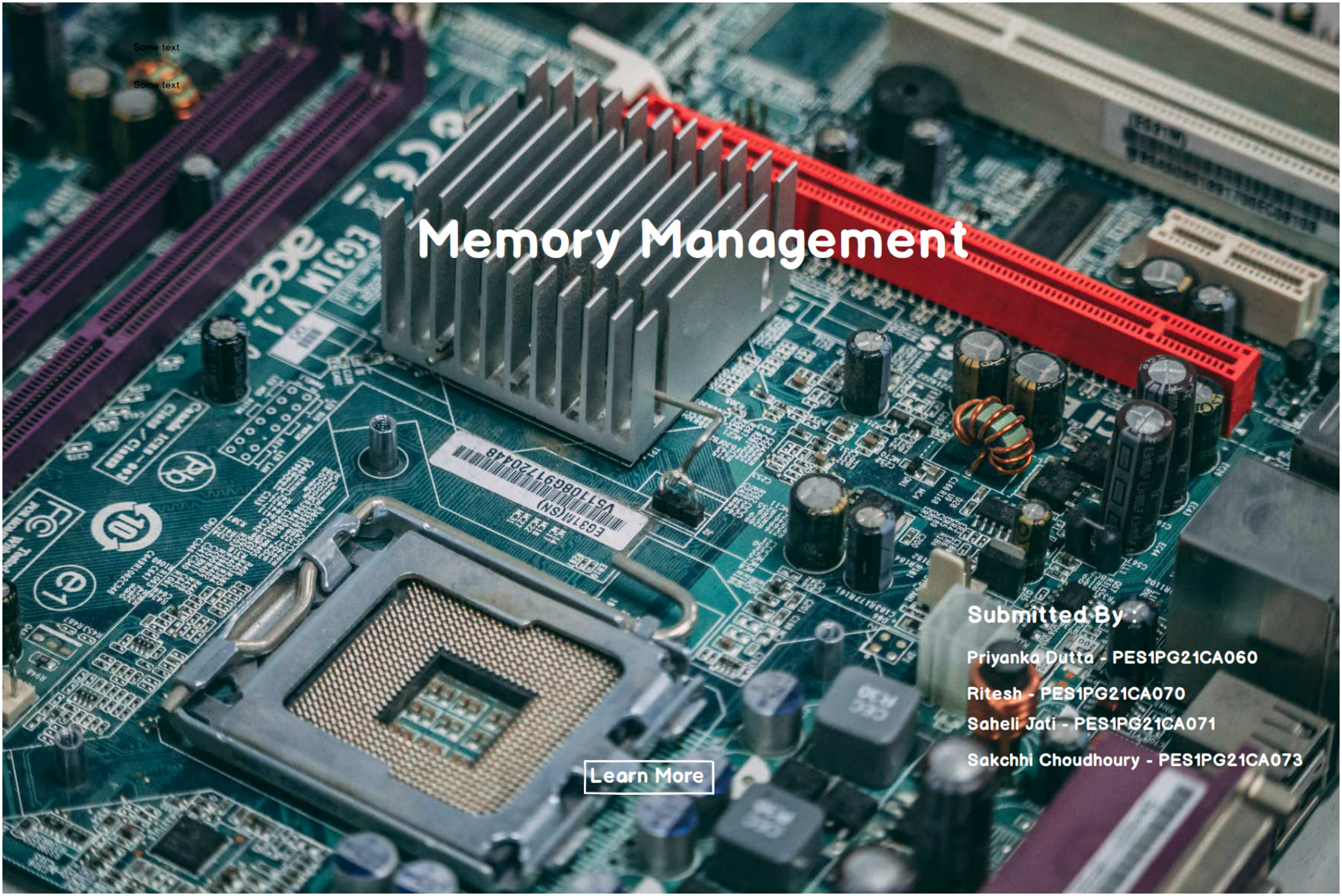

Some text
Some text
Memory Management
Learn More
Submitted By :
Priyanka Dutta - PES1PG21CA060
Ritesh - PES1PG21CA070
Saheli Jati - PES1PG21CA071
Sakchhi Choudhoury - PES1PG21CA073

Memory Management

What is Memory Management...?

The term Memory can be defined as a collection of data in a specific format. It is used to store instructions and processed data. The memory comprises a large array or group of words or bytes, each with its own location. The primary motive of a computer system is to execute programs. These programs, along with the information they access, should be in the main memory during execution. The CPU fetches instructions from memory according to the value of the program counter.

Why it is Required...?

Allocate and de-allocate memory before and after process execution.

--> To keep track of used memory space by processes.

-->To minimize fragmentation issues.

-->To proper utilization of main memory.

-->To maintain data integrity while executing of process.

What is Main Memory...?

The main memory is central to the operation of a modern computer. Main Memory is a large array of words or bytes, ranging in size from hundreds of thousands to billions. Main memory is a repository of rapidly available information shared by the CPU and I/O devices. Main memory is the place where programs and information are kept when the processor is effectively utilizing them. Main memory is associated with the processor, so moving instructions and information into and out of the processor is extremely fast. Main memory is also known as RAM(Random Access Memory)..

CONTIGUOUS MEMORY ALLOCATION

Contiguous memory allocation is basically a method in which a single contiguous section/part of memory is allocated to a process or file needing it

Types of Contiguous Memory Allocation--

STATIC AND FIXED

Contiguous memory allocation is basically a method in which a single contiguous section/part of memory is allocated to a process or file needing it.

DYNAMIC

The entire program and all data of a process must be in physical memory for the process to execute. So, the size of a process is limited to the size of physical memory. To gain proper memory utilization, dynamic loading is used. In dynamic loading, a routine is not loaded until it is called.

FIXED PARTITION

The operating system maintains a table that indicates which parts of memory are available and which are occupied by processes. Initially, all memory is available for user processes and is considered one large block of available memory. This available memory is known as "Hole". When the process arrives and needs memory, we search for a hole that is large enough to store this process. If the requirement fulfills then we allocate memory to process, otherwise keeping the rest available to satisfy future requests. While allocating a memory sometimes dynamic storage allocation problems occur, which concerns how to satisfy a request of size n from a list of free holes.

FIRST FIT

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.

BEST FIT

The best fit deals with allocating the smallest free partition which meets the requirement of the requesting process. This algorithm first searches the entire list of free partitions and considers the smallest hole that is adequate. It then tries to find a hole which is close to actual process size needed.

WORST FIT

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.

Fragmentation

Fragmentation is an unwanted problem where the memory blocks cannot be allocated to the processes due to their small size and the blocks remain unused. It can also be understood as when the processes are loaded and removed from the memory they create free space or hole in the memory and these small blocks cannot be allocated to new upcoming processes and results in inefficient use of memory.

Types of Fragmentation--

Internal Fragmentation

In this fragmentation, the process is allocated a memory block of size more than the size of that process. Due to this some part of the memory is left unused and this cause internal fragmentation. Example: Suppose there is fixed partitioning (i.e. the memory blocks are of fixed sizes) is used for memory allocation in RAM. These sizes are 2MB, 4MB, 4MB, 8MB. Some part of this RAM is occupied by the Operating System (OS).

External Fragmentation

In this fragmentation, although we have total space available that is needed by a process still we are not able to put that process in the memory because that space is not contiguous. This is called external fragmentation. Example: Suppose in the above example, if three new processes P2, P3, and P4 come of sizes 2MB, 3MB, and 6MB respectively. Now, these processes get memory blocks of size 2MB, 4MB and 8MB respectively allocated.

Algorithms for Memory Allocation

The operating system uses different memory management schemes to optimize resource allocation. The responsibility of these schemes is to allocate memory chunks based on the demand by the operating system. The three most commonly used allocation schemes are

-> First Fit
-> Best Fit
--> Worst Fit

FIRST FIT

First Fit Algorithm is the simplest technique of allocating the memory block to the processes amongst all. In this algorithm, the pointer keeps track of all the free blocks in the memory and accepts the request of allocating a memory block to the coming process. After that pointer start searching for the largest first free block for the process and allocate that memory block to the coming process. In this, two partitions are created one is for the hole and the one will store the processes.

Algorithm:

Step 1.Get the number of process and number of blocks.

Step 2.Get the size of each block.

Step 3. Allocate process If (size of the block > = size of the process)

//allocate the process

else

//move on to the next blog

Step 4.Display the process with the blocks allocated to a respective process

Step 5.Stop
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

The best fit deals with allocating the smallest free partition which meets the requirement of the requesting process. This algorithm first searches the entire list of free partitions and considers the smallest hole that is adequate. It then tries to find a hole which is close to actual process size needed.

Algorithm:
Step1. Enter the memory blocks with size.
Step2. Enter the process blocks with size.
Step3. Set all the memory blocks as free.
Step4. Start by picking up each process.
Step5. Find the minimum block size that is best to assign to the current process.
Step6. If the best fit memory size is found, it is allocated to the process.
Step7. If the memory block and memory demand do not match, leave the process and search for another process.
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

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.
Algorithm:

Step 1: Input memory block with a size.

Step 2: Input process with size.

Step 3: Initialize by selecting each process to find the maximum block size that can be assigned to the current process.

Step 4: If the condition does not fulfill, they leave the process.

Step 5: If the condition is not fulfilled, then leave the process and check for the next process.

Step 6: Stop.

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.

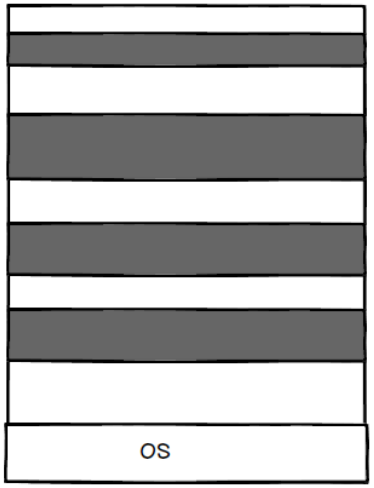
A Web Page

http://

Memory Management

HOME | DESCRIPTION | ALGORITHM | IMPLEMENTATION

Implentation of Memory Allocation Algorithm



Enter the size of the memory

Enter the process size for P1

Enter the process size for P2

Enter the process size for P3

Enter the process size for P4

Choose the algorithm ☐ First Fit ☐ Best Fit ☐ Worst Fit

Submit