM is RAM is much faster than hard drive even though they are the same storage. RAM is only temporary memory which when you cut the power from the RAM the program and files that have been copied from it will be deleted instantly.



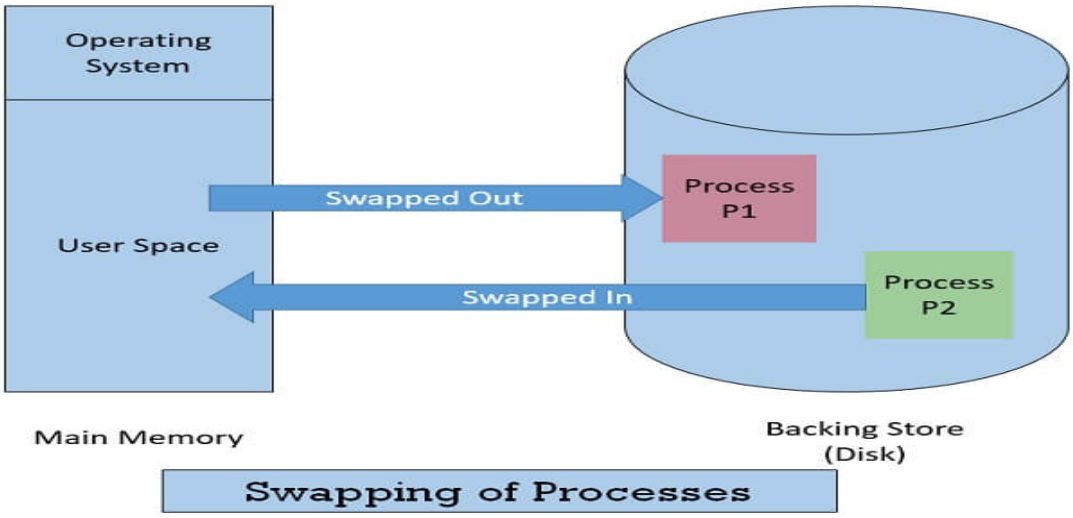
III. Memory Management

To enhance the performance of the system Memory Management is needed. The function of memory management is to keep track of the status of each memory location if it is free or occupied. By memory management it checks how much memory is needed to process. And it sets which process will get memory at what time.

There are different techniques used by memory management. The 1st technique is “single contiguous allocation”, this is the simplest and easiest way. In this method all types of computer memory is available for one application except for the small portion which is reserved for OS. The second technique is “ Partition Allocation”, this technique is dividing the main memory into multiple memory partitions. In every partition stores all the information for specific tasks. In this method when the task is starting there is an allocated partition and it is unloaded when the task is ended. The third technique is “Paged Memory Management” this technique divides the memory into fixed-sized units and it is known as page frames. This method is mapped the pages into frames which should be allocated on a page basis. The last technique is “Segmented Memory Management”, in this technique it does not provide the user’s program with a shared address space. In this method it need hardware support in the form of segment table

IV. Swapping

Computers often had a small main memory to hold all the data they needed. Computer Engineers came up with a solution to overcome this and this technique is called “swapping”. Swapping in memory management is used to improve the main memory utilization by temporarily exchanging any process from the main memory to the secondary memory, which makes the main memory available for other processes. The purpose of swapping is to access the data in the secondary memory and carry it to the main memory and only be used when the data is not already in the main memory by application programs. The performance of the system is usually affected due to the swapping process but, it helps in running large and multiple processes. That is the reason why swapping is called or referred to as memory compaction.



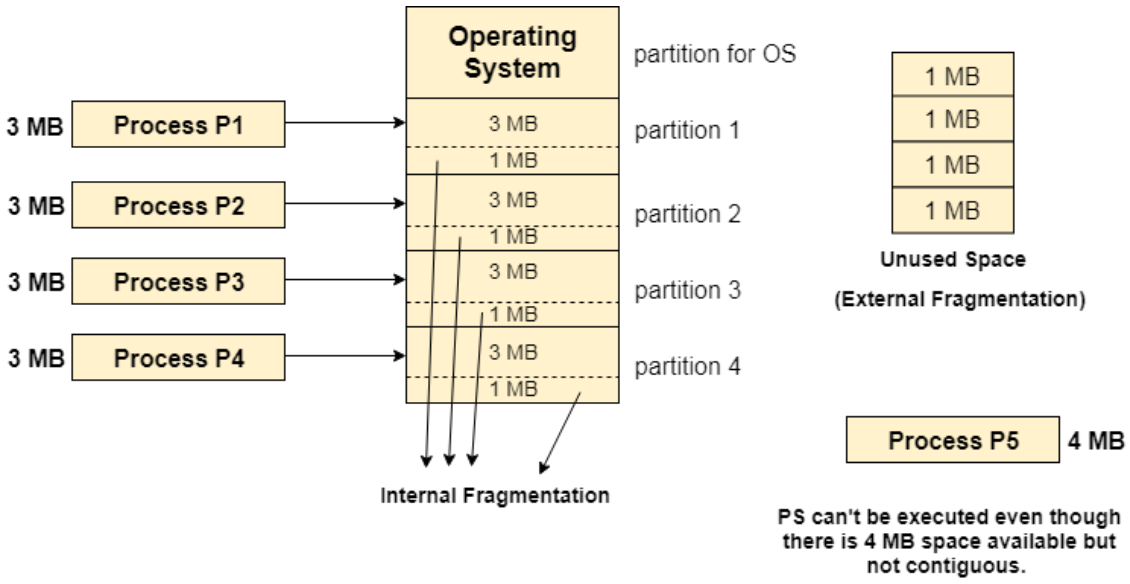
Swapping of Processes

The concept of swapping is divided into two more concepts called Swap-in and Swap-out.

- Swap-out is to remove a process from the main memory and add them to the secondary memory.
- Swap-it is to remove a process from the secondary memory then bringing it back to the main memory.

V. Partitioning

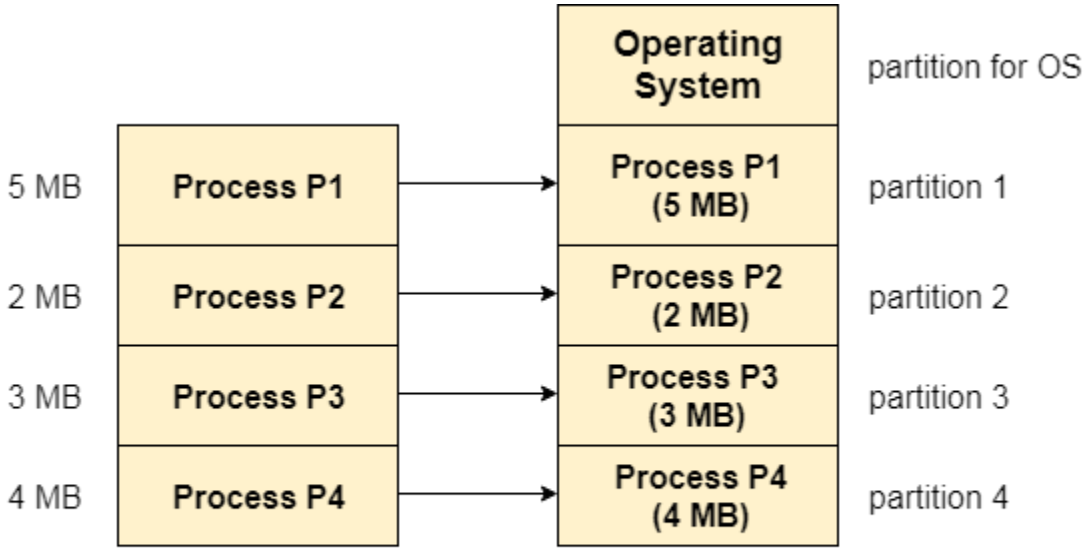
Partitioning in memory management is a contiguous memory allocation technique that is divided into two called Fixed(or Static) Partitioning and Variable (or Dynamic) Partitioning.



Fixed Partitioning

(Contiguous memory allocation)

The oldest and simplest technique is the fixed partitioning in which can be used to load more than one process into the main memory. This technique divides the main memory into partitions of equal or unequal sizes and the first partition is constantly resided by the operating system, while others are for storing user processes to be used. The memory is assigned to the processes in a contiguous way.



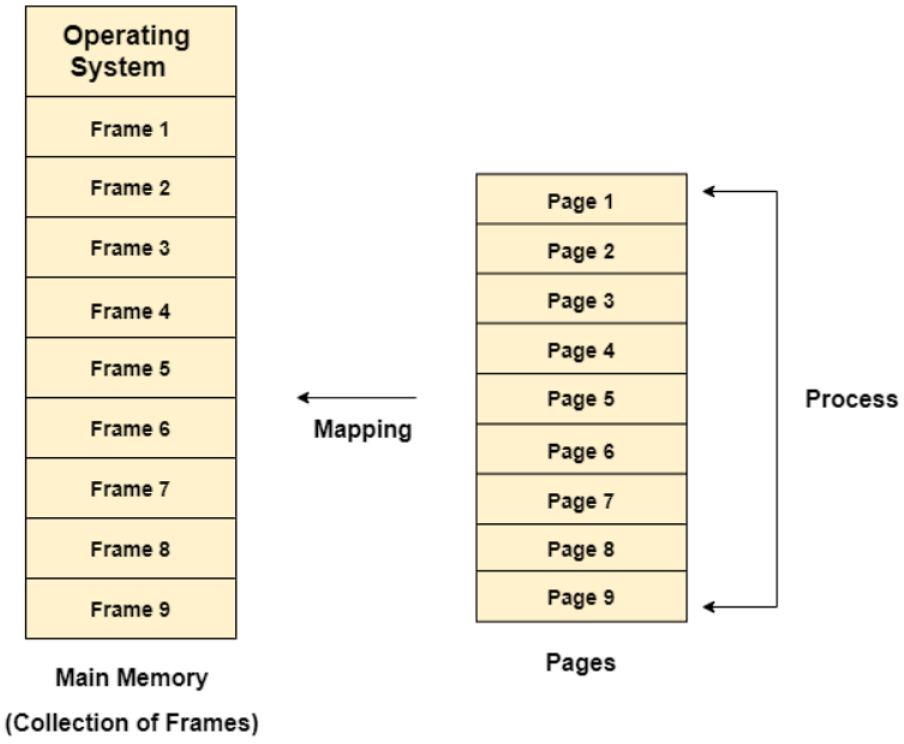
Dynamic Partitioning

(Process Size = Partition Size)

On the other hand, dynamic partitioning tries to overcome the problems caused by the fixed partitioning. In this technique, the partition size is not initially declared and only declared at the time of process loading. The first partition in dynamic partitioning is for the operating system, while the remaining space is divided into parts. Each size of the partition is equal to the size of the process, and the partition size varies according to the need of the process to avoid internal fragmentation.

VI. Paging

Paging in memory management is a technique that allows the operating system to retrieve processes from the secondary memory to the main memory in the form of pages. To avoid external fragmentation and have maximum utilization, this technique divides the main memory into small fixed-size blocks of physical memory called frames, which are similar in size to pages.

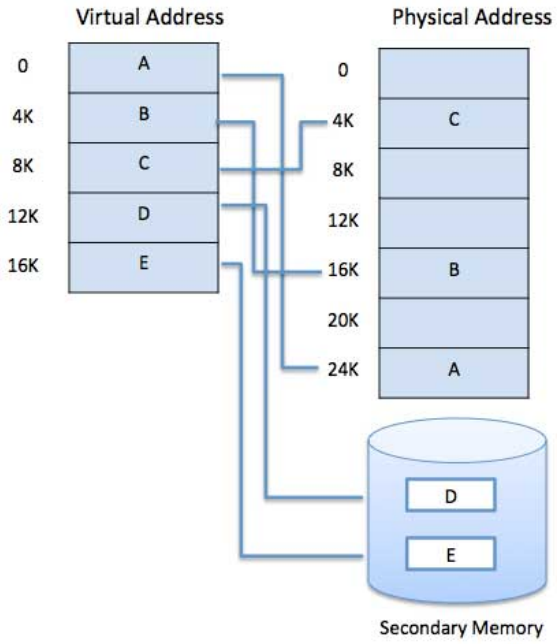


A single page of a process is to be stored in a single frame of the memory, which can be in different locations but, the priority is always to find the contiguous frames. The pages of the process are carried to the main memory only when they are needed. Otherwise, they stay in the secondary storage. The various operating systems define different frame sizes, and each frame must be equal. Considering that the pages are mapped to the frames, the page size must be the same as the frame size.

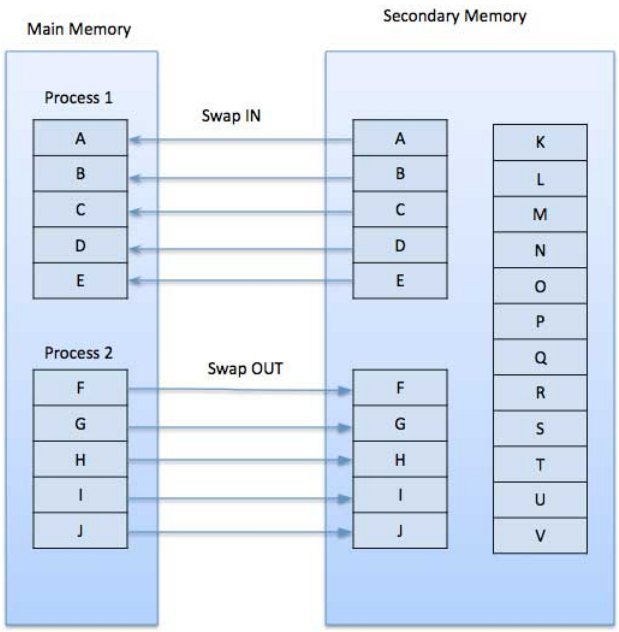
VII. Virtual Memory

The Virtual Memory is a memory management that offers the user to have extra additional main memory where the second memory can be used as a part of main memory. With this method the user can store processes with a larger size than the available memory. Virtual memory can improve the system performance. However if the user relies on virtual memory too much due to virtual data is slower than the main memory. If the Operating System often swapped data between virtual and main memory it can make the computer slow. This is called “thrashing”. Thrashing happens when virtual memory resources are overused.

The example below shows how memory management units translate virtual addresses into physical addresses.



Virtual memory is commonly used by demanding paging, and also can be used in demand segmentation. The demand paging system is like a paging system with swapping where the processes are in secondary memory and pages are loaded on demand and not in advance.

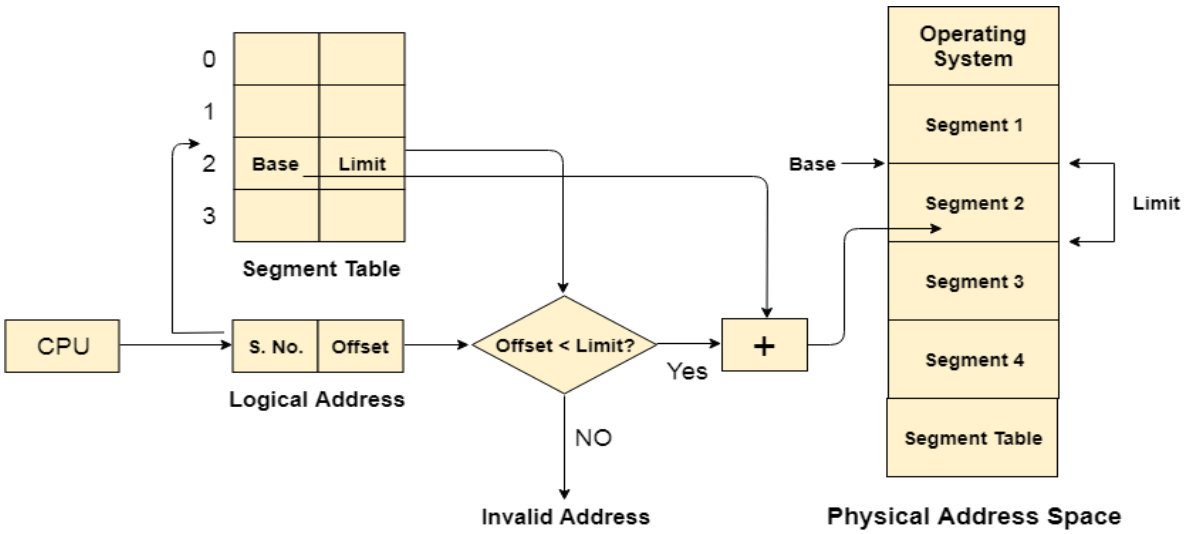


While executing a program, if the program references a page which is not available

VIII. Segmentation

Segmentation in an operating system generally supports the user view of memory and is a technique similar to paging which divides the memory but into variable-sized blocks, called segments. Each of the segment details is put away in a table called the segment table. This segment table is stored in one or many of the segments and mainly contains two pieces of information called the base, which is the address of the segment, and the limit, which is the length of the segment.

In the translation of a logical address to a physical address by the segment table, the CPU generates a local address that contains the segment number and the offset. The segment number is mapped to the segment table, while the limit of the corresponding segment is compared to the offset. On the condition that the offset is lesser than the limit, it is a valid address, or else it will be invalid. If a valid address occurs, the base address of the segment is added to the offset that results in getting the physical address of the actual word in the main memory.



IX. Reference

[1] W. Stallings. *Computer Organization and Architecture*, 10th Ed. Hoboken, NJ: Pearson Education, pp. 276-304, (2016).

[2] “Memory Management”. Tutorials Point. https://www.tutorialspoint.com/operating_system/os_memory_management.htm

[3] “Memory Management in OS: Contiguous, Swapping, Fragmentation”. Guru99. https://www.guru99.com/os-memory-management.html

[4] “Operating System - Overview”. Tutorials Point. https://www.tutorialspoint.com/operating_system/os_overview.htm

[5] “Operating System”. Techterms. https://techterms.com/definition/operating_system

[6] “Operating System”. The University of Rhode Island. https://homepage.cs.uri.edu/faculty/wolfe/book/Readings/Reading07.htm

[7] “Memory management in Operating System”. Include Help. https://www.includehelp.com/operating-systems/memory-management-in-operating-system.aspx

[8] “Swapping in Operating System”. Java T Point. https://www.javatpoint.com/swapping-in-operating-system

[9] “Memory Management Techniques in Operating System”. Include Help. https://www.includehelp.com/operating-systems/memory-management-techniques-in-operating-system.aspx

[10] “RAM vs Hard Drive”. Lehigh University.
https://www.lehigh.edu/~inimr/computer-basics-tutorial/ramvsdiskspacehtm.htm

[11] “Memory Management (Operating System)”. Wikipedia.
https://en.wikipedia.org/wiki/Memory_management_(operating_systems)

[12] “Introduction to Memory Management”. Study Tonight.
https://www.studytonight.com/operating-system/memory-management

[13] “OS Segmentation”. Java T Point. https://www.javatpoint.com/os-segmentation

[14] “OS Fixed Partitioning”. Java T Point.
https://www.javatpoint.com/os-fixed-partitioning

[15] “OS Dynamic Partitioning”. Java T Point.
https://www.javatpoint.com/os-dynamic-partitioning

[16] “OS Paging”. Java T Point.
https://www.javatpoint.com/os-paging-with-example