International Journal of Scientific Research and Engineering Development-– Volume 8 Issue 5, Sep-Oct 2025
Available at www.ijsred.com

REVIEW PAPER

Review on Memory Management in Operating System

Prof. Nisha Pawar*, Harsh Shingare**, Tushar Gawai***, Saurav Shirsath****
*(Lecturer at P.E.S Polytechnic, Chh. Sambhji Nagar, Maharashtra, India
Email: pawarnisha1797@gmail.com)
**(Computer Engineering, P.E.S Polytechnic, Chh. Sambhaji Nagar
Email: harshingare235@gmail.com)
*** (Computer Engineering, P.E.S Polytechnic, Chh. Sambhaji Nagar
Email: tushargawai17306@gmail.com)
****(Computer Engineering, P.E.S Polytechnic, Chh. Sambhaji Nagar
Email: sshirsath594@gmail.com)

----------**********----------

Abstract:

In today's computers, it's necessary to have efficient memory management, which means how the computer handles its main memory. This is important for applications and the operating system to run smoothly, especially if a program needs to stay running for a long time without shutting down. The computer's resources are to be used wisely to make everything perform better. This paper explains the methods an operating system uses to manage memory. Specifically, it will show the basic design of segmentation—a memory management technique. We'll demonstrate how memory is allocated using segmentation, the fundamental concepts of virtual memory.

Keywords **— Memory Management, Operating System, Segmentation, Virtual Memory.**

----------**********----------

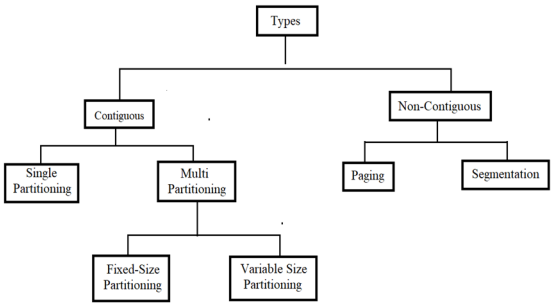
I. INTRODUCTION

An operating system is a required type of software that manages the computer's resources, like its hardware and programs, and provides the basic services that all software needs. Most computer programs cannot work without an operating system. The main goal of any computer is to run programs. To be run, these programs, along with the data they use, must be placed in the main memory (RAM). Managing this main memory is one of the most important and difficult jobs for the operating system, especially in a multiprogramming system where many programs share the memory. Memory is structured like a very long list of words or bytes, and each one has its own unique address. Modern operating systems are designed to manage this memory efficiently. However, researchers are still looking for ways to improve how memory is given out or allocated to applications, as the methods used for memory allocation.

II. LITERATURE SURVEY

The Literature Survey explains the main research and dependable information that backs up this paper. To start, we looked at trusted, classic books that teach us the basic rules for how the Operating System (OS) manages memory. These books explain two main things: how programs share the computer's RAM when you're multitasking, and the two methods for storing programs—contiguous allocation (putting the whole program in one continuous piece) and non-contiguous allocation (scattering it in different available spots). After reviewing these basics, we also checked other studies and research papers. These papers give us real-world evidence supporting the advanced methods we discuss, like segmentation and virtual memory. using all the important information, we can clearly and simply explain each key concept of memory management in this paper.

III. TYPES OF MEMORY MANAGEMENT



1.Contiguous Memory Allocation:

Contiguous allocation means that every part of a program is a logical object that is placed in memory locations that are all strictly next to each other in continuous addresses. This method is used in very simple systems and with programs that always stay in memory as resident monitors. It is also used in multiprogramming, which means running many programs at once, which is further divided into two types: fixed partition and variable partition.

2. Non-Contiguous Memory Allocation:

Non-contiguous allocation means that a single program can be spread out in memory locations that are not next to each other. This approach was developed because it is a better way to solve the fragmentation problem, which means the wasted memory space problem. It makes the program's logical address spaces non-continuous. The two main methods used for non-contiguous memory management are paging and segmentation.

3.Paging:

Paging is a memory management method that removes the requirement that physical memory be given out in one continuous block. This scheme allows the physical address space, meaning the space where the program actually sits in RAM of a running process, to be non-contiguous and scattered. Because of this advantage over older methods, paging has been used in most operating systems. In contrast to segmentation, paging supports the system view of memory by dividing the physical memory into fixed-size blocks called frames.

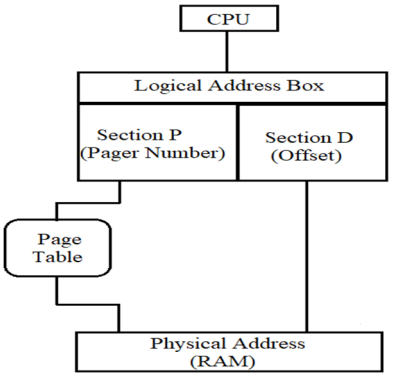
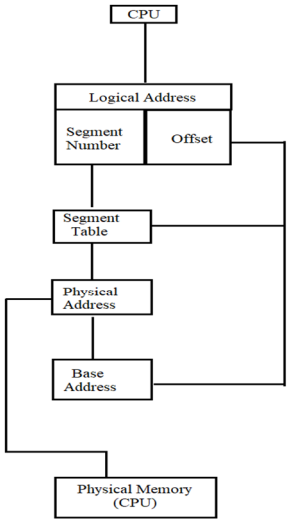


Fig. Paging Diagarm

4.Segmentation:

Segmentation is another method for putting a program into memory locations that are not next to each other, which means non-contiguous. It is different from paging because segments are logical parts of a program, like a function or a data block, so they are of variable size. It is a scheme that supports the user's view of memory, meaning how the programmer sees the parts of their program rather than the system's technical view. The system divides the logical address space into these different segments. The size of a segment changes based on the data it holds or the type of operation performed on it.



Segmentation Diagram

5. Virtual Memory:

Virtual memory is a smart trick a computer uses to make it seem like it has much more main memory (RAM) than it physically does. This illusion is extremely helpful because a program can run without worrying about the actual size of the computer's

RAM. It makes the program independent of the physical memory size. Many different programs can share the computer system at the same time without interfering with each other's memory needs. The complex way virtual memory is managed is hidden from the programs (transparent). The programs don't need to know the detailed process. It's an advantage for the operating system because a program can start running even if only part of it is loaded into the main memory. It also helps to reduce external fragmentation (wasted, unused memory gaps) without changing the order in which programs are scheduled.

6. Hardware Implementation:

(How the Computer's Parts Manage Memory)

In computer systems, why use segmentation? The addresses that the CPU uses are made of two parts: the segment ID, which means which part of the program it is, and an offset, which means the location within that part.

CONCLUSION

This paper helped us understand the importance of memory management in operating systems. We looked closely at segmentation, which is a method that helps organize memory logically, and reviewed the fundamental concepts of how memory is handled. We write about the different ways programs can be placed in memory, comparing contiguous, which means all in one piece, and non-contiguous, which means scattered allocation. We also learned about virtual memory, which is the key technique that makes computers more powerful and allows for smooth multitasking. Ultimately, we have seen that using these different memory allocation techniques is important for making a computer efficient and reliable.

ACKNOWLEDGMENT

We would like to express sincere and heartfelt gratitude to Prof. Nisha Pawar of the Department of Computer Engineering, P.E.S. Polytechnic, Chh. Sambhaji Nagar, Maharashtra. For supporting and guidance regarding resources that helped in the successful completion of the review paper

REFERENCES

[1]Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts* (10th ed.). Wiley. Link to bookhttps://os.ecci.ucr.ac.cr/slides/Abraham-Silberschatz-Operating-System-Concepts-10th-2018.pdf

[2]Muhammad Abdullah Awais: MS150200157, Virtual University of Pakistan. https://scholar.google.com/scholar?q="Muhammad+Abdullah+Awais"+Virtual+University+of+Pakistan

[3]Martin Bohmert: Department of Computer Science, University of Freiburg, Germany. https://scholar.google.com/scholar?q="Martin+Bohmert"+"University+of+Freiburg"

[4]Christoph Scholl: Department of Computer Science, University of Freiburg, Germany. https://scholar.google.com/scholar?q

