What is Fragmentation and what are its types?

In **contiguous memory allocation** whenever the processes come into RAM, space is allocated to them. These spaces in RAM are divided either on the basis of **fixed partitioning** (the size of partitions are fixed before the process gets loaded into RAM) or **dynamic partitioning** (the size of the partition is decided at the run time according to the size of the process). As the process gets loaded and removed from the memory these spaces get broken into small pieces of memory that it can't be allocated to the coming processes. This problem is called **fragmentation** . In this blog, we will study how these free space and fragmentations occur in memory. So, let's get started.

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. Basically, there are two types of fragmentation:

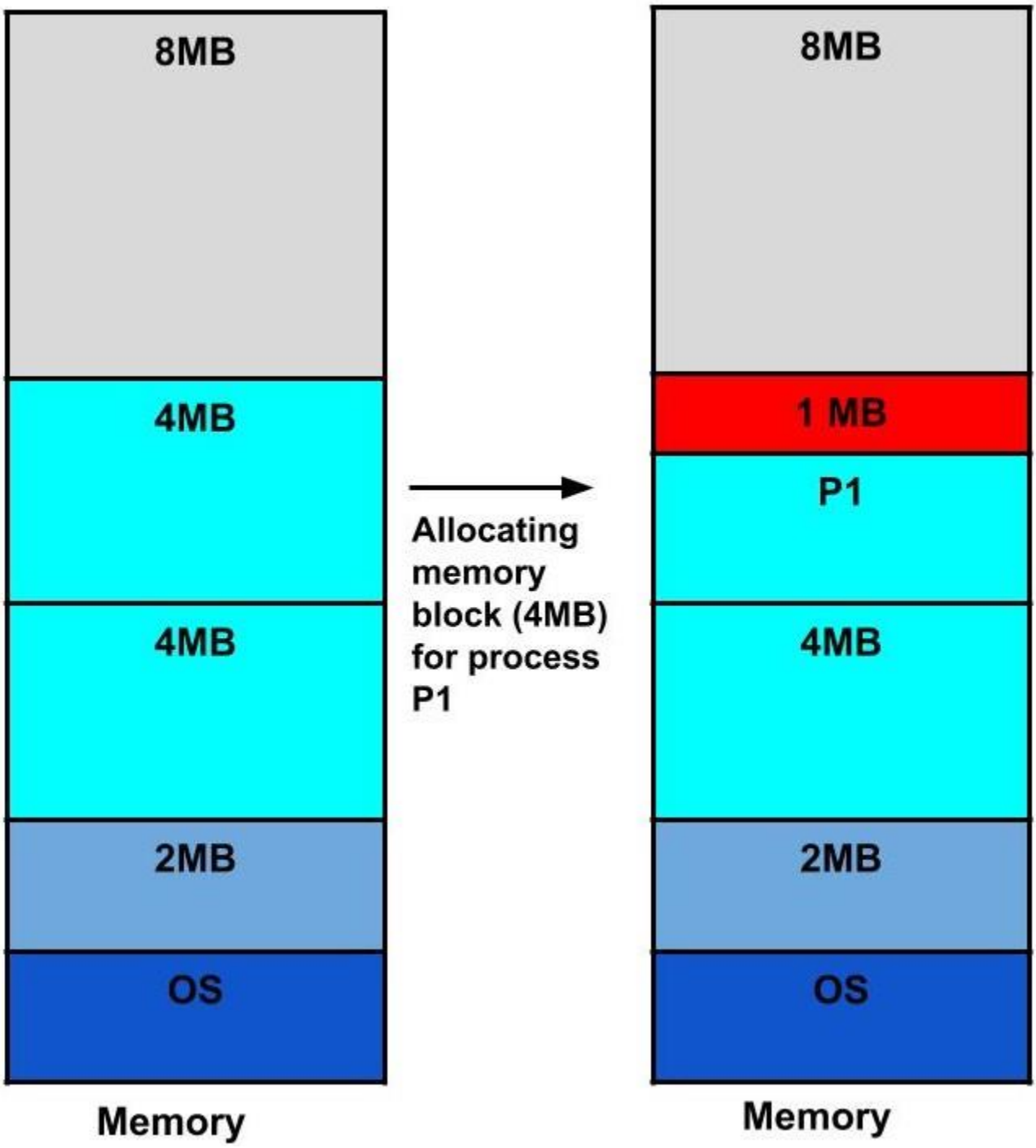
- Internal Fragmentation
- External Fragmentation

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).

Now, suppose a process P1 of size 3MB comes and it gets memory block of size 4MB. So, the 1MB that is free in this block is wasted and this space can't be utilized for allocating memory to some other process. This is called **internal fragmentation** .



How to remove internal fragmentation?

This problem is occurring because we have fixed the sizes of the memory blocks. This problem can be removed if we use dynamic partitioning for allocating space to the process. In dynamic

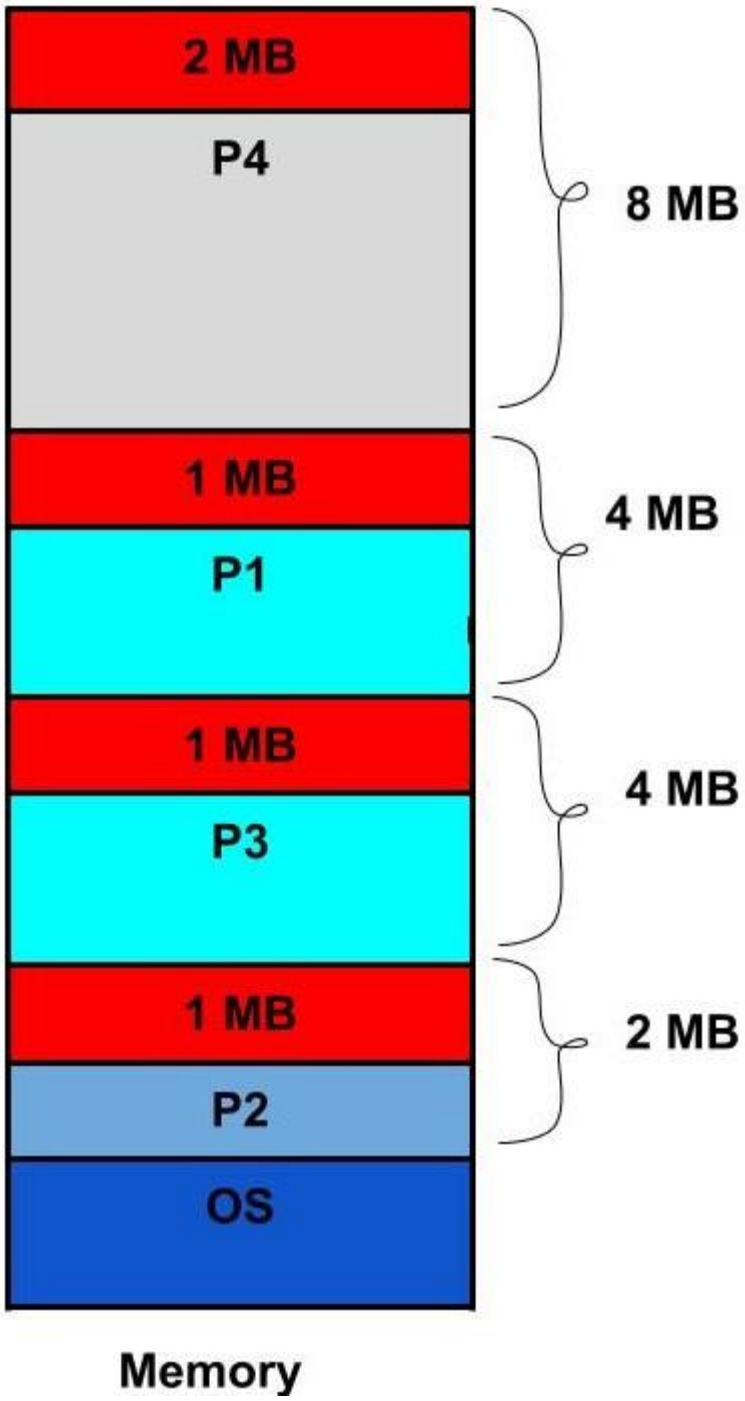
partitioning, the process is allocated only that much amount of space which is required by the process. So, there is no internal fragmentation.

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.

So, now if we closely analyze this situation then process P3 (unused 1MB)and P4(unused 2MB) are again causing internal fragmentation. So, a total of 4MB (1MB (due to process P1) + 1MB (due to process P3) + 2MB (due to process P4)) is unused due to internal fragmentation.



Now, suppose a new process of 4 MB comes. Though **we have a total space of 4MB** still **we can't allocate this memory** to the process. This is called **external fragmentation** .

How to remove external fragmentation?

This problem is occurring because **we are allocating memory continuously** to the processes. So, if we remove this condition external fragmentation can be reduced. This is what done in **paging & segmentation** (non-contiguous memory allocation techniques) where memory is allocated non-contiguously to the processes. We will learn about paging and segmentation in the next blog.

Another way to remove external fragmentation is **compaction** . When dynamic partitioning is used for memory allocation then external fragmentation can be reduced by **merging all the free memory** together in one large block. This technique is also called **defragmentation.** This larger block of memory is then used for allocating space according to the needs of the new processes.