**Implementing First, Best and Worst Fit Contiguous memory allocation techniques by keeping a free/busy list of jobs organized by memory location**

**Problem Statement Description**

* **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.
* **Description**: When the blocks are allocated to a file in such a manner that every logical block of the file gets a contiguous physical block in the hard disk, then this allocation scheme would be known as a contiguous allocation. Contiguous memory allocation can be achieved when we divide the memory into Fixed-Sized Partitions and Variable Size Partitions. It supports a user’s random access to files. The user gets excellent read performance. It is fairly simple to implement.

**Theory**

* **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, first job claims the first available memory with space more than or equal to it’s size. The operating system doesn’t search for appropriate partition but just allocate the job to the nearest memory partition available with sufficient size.

* **Advantages of First-Fit Memory Allocation:**  
  It is fast in processing. As the processor allocates the nearest available memory partition to the job, it is very fast in execution.
* **Disadvantages of First-Fit Memory Allocation :**  
  It wastes a lot of memory. The processor ignores if the size of partition allocated to the job is very large as compared to the size of job or not. It just allocates the memory. As a result, a lot of memory is wasted and many jobs may not get space in the memory, and would have to wait for another job to complete.
* **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. Here the jobs are in the order from smallest job to largest job.

* **Advantages of Best-Fit Allocation :**   
  Memory Efficient. The operating system allocates the job minimum possible space in the memory, making memory management very efficient. To save memory from getting wasted, it is the best method.
* **Disadvantages of Best-Fit Allocation :**   
  It is a Slow Process. Checking the whole memory for each job makes the working of the operating system very slow. It takes a lot of time to complete the work

**Worst Fit Memory Allocation:-**

In this allocation technique, the process traverses the whole memory and always search 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.

* **Advantages of Worst-Fit Allocation :**   
  Since this process chooses the largest hole/partition, therefore there will be large internal fragmentation. Now, this internal fragmentation will be quite big so that other small processes can also be placed in that leftover partition.
* **Disadvantages of Worst-Fit Allocation :**   
  It is a slow process because it traverses all the partitions in the memory and then selects the largest partition among all the partitions, which is a time-consuming process.