IMPLEMENTATION OF MEMORY PARTITIONING ALGORITHMS IN DISTRIBUTED OPERATING SYSTEM

Stuti Mohan Srivastava[1] 19MCA0014
Yogita[2] 19MCA0009
Deepali Poddar[3] 19MCA0019
Vasudev Gupta[4] 19MCA0011

ABSTRACT

Memory space is divided into blocks of different sizes. Memory management is one of the crucial functions of the Operating System. In the operating system various algorithms are used to allocate empty memory blocks for storing data that is present in the system. It also plays a major role in allocating, identifying and allocating empty memory chunks for different processes which need space for the execution. In this paper, implementation of various memory partitioning algorithms is done using C, along with the comparison of various memory partitioning techniques to find out the best amongst First, Best and Worst Fit algorithms.

Keywords: Operating System, Memory, Memory Management, First Fit, Best Fit, Worst Fit, Partitioning algorithms.

INTRODUCTION

There are multiple blocks of varying sizes and these blocks collectively form the memory space. The function of the operating system is to assign these blocks of memory efficiently to the multiple processes whenever there is a demand for empty memory spaces.
Assignment of these memory blocks to different processes needs to be done efficiently as the primary memory is divided among the user processes and operating system processes. Thus, different algorithms are used by the system for allocating memory.
And these algorithms are also known as memory partitioning algorithms which are mentioned below:

- First Fit
- Best Fit
- Worst Fit

First Fit
It is the resource allocation scheme specially used for allocating memory space. First fit as the name suggests starts scanning the memory blocks from the beginning of the memory to the end of the available memory. It keeps on scanning until the first biggest free memory space is found which can accommodate the data. Then the data is allocated to that memory space. And the remaining or left over space becomes a smaller, separate free space. In case the size of the data which is to be allocated is bigger than the biggest free space in the memory, then the request cannot be met and an error will be generated. One of the advantages of this algorithm is that it is easy to implement as compared to other algorithms and produces bigger holes as compared to other algorithms.

Best Fit
In the best fit partitioning algorithm, it tries to find out the smallest hole that is present in the available memory to fit the requirement of the process. It usually scans the entire block of memory to find out the best suitable block for fulfilling the requirement of the process. This process of scanning the list again and again makes it somewhat slower. One of the drawbacks associated with the best fit algorithm is that even though it selects the nearest size available for the process, it produces smaller separate holes which in future cannot be used to load any other processes. Therefore, the smaller holes become useless. Despite the fact that its name is Best fit, actually it is not best among other partitioning algorithms.

Worst Fit
It is another memory allocation algorithm. Worst fit algorithm scans the entire available memory block again and again to find out the largest hole or memory space in the list so that the requirement of the process can be fulfilled. Even if it also produces larger holes to load other processes still it is not a better approach because it is a very slow process as it searches the entire list every time again and again. This algorithm is not used in Windows as well as Linux operating systems.

PROPOSED ARCHITECTURE

A request of size *n* from a list of free holes

First-fit:

- Allocate the *first* hole that is big enough

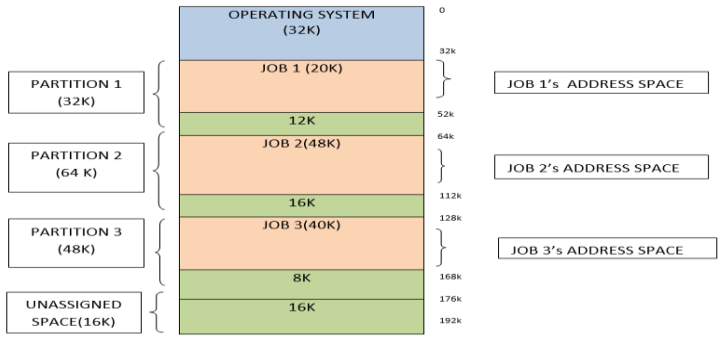
Best-fit:

- Allocate the *smallest* hole that is big enough; must search the entire list, unless ordered by size.
- Produces the smallest leftover hole.

Worst-fit:

- Allocate the largest hole; must also search the entire list.
- Produces the largest leftover hole.

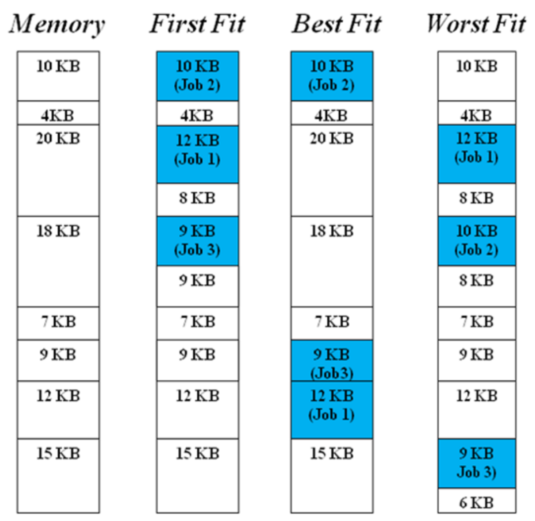
First-fit and best-fit are better than worst-fit in terms of speed and storage utilization.
Partitioned Memory Management:



In windows, the best fit and first fit is more significant as compared to the worst case which is prohibited. As a comparison between three memory partitions, among these we basically prefer the best-fit and first-fit, but worst-fit doesn't support the window of a distributed operating system.

As we have assigned some memory capacity to their respective jobs as job 1, job 2, job 3 with their address space allocated in them. These jobs have been parted into three partition sub categories with allocated memory allocation to control and operate the memory management. Operating system as 32k memory allocation to hold the data and to operate its function dynamically.

First fit and worst case while assigning memory blocks to the process is challenging as the primary memory is needed to be divided among the operating system, user process, and the operating system process. Therefore, the system uses different algorithms to allocate memory from the main memory segment.
In the windows, Worst Fit allocates a process to the partition which is largest sufficient among the freely available partitions available in the main memory. If a large process comes at a later stage, then memory will not have space to accommodate it. Nowadays the technology window does not support the worst fit algorithm due to large wastage of memory in distributed operating systems.
Memory management is the functionality of a distributed operating system which handles or manages primary memory and moves processes back and forth between main memory and disk during execution. It checks how much memory is to be allocated to processes. It decides which process will get memory at what time.
Allocation of memory block in first-fit done first in Job 2(10KB) -> then Job 1(12KB)-> Job 3(9KB) according to first-fit algorithm, as in Best-fit Job 2(10KB)->Job 3(9KB)->Job1(12KB)
In worst-fit memory allocation Job1(12kb)->allocating then Job 2(10kb)->Job 3(9kb) according to the worst fit algorithm, the largest capacity of processor will allocate first with greater capacity.



IMPLEMENTATION

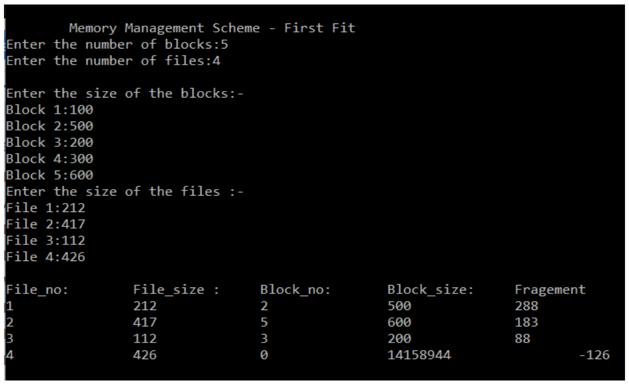
Implementing the first fit, Best fit and worst fit algorithm using C programming language.

First Fit
Implementation of First Fit:

1. Provide Number of Blocks.
2. Enter Number of files.
3. Enter the size of blocks which are free.
4. Enter the size of files.
5. Start by picking each process and check if the process or job is <= memory size or block then allocate the job to the first fitted partition.
6. If not, then the request will not be completed and an error is generated.

Code:
```
#include<stdio.h>
#include<conio.h>
#define max 25
 void main()
{
 int frag[max],b[max],f[max],i,j,nb,nf,temp;
 static int bf[max],ff[max];
 printf("\n\tMemory Management Scheme - 
First Fit");
 printf("\nEnter the number of blocks:");
scanf("%d",&nb);
printf("Enter the number of files:");
 scanf("%d",&nf);
printf("\nEnter the size of the blocks:-\n");
for(i=1;i<=nb;i++)
 {
 printf("Block %d:",i);
 scanf("%d",&b[i]);
 }
 printf("Enter the size of the files :-\n");
 for(i=1;i<=nf;i++)
 {
 printf("File %d:",i);
scanf("%d",&f[i]);
}
 for(i=1;i<=nf;i++)
 {
 for(j=1;j<=nb;j++)
 {
 if(bf[j]!=1)
{
temp=b[j]-f[i];
 if(temp>=0)
{
ff[i]=j;
break;
 }
 }
}
 frag[i]=temp;
 bf[ff[i]]=1;
 }
 printf("\nFile_no:\tFile_size 
:\tBlock_no:\tBlock_size:\tFragement");

for(i=1;i<=nf;i++)
printf("\n%d\t\t%d\t\t%d\t\t%d\t\t%d",i,f[i],f
f[i],b[ff[i]],frag[i]);
 getch();
 }
```



```
        Memory Management Scheme - First Fit
Enter the number of blocks:5
Enter the number of files:4

Enter the size of the blocks:-
Block 1:100
Block 2:500
Block 3:200
Block 4:300
Block 5:600
Enter the size of the files :-
File 1:212
File 2:417
File 3:112
File 4:426

File_no:        File_size :     Block_no:       Block_size:     Fragement
1               212             2               500             288
2               417             5               600             183
3               112             3               200             88
4               426             0               14158944                -126
```

Best Fit

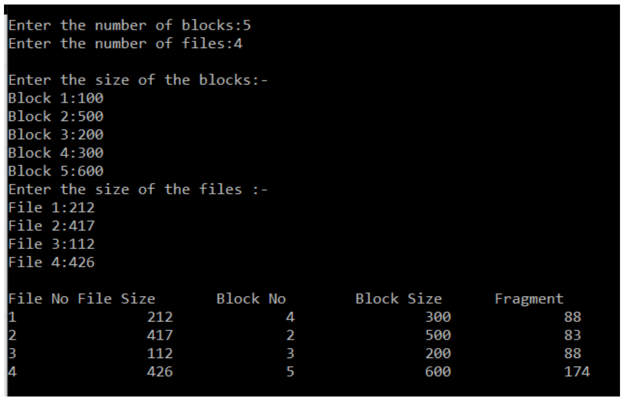
Implementation of Best Fit:

1. Provide Number of Blocks.
2. Enter Number of files.
3. Enter the size of blocks which are free.
4. Enter the size of files.
5. Start by picking each process and find the minimum block size that can be assigned to current process i.e., find min(bockSize[1], blockSize[2],.....blockSize[n]) > process Size[current], if found then assign it to the current process.
6. If no free block is found equal or more than that file size, then ignore that particular process and start checking the next processes.

Code:

```
#include<stdio.h>
#include<conio.h>
#define max 25
 void main()
{
 int 
frag[max],b[max],f[max],i,j,nb,nf,temp,lowe
st=10000;
 static int bf[max],ff[max];
printf("\nEnter the number of blocks:");
scanf("%d",&nb);
printf("Enter the number of files:");
scanf("%d",&nf);
printf("\nEnter the size of the blocks:-\n");
for(i=1;i<=nb;i++) {
 printf("Block %d:",i);
scanf("%d",&b[i]);
}
 printf("Enter the size of the files :-\n");
for(i=1;i<=nf;i++)
 {
 printf("File %d:",i);
scanf("%d",&f[i]);
}
 for(i=1;i<=nf;i++)
{
 for(j=1;j<=nb;j++)
 {
 if(bf[j]!=1)
{
temp=b[j]-f[i];
 if(temp>=0)
if(lowest>temp)
{
 ff[i]=j;

 lowest=temp;
 }
 }
 }
 frag[i]=lowest;
bf[ff[i]]=1;
lowest=10000;
}
 printf("\nFile No\tFile Size \tBlock
No\tBlock Size\tFragment");
for(i=1;i<=nf && ff[i]!=0;i++)
printf("\n%d\t\t%d\t\t%d\t\t%d\t\t%d",i,f[i],f
f[i],b[ff[i]],frag[i]);
 getch();
 }
```



```
Enter the number of blocks:5
Enter the number of files:4

Enter the size of the blocks:-
Block 1:100
Block 2:500
Block 3:200
Block 4:300
Block 5:600
Enter the size of the files :-
File 1:212
File 2:417
File 3:112
File 4:426

File No File Size       Block No        Block Size      Fragment
1               212             4               300             88
2               417             2               500             83
3               112             3               200             88
4               426             5               600             174
```

Worst Fit

Implementation of Worst Fit:

1. Input memory blocks and processes with sizes.
2. Initialize all memory blocks as free.
3. Start by picking each process and find the maximum block size that can be assigned to current process i.e., find max(bockSize[1], blockSize[2],..................blockSize[n]) > processSize[current], if found then

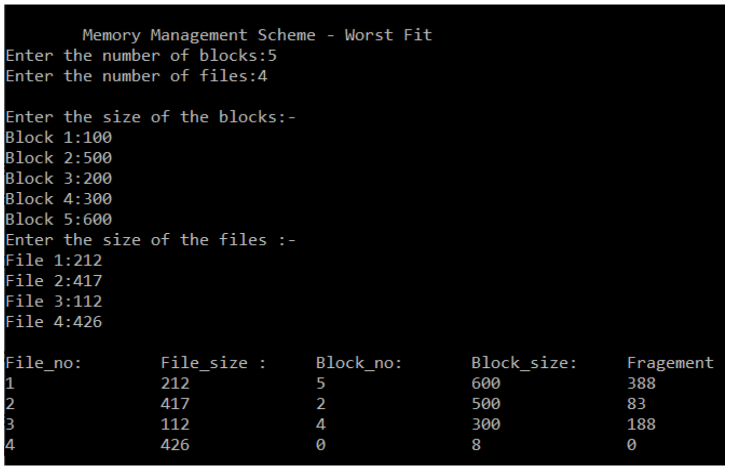
assign it to the current process.

4. If not, then leave that process and keep checking the further processes.

Code:

```
#include<stdio.h>
#include<conio.h>
#define max 25
void main()
{
 int
frag[max],b[max],f[max],i,j,nb,nf,temp,high
est=0;
static int bf[max],ff[max];
 printf("\n\tMemory Management Scheme -
Worst Fit");
 printf("\nEnter the number of blocks:");
 scanf("%d",&nb);
 printf("Enter the number of files:");
 scanf("%d",&nf);
printf("\nEnter the size of the blocks:-\n");
for(i=1;i<=nb;i++)
{
printf("Block %d:",i);
 scanf("%d",&b[i]);
}
printf("Enter the size of the files :-\n");
for(i=1;i<=nf;i++)
{
 printf("File %d:",i);
 scanf("%d",&f[i]);
}
for(i=1;i<=nf;i++)
 {
 for(j=1;j<=nb;j++)
{
 if(bf[j]!=1) //if bf[j] is not allocated
{
 temp=b[j]-f[i];
if(temp>=0)
 if(highest<temp)
{
 ff[i]=j;
highest=temp;
}

}
 }
 frag[i]=highest;
 bf[ff[i]]=1;
 highest=0;
}
 printf("\nFile_no:\tFile_size
:\tBlock_no:\tBlock_size:\tFragement");
 for(i=1;i<=nf;i++)
printf("\n%d\t\t%d\t\t%d\t\t%d\t\t%d",i,f[i],f
f[i],b[ff[i]],frag[i]);
 getch();
}
```



```
        Memory Management Scheme - Worst Fit
Enter the number of blocks:5
Enter the number of files:4

Enter the size of the blocks:-
Block 1:100
Block 2:500
Block 3:200
Block 4:300
Block 5:600
Enter the size of the files :-
File 1:212
File 2:417
File 3:112
File 4:426

File_no:        File_size :     Block_no:       Block_size:     Fragement
1               212             5               600             388
2               417             2               500             83
3               112             4               300             188
4               426             0               8               0
```

LITERATURE SURVEY

[1] **Ahmad Ali Alzubi, Mohammed Al-Maitah, Abdulaziz Alarifi,2019** this paper is introduced a best-fit routing algorithm for non-redundant communication in large-scale IoT based network, best-fit algorithm. Flawless neighbors to prolonged resumption of interrupte communications.It improves application performance by minimizing transmission delay, redundancy and increasing communication rates.

[2] **C. Christober Asir Rajan, A. Amudha,2011** Effect of Reserve in Profit Based Unit Commitment Using Worst Fit Algorithm introduced this paper as, Profit Based Unit Commitment (PBUC) is solved by Worst Fit Algorithm of Memory Management Algorithm. MMA uses Worst Fit allocation for generator scheduling in order to receive the maximum profit in large scale power system by considering the softer demand.

[3] **Smail Niar; Bouziane Beldjilali, Asmaa Bengueddach,2012**, Online First Fit Algorithm for modeling the problem of

configurable cache architecture. Technique is First fit algorithm to find obtains better optimization than previous on-line decision algorithm.

[4] **Ledisi G. Kabari,Tamunoomie S. Gogo,2015**,Efficiency of Memory Allocation Algorithms Using Mathematical Model technique used this paper is First Fit, Best Fit, Worst Fit, Next Fit.
Since similar results were obtained on three different data sets of processes using the same partitions configuration, one can conclude by mathematical induction that it is generally true that best fit algorithm is the best in terms of memory utilization.

[5] **J.P.Hayes,S.Dutt, 1991,** Subcube Allocation in Hypercube Computers this paper uses Best fit algorithm. In this coalescing approach is coupled with a simple best-fit allocation scheme to form the basis of a class of MSS-based strategies that give a substantial performance (hit ratio) improvement over the buddy strategy.

[6] J**ohn E. Shore**, On the external storage fragmentation produced by first-fit and best-fit allocation strategies with technique used is First-fit and Best-fit ALGORITHM
Comparisons of the external fragmentation generated by first-fit and best fit allocation of memory were not consistent.

[7] **Lei Huang, Zhong Liu, Zhi Liu,2014**, An Improved Lowest-level Best Fit Algorithm with Memory for the 2D Rectangular Packing Problem, technique used is Best-Fit algorithm. Heuristic placement algorithms based on three heuristic rules LLLBF, WHBF, LL, and combined with PSO algorithm. -Three new heuristic rules (ILBF) which belong to the class of packing procedure that preserve lowest-level best-fit first.

[8] **Dayong Cao1, V. M. Kotov,2011**, A best-fit heuristic algorithm for two-dimensional bin packing problem this paper uses Best-Fit algorithm.
Two-dimensional bin packing problem is to minimize the number of the used large rectangles for packing a set of small rectangles (items).
- heuristic algorithm could obtain better and reliable results for almost all test instances in less time than some classical algorithms.

[9] **Ziqian Dong, Wenjie Zhuang, Roberto Rojas-Cessa,2019**, Delayed Best-Fit Task Scheduling to Reduce Energy Consumption in Cloud Data Centers, technique used is Best-Fit.
Reducing energy consumption of cloud data center is critical for its sustainable growth.
- delayed best-fit task-scheduling scheme reduces datacenter energy consumption by 15% of that attained by the best-fit algorithm on the same trace, without compromising the average task completion time.

[10] **Xuehong Sun', Yunheo Lit, IoannisLambadaristand YiqiangQ. Zhao',2003,** Performance Analysis of First-Fit Wavelength Assignment Algorithm in Optical Networks, technique used is First-Fit. New analytical technique for the performance analysis of optical networks which uses the first- fit algorithm for wavelength assignment.

[11] **AsmaaBengueddach, SmailNiar, BouzianeBeldjilali,2011,** Online First Fit Algorithm for Modeling the Problem of

Configurable Cache Architecture, technique is used is First-Fit Worst-Fit algorithm. Customizing the cache configuration: number of line, line size and associativity to a particular program needs is well known to have tremendous benefits for performance and energy.

[12**] A. Amudha, IEEE member and Dr. C. ChristoberAsirRajan,2011**, Effect of Reserve in profit based unit commitment using Worst Fit Algorithm.
MMA (Memory Management Algorithm) uses Worst Fit allocation for generator scheduling in order to receive the maximum profit in large scale power system by considering the softer demand.
Also this method gives an idea regarding how much power and reserve should be sold in markets.

[13] **H.M.R.A. Herath; P.A.P. Madushanka; H.P.C. Sampath ; J.H.A.S. Jayamaha, N.D. Jayaweera,2017**,Development of a best-fit algorithm for feature based assembly used best fit algorithms.
feature based assembly and a process of proving the feasibility of robotic assembly applications.

[14] **Dayong Cao1, V. M. Kotov,2011**, A best-fit heuristic algorithm for two dimensional bin packing problem, technique used is Best-Fit algorithm.
Two-dimensional bin packing problem is to minimize the number of the used large rectangles for packing a set of small rectangles (items). - heuristic algorithm could obtain better and reliable results for almost all test instances in less time than some classical algorithms.

CONCLUSION

In this paper, we have compared all the algorithms and have seen that both the algorithms BEST FIT and WORST FIT have their own advantages. This paper concludes that the best fit algorithm is more efficient in terms of memory but if we require the fastest search operation to be performed then in that case first fit algorithm is a better choice. Depending on the requirements we can choose the algorithm which fits best. It also depends on the operating system on which the algorithm is running.

REFERENCES

[1] Dong, Z., Zhuang, W., & Rojas-Cessa, R. (2019, July). Delayed Best-Fit Task Scheduling to Reduce Energy Consumption in Cloud Data Centers. In *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)* (pp. 729-736). IEEE.

[2] Sun, X., Li, Y., Lambadaris, I., & Zhao, Y. Q. (2003, June). Performance analysis of first-fit wavelength assignment algorithm in optical networks. In *Proceedings of the 7th International Conference on*

Telecommunications, 2003. ConTEL 2003. (Vol. 2, pp. 403-409). IEEE.

[3] Bengueddach, A., Niar, S., & Beldjilali, B. (2011, December). Online First Fit Algorithm for modeling the problem of configurable cache architecture. In *ICM 2011 Proceeding* (pp. 1-6). IEEE.

[4] Amudha, A., & Rajan, C. C. A. (2011, July). Effect of Reserve in profit based unit commitment using Worst Fit Algorithm. In *2011 International Conference on Process Automation, Control and Computing* (pp. 1-7). IEEE.

[5] Thompson, R. N., & Wilkinson, J. A. (1963, May). The D825 automatic operating and scheduling program. In *Proceedings of the May 21-23, 1963, spring joint computer conference* (pp. 41-49).

[6] Diwan, A., Tarditi, D., & Moss, E. (1995). Memory system performance of programs with intensive heap allocation. ACM Transactions on Computer Systems (TOCS), 13(3), 244-273.

[7] Losilla, F., Vicente-Chicote, C., Álvarez, B., Iborra, A., & Sánchez, P. (2007, September). Wireless sensor network application development: An architecture-centric mde approach. In *European Conference on Software Architecture* (pp. 179-194). Springer, Berlin, Heidelberg.

[8] Ermedahl, A., & Gustafsson, J. (1997, August). Deriving annotations for tight calculation of execution time. In *European Conference on Parallel Processing* (pp. 1298-1307). Springer, Berlin, Heidelberg.

[9] Rajan, C. C. A., & Mohan, M. R. (2004). An evolutionary programming-based tabu search method for solving the unit commitment problem. *IEEE Transactions on power systems*, *19*(1), 577-585.

[10] Rajan, C. C. A., & Mohan, M. R. (2004). An evolutionary programming-based tabu search method for solving the unit commitment problem. *IEEE Transactions on power systems*, *19*(1), 577-585.

[11] Patil, N. V., & Irabashetti, P. S. (2014). Dynamic memory allocation: role in memory management.

[12] Yadav, D., & Sharma, A. K. (2010, February). Tertiary buddy system for efficient dynamic memory allocation. In *Conference: Proceeding SEPADS* (Vol. 10).

[13] AlZubi, A. A., Al-Maitah, M., & Alarifi, A. (2019). A best-fit routing algorithm for non-redundant communication in large-scale IoT based networks. *Computer Networks*, *152*, 106-113.

[14] Dutt, S., & Hayes, J. P. (1991). Subcube allocation in hypercube computers. *IEEE Transactions on Computers*, (3), 341-352.