# **Contiguous Memory Allocation: First Fit, Best Fit, and Worst Fit**

[](https://medium.com/@khanzadaaneeda?source=post_page-----734fd6f78ab--------------------------------)

[Adeena Ali Syed](https://medium.com/@khanzadaaneeda?source=post_page-----734fd6f78ab--------------------------------)

·

[Follow](https://medium.com/m/signin?actionUrl=https://medium.com/_/subscribe/user/b6b1e9f71a84&operation=register&redirect=https://medium.com/@khanzadaaneeda/contiguous-memory-allocation-first-fit-best-fit-and-worst-fit-734fd6f78ab&user=Adeena+Ali+Syed&userId=b6b1e9f71a84&source=post_page-b6b1e9f71a84----734fd6f78ab---------------------post_header-----------)

3 min read

·

Apr 30, 2023

57

💾 Memory allocation is a critical task in modern operating systems, and one of the most commonly used techniques is contiguous memory allocation. Contiguous memory allocation involves allocating memory to processes in contiguous blocks, where the starting address of each block is adjacent to the previous one. In this blog post, we’ll explore three cases of contiguous memory allocation: First Fit, Best Fit, and Worst Fit, with humor, emojis, and solved examples.

# **🎉 First Fit**

The first-fit algorithm searches for the first free partition that is large enough to accommodate the process. The operating system starts searching from the beginning of the memory and allocates the first free partition that is large enough to fit the process.

For example, suppose we have the following memory partitions:

| 10 KB | 20 KB | 15 KB | 25 KB | 30 KB |

Now, a process requests 18 KB of memory. The operating system starts searching from the beginning and finds the first free partition of 20 KB. It allocates the process to that partition and keeps the remaining 2 KB as free memory.

# **🧐 Best Fit**

The best-fit algorithm searches for the smallest free partition that is large enough to accommodate the process. The operating system searches the entire memory and selects the free partition that is closest in size to the process.

For example, suppose we have the following memory partitions:

| 10 KB | 20 KB | 15 KB | 25 KB | 30 KB |

Now, a process requests 18 KB of memory. The operating system searches for the smallest free partition that is larger than 18 KB, and it finds the partition of 20 KB. It allocates the process to that partition and keeps the remaining 2 KB as free memory.

# **🤔 Worst Fit**

The worst-fit algorithm searches for the largest free partition and allocates the process to it. This algorithm is designed to leave the largest possible free partition for future use.

For example, suppose we have the following memory partitions:

| 10 KB | 20 KB | 15 KB | 25 KB | 30 KB |

Now, a process requests 18 KB of memory. The operating system searches for the largest free partition, which is 30 KB. It allocates the process to that partition and keeps the remaining 12 KB as free memory.

# **👍 Pros and Cons**

First Fit is fast and simple to implement, making it the most commonly used algorithm. However, it can suffer from external fragmentation, where small free partitions are left between allocated partitions.

Best Fit reduces external fragmentation by allocating processes to the smallest free partition, but it requires more time to search for the appropriate partition.

Worst Fit reduces external fragmentation by leaving the largest free partition, but it can lead to inefficient use of memory.

# **🤖 Conclusion**

Contiguous memory allocation is an essential technique used in modern operating systems to allocate memory to processes. First Fit, Best Fit, and Worst Fit are popular algorithms used for contiguous memory allocation. Each algorithm has its advantages and disadvantages, but all are designed to optimize memory allocation and reduce fragmentation. With these techniques, operating systems can efficiently manage memory, making them a critical component of computer science and software engineering.

# Partitioning Algorithms

There are various algorithms which are implemented by the Operating System in order to find out the holes in the linked list and allocate them to the processes.

The explanation about each of the algorithm is given below.

****1. First Fit Algorithm****

First Fit algorithm scans the linked list and whenever it finds the first big enough hole to store a process, it stops scanning and load the process into that hole. This procedure produces two partitions. Out of them, one partition will be a hole while the other partition will store the process.

PlayNext

Unmute

Current TimeÂ 0:00

/

DurationÂ 0:00

Loaded: 0%

Â

Fullscreen

Backward Skip 10sPlay VideoForward Skip 10s

First Fit algorithm maintains the linked list according to the increasing order of starting index. This is the simplest to implement among all the algorithms and produces bigger holes as compare to the other algorithms.

****2. Next Fit Algorithm****

Next Fit algorithm is similar to First Fit algorithm except the fact that, Next fit scans the linked list from the node where it previously allocated a hole.

Next fit doesn't scan the whole list, it starts scanning the list from the next node. The idea behind the next fit is the fact that the list has been scanned once therefore the probability of finding the hole is larger in the remaining part of the list.

Experiments over the algorithm have shown that the next fit is not better then the first fit. So it is not being used these days in most of the cases.

****3. Best Fit Algorithm****

ADVERTISEMENT

ADVERTISEMENT

The Best Fit algorithm tries to find out the smallest hole possible in the list that can accommodate the size requirement of the process.

Using Best Fit has some disadvantages.

1. 1. It is slower because it scans the entire list every time and tries to find out the smallest hole which can satisfy the requirement the process.
2. Due to the fact that the difference between the whole size and the process size is very small, the holes produced will be as small as it cannot be used to load any process and therefore it remains useless.  
   Despite of the fact that the name of the algorithm is best fit, It is not the best algorithm among all.

****4. Worst Fit Algorithm****

The worst fit algorithm scans the entire list every time and tries to find out the biggest hole in the list which can fulfill the requirement of the process.

Despite of the fact that this algorithm produces the larger holes to load the other processes, this is not the better approach due to the fact that it is slower because it searches the entire list every time again and again.

****5. Quick Fit Algorithm****

The quick fit algorithm suggestsmaintaining the different lists of frequently used sizes. Although, it is not practically suggestible because the procedure takes so much time to create the different lists and then expending the holes to load a process.

ADVERTISEMENT

The first fit algorithm is ****the best algorithm**** among all because

1. It takes lesser time compare to the other algorithms.
2. It produces bigger holes that can be used to load other processes later on.
3. It is easiest to implement.