International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 10, pp: 86-91 www.ijaem.net ISSN: 2395-5252

Analysis of Allocation Algorithms in fixed partitioning

Pooja Shri. M. S, Sathya Sri. A, Yaamini Samyugtha. D, Dr.M. Sujithra M.C.A, M.Phil., PhD,
Dr.A.D.Chitra M.C.A, M.Phil., PhD,
2nd year, MSc Software Systems (Integrated), Coimbatore Institute Of Technology, Coimbatore.
Assistant Professor,Department of Data Science, Coimbatore Institute of Technology, Coimbatore.
Assistant Professor, Department of Software Systems, Coimbatore Institute of Technology, Coimbatore.

Date of Submission: 20-11-2020 Date of Acceptance: 03-12-2020

ABSTRACT: Memory management is that the process of controlling and coordinating memory, assigning portions called blocks to varied running programs to optimize overall system performance and also referred to as memory allocation. Placement algorithms are implemented to work out the slot which will be allocated process amongst the available ones within the partitioned memory. Memory slots allocated to processes could be too big when using the prevailing placement algorithms hence losing tons of space thanks to internal fragmentation. In dynamic partitioning, external fragmentation occurs when there's a sufficient amount of space within the memory to satisfy the memory request of a process but the process's memory request can't be satisfied because the memory available is during a non-contiguous manner. This paper describes the way to resolve external fragmentation using three allocation algorithms. These algorithms are First-fit, Best-fit and Worst-fit. we'll present the implementation of three algorithms and compare their performance on generated virtual trace.

Keywords: Contiguous Memory Management, First-fit,Best-fit, Worst fit, External Fragmentation, Internal fragmentation, Fixed partitioning, Buddy system

I. INTRODUCTION

Contiguous memory allocation may be a memory allocation technique. It allows to store the method only during a contiguous fashion. Thus, entire process has got to be stored as one entity at one place inside the memory. OS occupies some fixed portion of main memory, and therefore the remainder of main memory is out there to be used by multiple processes. the only scheme for managing this available memory is to partition it into regions with fixed boundaries. Static partitioning may be a fixed size partitioning scheme. during this technique, main memory is pre-divided into fixed size partitions. the dimensions of every partition is fixed and can't be changed. Each partition is allowed to store just one process.

TYPES OF PARTITIONING

a. Contiguous Memory Allocation

Contiguous memory allocation may be a memory allocation technique.It allows to store the method only during a contiguous fashion.Thus, entire process has got to be stored as one entity at one place inside the memory.

Types:

1.Fixed Partitioning
2.DynamicPartitioning

1. Fixed partitioning:

Defined as the system of dividing memory into non-overlapping sizes that are fixed, unmoveable, static. A process may be loaded into a partition of equal or greater size and is confined to its allocated partition

2. Dynamic partitioning:

Dynamic partitioning may be a variable size partitioning scheme.It performs the allocation dynamically.When a process arrives, a partition of size adequate to the dimensions of process is made .Then, that partition is allocated to the method .

b.Non-Contiguous Memory Allocation

Non-contiguous memory allocation may be a memory allocation technique.It allows to store parts of one process during a non-contiguous fashion.Thus, different parts of an equivalent process are often stored at different places within the main memory. Its types:

1. Simplepaging
2. Simplesegmentation

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 10, pp: 86-91 www.ijaem.net ISSN: 2395-5252

1.Simple Paging:

Paging may be a fixed size partitioning scheme.In paging, secondary memory and main memory are divided into equal fixed size partitions.The partitions of secondary memory are called as pages.The partitions of main memory are called as frames.Each process is split into parts where size of every part is same as page size.The size of the last part could also be but the page size.The pages of process are stored within the frames of main memory depending upon their availability.

2.Simple Segmentation:

Segmented paging may be a scheme that implements the mixture of segmentation and paging.Paging and Segmentation are the non-contiguous memory allocation techniques.Paging divides the method into equal size partitions called as pages.Segmentation divides the method into unequal size partitions called as segments.

FIXED PARTITIONING

If a process is chosen to allocate memory, then it goes into memory and competes for the processor. the amount of fixed partition gives the degree of multiprogramming. Since each queue has its own memory region, there's no competition between queues for the memory.In this method, memory is split into partitions whose sizes are fixed. OS is placed into rock bottom bytes of memory. Processes are classified on entry to the system consistent with their memory they requirements. we'd like one Process Queue (PQ) for every class of process.

Types of fixed Partitioning Process

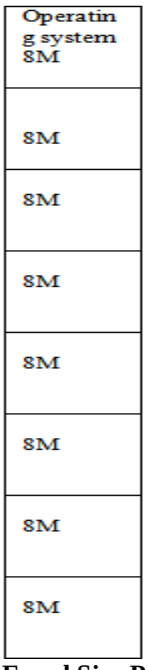
a. Equal size
b. Unequal size

a.Equal size

Any process whose size is a smaller amount than or adequate to the partition size are often loaded into an available partition. If all partitions are full, the OS can swap a process out of a partition.A program might not slot in a partition. The programmer must design the program with overlaysMain memory use is inefficient. Any program, regardless of how small, occupies a whole partition. this is often called internal fragmentation.

b.Unequal size

Using unequal size partitions helps lessen the issues programs up to 16M are often accommodated without overlays partitions smaller than 8M allow smaller programs to be accommodated with less internal fragmentation. Small jobs won't utilize partition space Efficiently Small jobs won't utilize partition space efficiently the memory that's external to all or any partitions becomes increasingly fragmented. it's called external fragmentation.



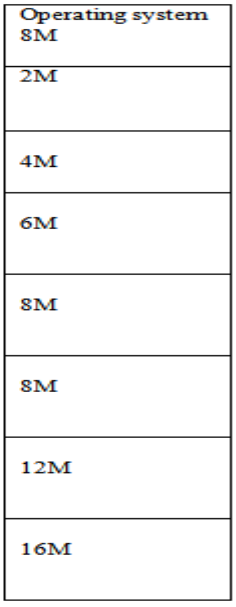


Fig .1. Equal Size Partitioning, Fig.2. Unequal Size Partitionin

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 10, pp: 86-91 www.ijaem.net ISSN: 2395-5252

Placement Algorithm For Partitions

a.Equal-size partitions because all partitions are of equal size, it does not matter which partition is used.

b.Unequal-size partitions can assign each process to the smallest partition within which it will fit queue for each partition processes are assigned in such a way as to minimize wasted memory within a partition.

Fixed Algorithms for Partition Allocation

Popular algorithms used for allocating the partitions to the arriving processes are-

1. First FitAlgorithm
2. Best FitAlgorithm
3. Worst FitAlgorithm

1. First FitAlgorithm-

This algorithm starts scanning the partitions serially from the starting.When an empty partition that's large enough to store the method is found, it's allocated to the method. Obviously, the partition size has got to be greater than or a minimum of adequate to the method size.

For example: Given ten memory partitions of 500KB, 200KB, 800KB, 400KB, 100KB, 700KB, 300KB, 600KB, 1000KB, 900KB (in order), How would the First-fit, Best-fit, Worst- fitalgorithms place processes of 212KB, 150KB, 375KB, 950KB, 350KB, 632KB, 400KB, 717KB, 811KB (in order)? Which algorithm makes the most efficient use of memory?

In First-fit:

212KB is put in 500KB partition
150KB is put in 288KB (new partition 288KB=500KB-212KB)
375KB is put in 800KB partition
950KB is put in 1000KB partition
350KB is put in 425KB (new partition 425KB=800KB-375KB
632KB is put in 700KB partition
400KB is put in 400KB partition
717KB is put in 900KB partition
811 must wait

2.Best FitAlgorithm-

This algorithm first scans all the empty partitions.It then allocates the littlest size partition to the method. Best Fit Algorithm works best. this is often because space left after the allocation inside the partition is of very small size. Thus, internal fragmentation is least.

For example: Given ten memory partitions of 500KB, 200KB, 800KB, 400KB, 100KB, 700KB, 300KB, 600KB, 1000KB, 900KB (in order), How would the First-fit, Best-fit, Worst- fitalgorithms place processes of 212KB, 150KB, 375KB, 950KB, 350KB, 632KB, 400KB, 717KB, 811KB (in order)? Which algorithm makes the most efficient use of memory?

In Best-fit:

212KB is put in 300KB partition
150KB is put in 200KB partition
375KB is put in 400KB partition
950KB is put in 1000KB partition
350KB is put in 500KB partition
632KB is put in 700KB partition
400KB is put in 600KB partition
717KB is put in 800KB partition
811KB is put in 900KB partition

3. Worst FitAlgorithm-

This algorithm first scans all the empty partitions.It then allocates the littlest size partition to the method.Worst Fit Algorithm works worst.This is because space left after the allocation inside the partition is of very large size. Thus, internal fragmentation is maximum.

For example: Given ten memory partitions of 500KB, 200KB, 800KB, 400KB, 100KB, 700KB, 300KB, 600KB, 1000KB, 900KB (in order), How would the First-fit, Best-fit, Worst- fitalgorithms place processes of 212KB, 150KB, 375KB, 950KB, 350KB, 632KB, 400KB, 717KB, 811KB (in order)? Which algorithm makes the most efficient use of memory?

In Worst-fit:

212KB is put in 1000KB partition
150KB is put in 900KB partition
375KB is put in 800KB partition
950KB must wait
350KB is put in 788KB (new partition 788KB=1000KB-212KB)
632KB is put in 750KB (new partition 750KB=900KB-150KB)
400KB is put in 700KB partition
717KB must wait
811KB must wait

ADVANTAGES OF FIXED PARTITIONING

It is simple and straightforward to implement.It supports multiprogramming since multiple processes are often stored inside the most memory.Only one access is required which reduces the time interval.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 10, pp: 86-91 www.ijaem.net ISSN: 2395-5252

Disadvantages Of Fixed Partitioning

It suffers from both internal fragmentation and external fragmentation. It utilizes memory inefficiently. The degree of multiprogramming is restricted adequate to number of partitions. there's a limitation on the dimensions of process since processes with size greater than the dimensions of largest partition can't be stored and executed. limited adequate to number of partitions. there's a limitation on the dimensions of process since processes with size greater than the dimensions of largest partition can't be stored and executed

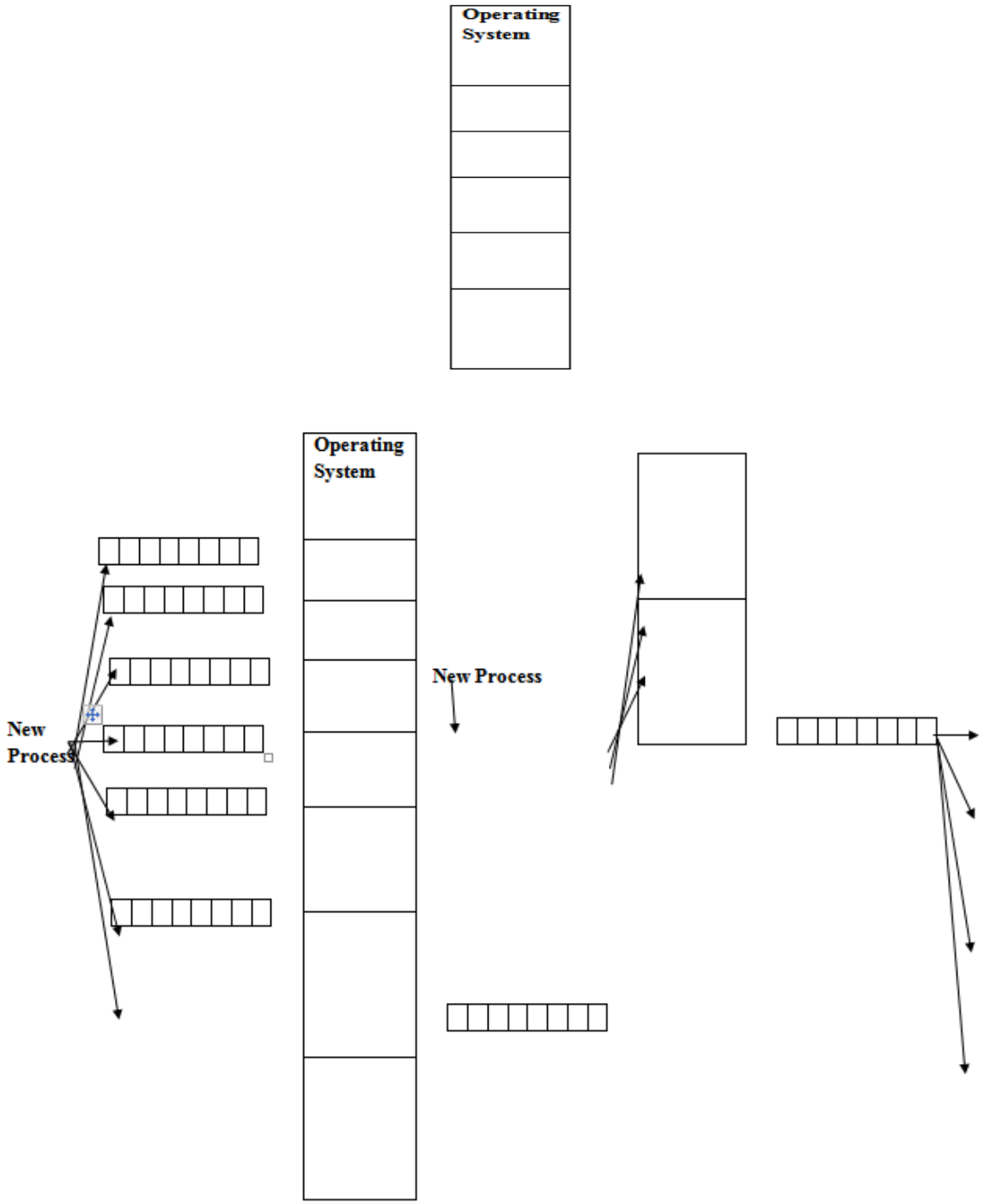


Fig. 3. One Process Queue per Partition Fig. 4. Single process queues

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 10, pp: 86-91 www.ijaem.net ISSN: 2395-5252

MEMORY ASSIGNMENT FOR FIXED PARTITIONING

PRACTICAL EXAMPLE- BUDDY SYSTEM

Entire space available is treated as a single block of 2UIf a request of size s such that 2U- 1 < s <= 2U, entire block is allocated.Otherwise block is split into two equal buddies.Process continues until smallest blockgreater than or equal to s is generated.

Problem

A 1MB block of memory is allocated using the buddy system.

i. Show the results of the following sequence in a figure: Request 70; Request 35; Request 80; Return A;

Request 60; Return B; Return D; Return C.

ii. Show the binary tree representation following Return B.

Table.1. ResultsSequence

Request 70	**A**	**128**		**256**		**512**
Request 35	**A**	**B**	**64**	**256**		**512**
Request 80	**A**	**B**	**64**	**C**	**128**	**512**
Return A	**128**	**B**	**64**	**C**	**128**	**512**
Request 60	**128**	**B**	**D**	**C**	**128**	**512**
Request B	**128**	**64**	**D**	**C**	**128**	**512**
Request D	**256**			**C**	**128**	**512**
Request C	**1024**					

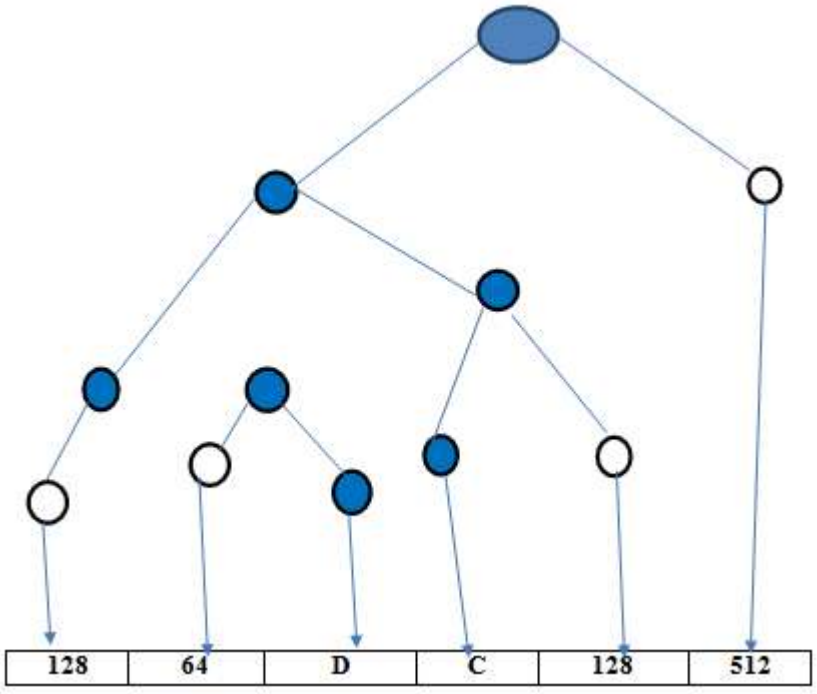


Fig. 5. Binary Tree Representation

II. SUMMARY

Types of partitioning (fixed, dynamic, simple paging, simple segmentation). Abstract of fixed partitioning. About fixed partitioning. Fixed partitioningplacement algorithm (Equal size, unequal size).

Internal/ External fragmentation.

Practical example Fixed Partitioning implementation).

III. CONCLUSION

In this paper Single Contiguous Memory Management and glued Partition

Memory Management schemes for management of main memory that are characterized by contiguous allocation of memory. Single contiguous memory management is inefficient in terms of both CPU and memory utilization and doesn't support multiprogramming. All other schemes support multiprogramming by allowing address spaces of several processes to reside in main memory simultaneously. One approach is to statically divide the available physical memory into variety of fixed partitions and to satisfy requests for memory by granting suitable free partitions, if any. Fixed partition sizes limit the utmost allowable virtual-address space of any given process to the dimensions of the most important partition (unless overlays are used). the entire number of partitions during a given system limits the amount of resident processes. Within the confines of this limit, the effectiveness of the short-term scheduler could also be improved by employing swapping to extend the ratio of resident to ready processes.

Systems with static partitioning suffer from internal fragmentation of memory. Fixed partitioning of memory believe hardware support for relocation and protection. Sharing is sort of restrictive in these systems.

REFERENCE

[1]. OS Concepts, 5th Edition, Silberschatz A., Galvin P.B., John Wiley and Sons.
[2]. Systems Programming and Operating Systems, 2nd Revised Edition, DhamdhereD.M.,Tata McGraw ill publisher Ltd., New Delhi.
[3]. Operating Systems, Godbole A.S., Tata McGraw Hill publisher Ltd., New Delhi.
[4]. Operating Systems, Madnick S.E., Donovan J.T., Tata McGraw Hill publisher Ltd., NewDelhi.
[5]. Operating Systems-A Modern Perspective, Gary Nutt, Pearson Education Asia, 2000.
[6]. Operating Systems, Harris J.A., Tata McGraw Hill publisher Ltd., New Delhi, 2002.