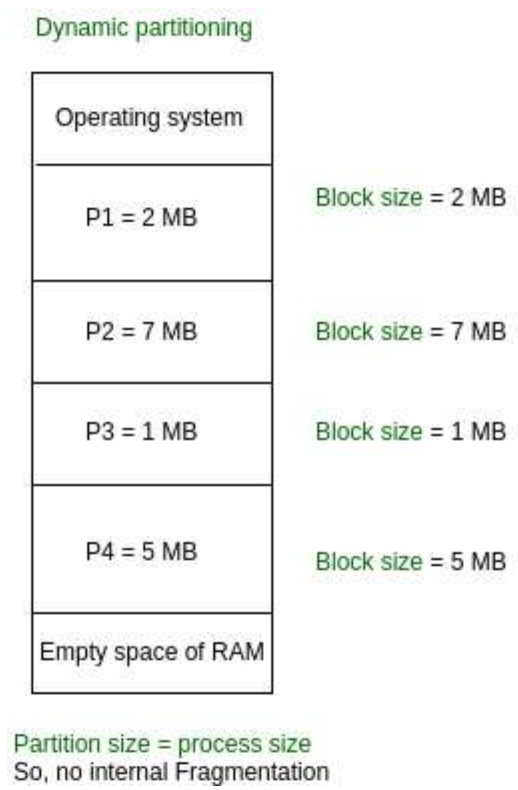
Unit: III
Lecture: 5
(Memory Management)
Memory Allocation Techniques (Part-II)

Variable Partition Scheme or Dynamic Partition Scheme:

In this partitions are not made before the execution or during system configure.

- Initially RAM is empty and partitions are made during the run-time according to process's need instead of partitioning during system configure.
- The size of partition will be equal to incoming process.
- The partition size varies according to the need of the process so that the internal fragmentation can be avoided to ensure efficient utilization of RAM.
- Number of partitions in RAM is not fixed and depends on the number of incoming process and main memory's size.



Advantages of Variable Partitioning:

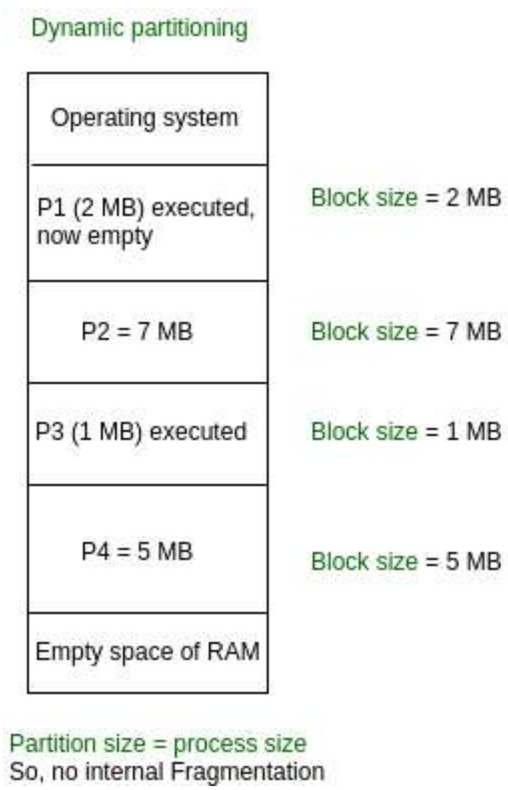
- No internal fragmentation:
  - In variable partitioning, space in main memory is allocated strictly according to the need of process; hence there is no case of internal fragmentation. There will be no unused space left in the partition.

- No restriction on Degree of Multiprogramming:
    - More number of processes can be accommodated due to absence of internal fragmentation. A process can be loaded until the memory is empty.
- No Limitation on the size of the process:
    - As the partition size is decided according to the process size, the process size can't be restricted in variable partitioning.

Disadvantages of Variable Partitioning:

- Difficult Implementation:
    - Implementing variable partitioning is difficult as compared to fixed partitioning as it involves allocation of memory during run time rather than during system configure.
- External Fragmentation:
    - There will be external fragmentation inspite of absence of internal fragmentation.

Example:



Suppose in above example:

Process P1 (2MB) and process P3(1MB) completed their execution.

➔ two spaces are left i.e. 2 MB and 1 MB.

Let's suppose process P5 of size 3MB comes. The empty space in memory cannot be allocated as no spanning is allowed in contiguous allocation.

Because, the process must be contiguously present in main memory to get executed.

➔ results in External Fragmentation.

So, P5 of size 3 MB cannot be accommodated in spite of required available space.

To overcome the external fragmentation problem:

- **Compaction:** In the compaction technique, all free memory space combines and makes one large block. So, this space can be used by other processes effectively.
- Another solution is to allow the logical address space of the processes to **be noncontiguous**, thus permit a process to be allocated physical memory wherever its available.

Allocation Methods in Contiguous Memory Management:

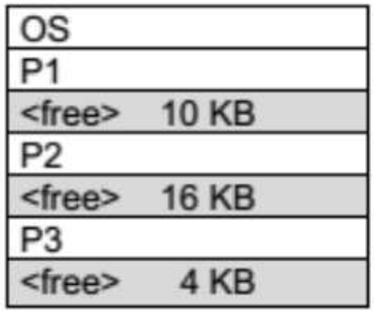
i.e. how holes (free memory blocks) will be allocated to the processes.

In general, there are three common algorithms for searching the list of free blocks for a specific amount of memory that may be allocated to a process:

1. **First Fit:**
   - Allocate the first free block that is large enough for the new process.
   - This is a fast algorithm.
2. **Best Fit:**
   - Allocate the smallest block among those that are large enough for the new process.
   - In this method, the OS has to search the entire list or it can keep it sorted and stop when it hits an entry which has a size larger than the size of new process.
   - This algorithm produces the smallest left over block.
   - However, it requires more time for searching the entire list or sorting it.
3. **Worst Fit:**
   - Allocate the largest block among those that are large enough for the new process.
   - Again a search of the entire list or sorting is needed.
   - This algorithm produces the largest over block.

Example:

Consider the following memory map:



Assumes a new process P4 comes with a memory requirement of 3KB, then it will be allocated as follows depending on the algorithm used:

New memory arrangements with respect to each algorithms will be as follows: