

“Implementing First, Best and Worst Fit Contiguous memory allocation ”

Rohin Sabharwal, Vivek Singh, Ujjwal Shaan and Hari Priya Dutta

2021alr032@mietjammu.in

2021alr036@mietjammu.in

2021alr038@mietjammu.in

2021alr046@mietjammu.in

Model Institute of Engineering and Technology, CSE department

ABSTRACT

The technique to control and coordinate the computer memory, to assign blocks to different running programs for the optimization of an entire system's performance, is called memory management. It resides in hardware of the Operating System (OS), and in applications and programs .

Memory management is the main part of the Operating System which is basically used to control or handle the primary memory. Processes move between the disk and the main memory, during the time of execution. It keeps track of every memory location. It also checks the memory requirement for the processes and allocates memory as per that requirement. It also makes decisions about when the memory is allocated to the process. It updates the status of the memory whenever memory gets freed.

First Fit Memory Allocation :- This method keeps the free/busy list of jobs organized by memory location, low-ordered to high-ordered memory. In this method, the first job claims the first available memory with space more than or equal to its size.

Best Fit Memory Allocation :- This method keeps the free/busy list in order by size – smallest to largest. In this method, the operating system first searches the whole of the memory according to the size of the given

job and allocates it to the closest-fitting free partition in the memory, making it able to use memory efficiently.

Worst Fit Memory Allocation:- In this allocation technique, the process traverses the whole memory and always searches for the largest hole/partition, and then the process is placed in that hole/partition. It is a slow process because it has to traverse the entire memory to search the largest hole.

Objectives

- For both fixed and dynamic memory allocation schemes, the operating system must keep a list of each memory location noting which are free and which are busy.
- Then as new jobs come into the system, the free partitions must be allocated.
- These partitions may be allocated by First-Fit Memory Allocation, Best-Fit Memory Allocation and Worst-Fit Memory Allocation.
- Objective is to implement these allocation schemes on the Linux operating system.
- Users should be given a choice to select the particular allocation strategy.

Algorithm

First Fit Memory Allocation Algorithm:

First-Fit Allocation In the first fit algorithm the allocator keeps a list of free blocks (known as the free list) and, on receiving a request for memory, scans along the list for the first block that is large enough to satisfy the request. If the chosen block is significantly larger than that requested, then it is usually split, and the remainder added to the list as another free block. The first fit algorithm performs reasonably well, as it ensures that allocations are quick. When recycling free blocks there is a choice as to where to add the blocks to the free list effectively in what order the free list is kept.

Algorithm for allocate (n)

```

Size(block) = n + size(header)
Scan free list for first block with
nWords >= size(block)
If block not found
    Failure (time for garbage collection!)
Else if free block nWords >= Free
    block nWords - size(block)
    In-use block n words = size(block)
    Else
        Unlink Block from free list
        Return pointer to block
    "Threshold must be at least size
    header) +1 to leave room for header and link
    Threshold can be set higher to combat
    fragmentation

```

Allocation time is $O(K)$ (K -number of Blocks in free list)

Best Fit Memory Allocation Algorithm:

- Get no. of Processes and no. of blocks.
- After that get the size of each block and process requests.
- Then select the best memory block that can be allocated using the above definition.
- Display the processes with the blocks that are allocated to a respective process.
- Value of Fragmentation is optional to display to keep track of wasted memory.
- Stop.

Worst Fit Memory Allocation Algorithm:

- Input memory blocks and processes with sizes.
- Initialize all memory blocks as free.
- Start by picking each process and find the maximum block size that can be assigned to current process i.e., find

```

max(blockSize[1],
    blockSize[2],.....blockSize[n]) >
    processSize[current],

```

- If found then assign it to the current process.
- If not then leave that process and keep checking the further processes.

Conclusion

Main memory management is very important in operating system. Simulations have shown that both First-fit and Bestfit are better than Worst-fit in terms of decreasing both time and storage utilization. Neither First-fit nor Best-fit is clearly better in terms of storage utilization but First-fit is generally faster. Among three algorithms, Best-fit algorithm makes the most efficient use of memory. In this paper, we also proved that Best-fit algorithm is the best in memory utilization.

References

- [1] "Implementation Of Contiguous Memory Management Techniques", Available At <https://www.geeksforgeeks.org/implementation-of-contiguous-memory-managementtechniques/>
- [2] "First-Fit Allocation In Operating System" Available At <https://www.geeksforgeeks.org/first-fit-allocation-in-operating-systems/>
- [3] "Best-Fit Allocation In Operating System" Available At <https://www.geeksforgeekss.org/best-fit-allocation-in-operating-system/?ref=rp>
- [4] "Worst-Fit Allocation In Operating System" Available At <https://www.geeksforgeeks.org/worst-fit-allocation-in-operating-system/?ref=rp>
- [5] M. A. Awais, "Memory Management: Challenges And Techniques For Traditional Memory Allocation Algorithms In Relation With Today's Real Time Needs", IJMST. Vol. 7, No. 3, March 2016 Bpb Publication, 1982.

[6]L. G. Kabari, T [8] “Fragmentation
(Computing)”, 2020. Available At
[https://en.wikipedia.org/wiki/
fragmentation_\(computing\)](https://en.wikipedia.org/wiki/fragmentation_(computing))