iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** III **Month of publication:** March 2021

DOI: https://doi.org/10.22214/ijraset.2021.33259

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue III Mar 2021- Available at www.ijraset.com

A Novel Approach on Memory Management Systems

V. Kalpana[1], M. Sushma Sri[2]
[1, 2]*Government Institute Of Electronics*

Abstract: ***In today's generations Memory Management plays a vital role in Operating Systems. The main efficient usage of memory in computer applications are most important aspect because the memory is an essential part of a computer system and manages maximum utilization in a computer system. In this paper the author presents a Memory Management technique to achieve better performance in the computer system.***

Keywords: ***Memory Management Strategies, Single Contiguous Allocation, Swapping, Fragmentation, Paging.***

I. INTRODUCTION

Operating System is a system software used in a computer which acts an interface between computer and the user for managing all resources efficiently. The most importance of Operating System to make easier to use the computer. In this regard Memory Management is the main function of an operating system that manages primary memory i.e., main memory and helps to processes the memory management in between the main memory and disk in an Operating System. This Memory Management keeps a track of each memory location, irrespective of whether it is allocated to some process or it remains idle in Operating System and it also allows to check the quantity of memory allocation. The main important aspect of Memory Management is providing protection in between processing system, stores the data most efficiently and also allocates the memory space for all applications.

II. MEMORY MANAGEMENT STRATEGIES

Memory Management Strategies are two types, one is Static Memory allocation strategy and Dynamic Memory allocation strategy. In Static Memory allocation strategy, the memory size is determined when the system is compiled and usage of memory will be more whereas Dynamic Memory allocation strategy [17], the memory is required only at that time of task and the memory is used correctly. The important aspect in static memory allocation strategy is that the testing any program will be easier and execution is also more efficient but the main drawback it requires or uses more memory. And this advantage is overcome by dynamic memory allocation strategy.

III.MEMORY MANAGEMENT TECHNIQUES

Memory management in operating system helps in allocating the main memory space to processes the data at the time of execution [15]. There are different Memory Management techniques involved in managing the memory in an Operating System as shown in Fig.1.

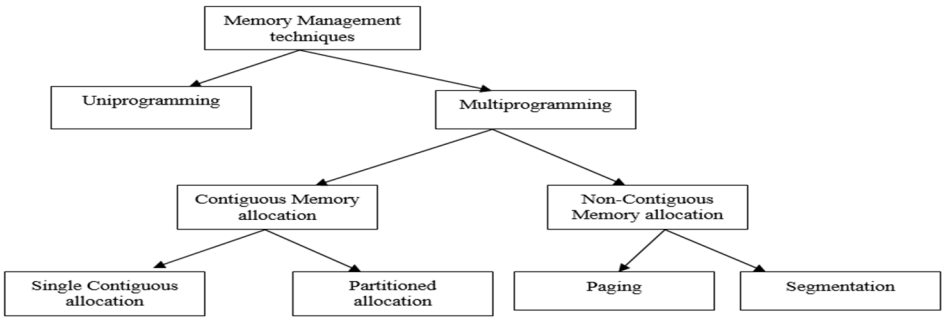


Fig.1 Memory Management System

Memory Management Techniques [22] is classified as Uniprogramming and Multiprogramming where the Uniprogramming technique, the RAM is isolated into two categories where is one category is for leaving the operating system and the other category is for the client process. In the multi-programming, the various clients can share the memory at the same time. Multiprogramming can be divided into contiguous and Non-contiguous memory allocation.

A. Contiguous Memory Allocation

In the contiguous memory allocation, both operating system and the client must in the primary memory and when any client demands the memory of a segment of the contiguous memory block will be allotted to that procedure [6]. Contiguous Memory allocation can be divided into single contiguous allocation and partitioned allocation.

1) *Single Contiguous Allocation:* Single Contiguous Allocation (SCA) is a simple technique for memory allocation and its doesnot require any special hardware features. The best example is MS-DOS operating system which allocates memory in same manner as SCA. In this technique the entire memory is allocated to a single job and the main memory is divided into three contiguous regions which is shown in Fig.1.

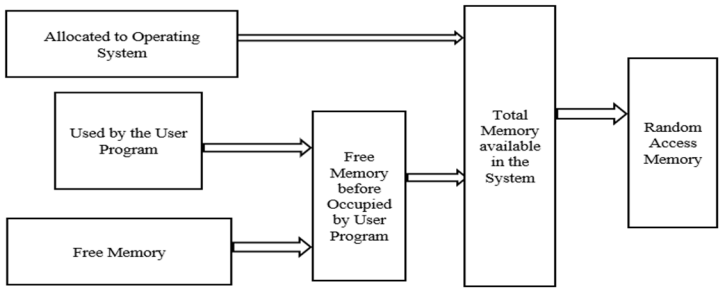


Fig.2 Single Contiguous Allocation Technique

2) *Partitioned Allocation:* In Partitioned Allocation technique as its name suggests the primary memory is divided into various memory partitions which is mostly contiguous areas of memory and every partition store all the information for a specified job in which allocation and deallocation of memory exists. In partition allocation more than one partition is available to place a process's request, in this case a partition must be selected. To choose a particular partition a partition method is needed. A partition allocation method is considered better if it avoids internal fragmentation.

B. Non- Contiguous Memory Allocation

In the non-contiguous memory allocation, the accessible free memory space is dispersed to a great extent for usage of processing system and all that free memory space is not in one location. This Non-Contiguous Memory Allocation can be divided into Paging and Segmentation.

1) *Paging:* Paging is a technique in which the main memory of computer system is organized in the form of equal sized blocks called pages. In this technique, the address of occupied pages of physical memory are stored in a table, which is known as page table. Paging enables the operating system to obtain data from the physical memory location without specifying lengthy memory address in the instruction. In this technique, the virtual address is used to map the physical address of the data. The length of virtual address is specified in the instruction and is smaller than physical address of the data. It consists of two different numbers; first number is the address of page called virtual page in the page table and the second number is the offset value of the actual data in the page in Fig.3.

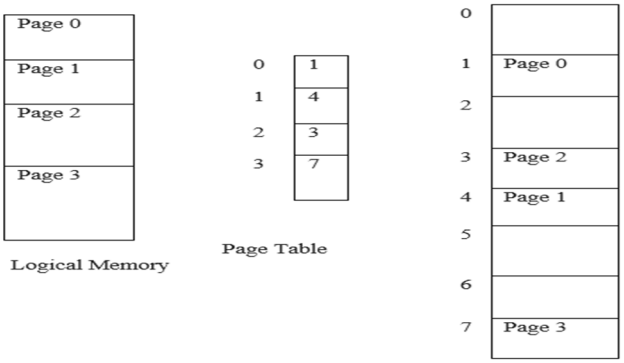


Fig.3 Paging Technique

2) *Segmentation:* In Operating Systems, Segmentation is a memory management technique in which, the memory is divided into the variable size parts. Each part is called as segment which can be allocated to a process and stored in a table called as segment table is seen in Fig.4. This segment table is stored in one or many of the segments which contains Base and limit. The Base is a base address of the segment and Limit is the length of the segment. From the main memory the logical address is generated which contains two parts namely, Segment Number and Offset. The Segment number is mapped to the segment table and the limit of the respective segment is compared with the offset. If the offset is less than the limit then the address is valid otherwise it throws an error. In the case of valid address, the base address of the segment is added to the offset to get the physical address of actual word in the main memory.

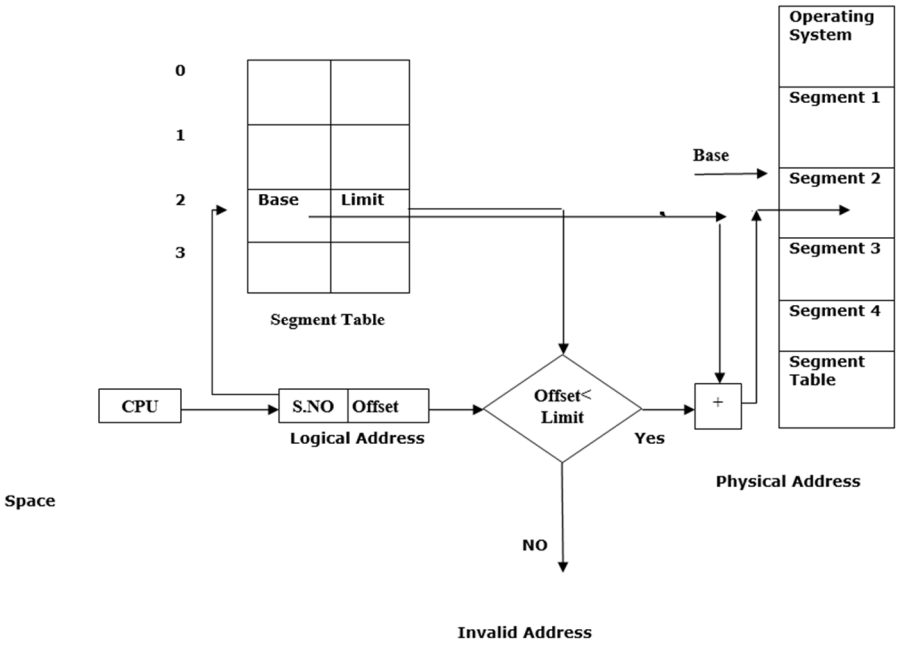


Fig.4 Segmentation Technique

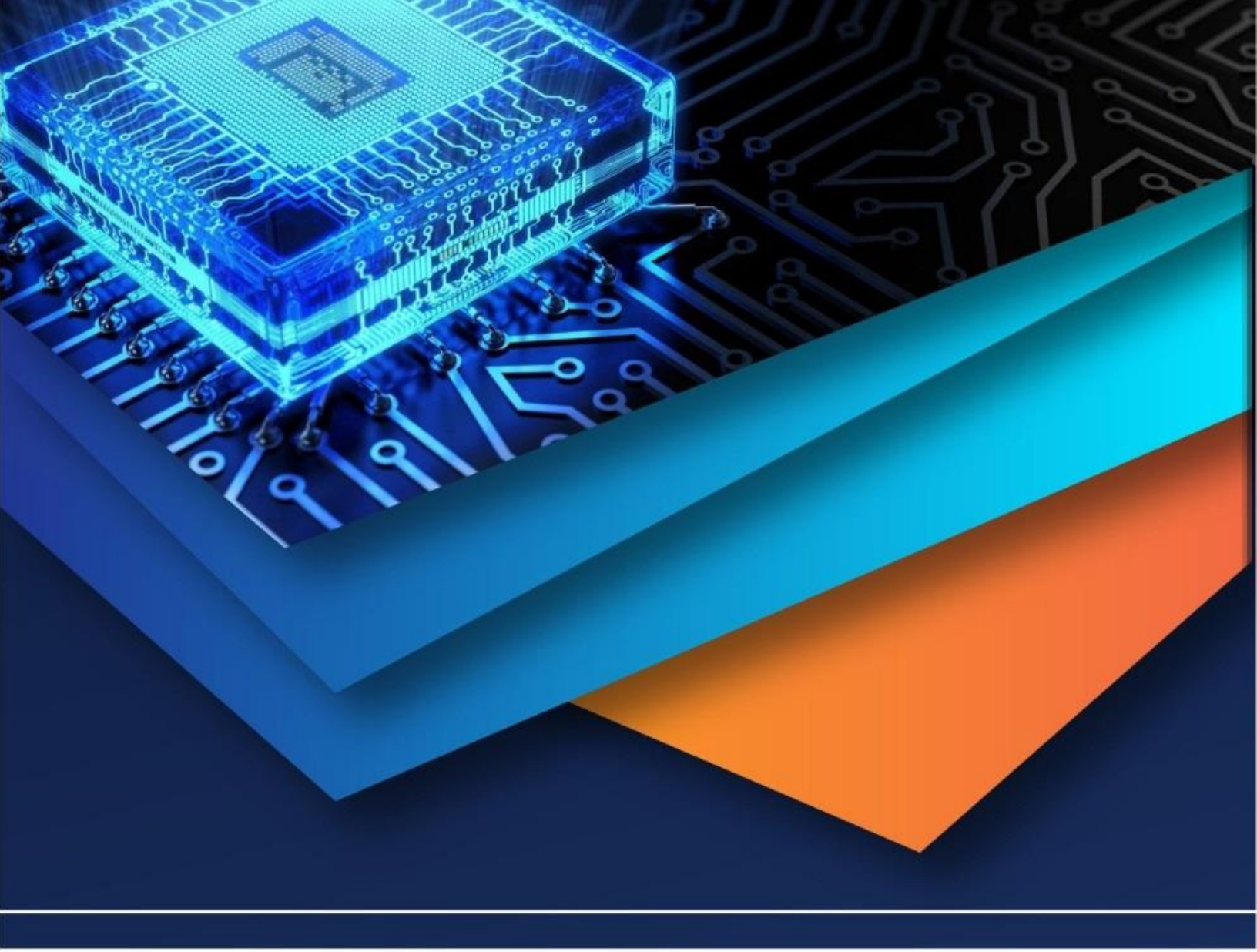
IV. CONCLUSION

Memory Management is an efficient technology to implement in all computer applications to utilize maximum resources. In this paper the author presents a Memory Management various technique to achieve overall performance in the computer system. The various techniques are used in managing memory in order to process the data.

REFERENCES

[1] M. Tofte and J.-P. Talpin, "Region-based memory management," Information and computation, vol. 132, pp. 109-176, 1997.
[2] C. A. Waldspurger, "Memory resource management in VMware ESX server," ACM SIGOPS Operating Systems Review, vol. 36, pp. 181-194, 2002.
[3] J. C. Lau, S. C. Roy, D. L. Callaerts, and I. E. N. Vandeweerd, "Method and apparatus for allocation and management of shared memory with data in memory stored as multiple linked lists," ed: Google Patents, 1999.
[4] P. R. Wilson, M. S. Johnstone, M. Neely, and D. Boles, "Dynamic storage allocation: A survey and critical review," in Memory Management, ed: Springer, 1995, pp. 1-116.
[5] K. Moronaga and M. Watanabe, "Storage management system for memory card using memory allocation table," ed: Google Patents, 1993.
[6] D. F. Hooper, G. Wolrich, M. J. Adiletta, and W. R. Wheeler, "Method for memory allocation and management using push/pop apparatus," ed: Google Patents, 2003.
[7] D. Gay and A. Aiken, Memory management with explicit regions vol. 33: ACM, 1998.
[8] C. A. Waldspurger and W. E. Weihl, "Lottery scheduling: Flexible proportional-share resource management," in Proceedings of the 1st USENIX conference on Operating Systems Design and Implementation, 1994, p. 1.
[9] T. J. Lehman and M. J. Carey, "A study of index structures for main memory database management systems," in Proc. VLDB, 1986.
[10] E. D. Berger, B. G. Zorn, and K. S. McKinley, "OOPSLA 2002: reconsidering custom memory allocation," ACM SIGPLAN Notices, vol. 48, pp. 46-57, 2013.
[11] O. Avissar, R. Barua, and D. Stewart, "An optimal memory allocation scheme for scratch-pad-based embedded systems," ACM Transactions on Embedded Computing Systems (TECS), vol. 1, pp. 6-26, 2002.
[12] R. Motwani, J. Widom, A. Arasu, B. Babcock, S. Babu, M. Datar, et al., "Query processing, resource management, and approximation in a data stream management system," 2003.

[13] F. Catthoor, S. Wuytack, G. de Greef, F. Banica, L. Nachtergaele, and A. Vandecappelle, Custom memory management methodology: Exploration of memory organisation for embedded multimedia system design: Springer Science & Business Media, 2013.
[14] I. Foster, C. Kesselman, C. Lee, B. Lindell, K. Nahrstedt, and A. Roy, "A distributed resource management architecture that supports advance reservations and co-allocation," in Quality of Service, 1999. IWQoS'99. 1999 Seventh International Workshop on, 1999, pp. 27-36.
[15] I. Puaut, "Real-time performance of dynamic memory allocation algorithms," in Real-Time Systems, 2002. Proceedings. 14th Euromicro Conference on, 2002, pp. 41-49.
[16] P. Zhou, V. Pandey, J. Sundaresan, A. Raghuraman, Y. Zhou, and S. Kumar, "Dynamic tracking of page miss ratio curve for memory management," in ACM SIGOPS Operating Systems Review, 2004, pp. 177-188.
[17] D. J. McMahon and G. A. Buzsaki, "Dynamic memory allocation in a computer using a bit map index," ed: Google Patents, 1998.
[18] K. Harty and D. R. Cheriton, Application-controlled physical memory using external page-cache management vol. 27: ACM, 1992.
[19] A. Aiken, M. Fähndrich, and R. Levien, Better static memory management: Improving region-based analysis of higher-order languages vol. 30: ACM, 1995.
[20] M. B. Jacobson, J. W. Fordemwalt, D. L. Voigt, M. D. Nelson, H. Vazire, and R. Baird, "Memory systems with data storage redundancy management," ed: Google Patents, 1995.
[21] John P. Hayes (1998) Computer Architecture and Organization, WCB/McGraw- Hill Books, McGraw-Hill International Editions.
[22] Denning P.J. and Graham G.S, “Multi-programmed Memory Management”, Proc. IEEE, vol 63, June 1975, pp 924-939
[23] William Stalling (2004) Computer Architecture and Organization, WCB/McGraw- Hill Books, McGraw-Hill International Editions.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY