Memory management:-

Introduction :-

- Memory management is the process of controlling and coordinating computer memory, assigning portions called blocks to various running programs to optimize (improve) overall system performance. Memory management resides (settle/live) in hardware, in the OS (operating system), and in programs and applications.

- In hardware, memory management involves components that physically store data, such as RAM (random access memory) chips, memory caches, and flash-based SSDs (solid-state drives).
- In the OS, memory management involves the allocation of specific memory blocks to individual programs as user demands change.
- At the application level, memory management ensures the availability of adequate (proper) memory for the objects and data structures of each running program at all times.
- Application memory management combines two related tasks, known as allocation and recycling.

- When the program requests a block of memory, a part of the memory manager called the allocator assigns that block to the program.
- When a program no longer needs the data in previously allocated memory blocks, those blocks become available for reassignment. This task can be done manually (by the programmer) or automatically (by the memory manager).

Contiguous Memory Allocation :-

- In contiguous memory allocation each process is contained in a single contiguous block of memory.
- Memory is divided into several fixed size partitions. Each partition contains exactly one process.
- When a partition is free, a process is selected from the input queue and loaded into it.
- The free blocks of memory are known as *holes*. The set of holes is searched to determine which hole is best to allocate.

1.)Memory Allocation:-

- Memory allocation is a process by which computer programs and services are assigned with physical or virtual memory space.

  Memory allocation is the process of reserving a partial or complete portion of computer memory for the execution of programs and processes. Memory allocation is achieved through a process known as memory management.

- Memory allocation is primarily(first) a computer hardware operation but is managed through operating system and software applications. Memory allocation process is quite similar in physical and virtual memory management.
- Programs and services are assigned with a specific memory as per their requirements when they are executed. Once the program has finished its operation or is idle, the memory is released and allocated to another program or merged within the primary memory.

- Main memory usually has two partitions −
- **Low Memory** − Operating system resides(live) in this memory.
- **High Memory** − User processes are held(keep) in high memory.
- Operating system uses the following memory allocation mechanism.

S.N.	Memory Allocation & Description
1	**Single-partition allocation** In this type of allocation, relocation-register scheme is used to protect user processes from each other, and from changing operating-system code and data. Relocation register contains value of smallest physical address whereas limit register contains range of logical addresses. Each logical address must be less than the limit register.
2	**Multiple-partition allocation** In this type of allocation, main memory is divided into a number of fixed-sized partitions where each partition should contain only one process. When a partition is free, a process is selected from the input queue and is loaded into the free partition. When the process terminates, the partition becomes available for another process.

2.) Fragmentation :-

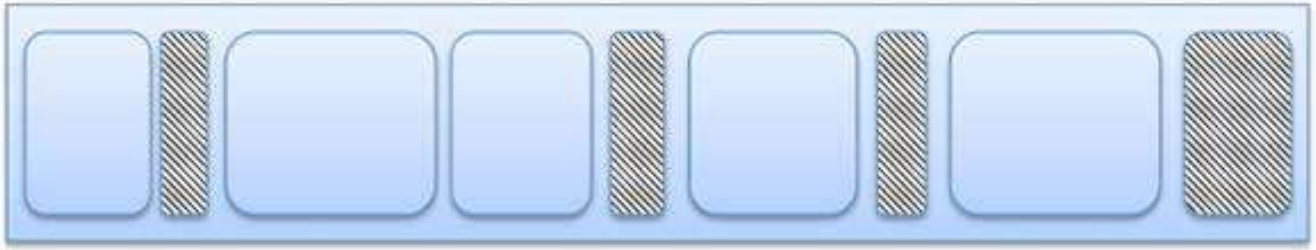
- As processes are loaded and removed from memory, the free memory space is broken into little pieces. It happens after sometimes that processes cannot be allocated to memory blocks considering their small size and memory blocks remains unused. This problem is known as Fragmentation.

Fragmentation is of two types –

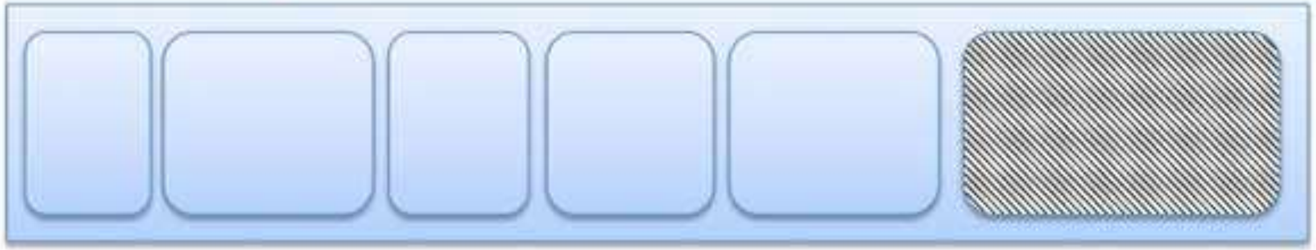
S.N.	Fragmentation & Description
1	**External fragmentation** **External fragmentation** arises when free **memory** is separated into small blocks and is interspersed by allocated **memory**. Total memory space is enough to satisfy a request or to reside a process in it, but it is not contiguous, so it cannot be used.
2	**Internal fragmentation** Memory block assigned to process is bigger. Some portion of memory is left unused, as it cannot be used by another process.

- The following diagram shows how fragmentation can cause waste of memory and a compaction technique can be used to create more free memory out of fragmented memory −
- (**Compaction** is a process in which the free space is collected in a large **memory**)

Fragmented memory before compaction



Memory after compaction



External fragmentation can be reduced by compaction or shuffle memory contents to place all free memory together in one large block. To make compaction feasible, relocation should be dynamic.
The internal fragmentation can be reduced by effectively assigning the smallest partition but large enough for the process.