Memory Management

Memory management is the functionality of an operating system which handles or manages primary memory and moves processes back and forth between main memory and disk during execution. Memory management keeps track of each and every memory location, regardless of either it is allocated to some process or it is free. It checks how much memory is to be allocated to processes. It decides which process will get memory at what time.

Memory Protection

Memory protection is a phenomenon by which we control memory access rights on a computer. Memory protection is required to protect Operating System from the user processes and user processes from one another.

Contiguous Memory Allocation

Main memory usually has two partitions −

Low Memory − Operating system resides in this memory.

High Memory − User processes are held in high memory.

Operating system uses the following memory allocation mechanism:

Single-Partition Allocation OR Variable Partition Method: 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.

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.

Memory Allocation

Each process is contained in a single contiguous section of memory. There are two methods namely:

Fixed Partition-Method: Divide memory into fixed size partitions, where each partition has exactly one process. The drawback is memory space unused within a partition is wasted.

(Example: when process size<partition size)

Variable Partition-Method: Divide memory into variable size partitions, depending upon the size of the incoming process. When a process terminates, the partition becomes available for another process. As the process complete and leave they create holes in the main memory. Now there comes a general dynamic storage allocation problem.

Dynamic Storage-Allocation Problem

The following are the solutions to the dynamic storage allocation problem:

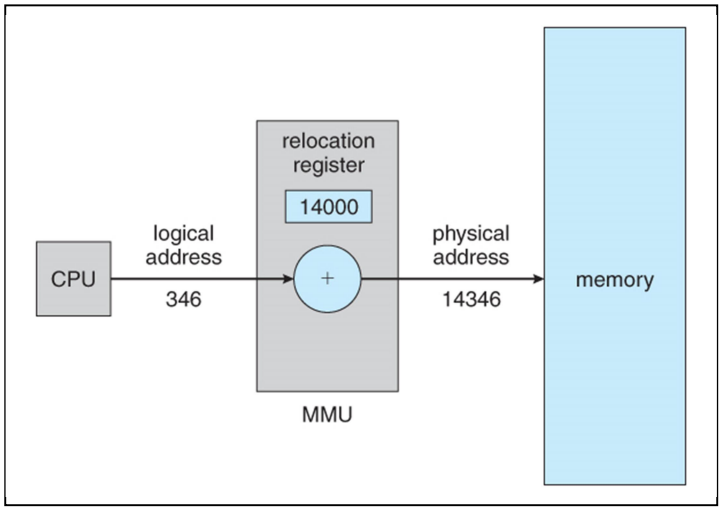
1. First Fit: In the first fit, partition is allocated which is first sufficient from the top of Main Memory.

2. Best Fit: Allocate the process to the partition which is first smallest sufficient partition among the free available partition.

3. Worst Fit: Allocate the process to the partition which is largest sufficient among the freely available partitions available in the main memory.

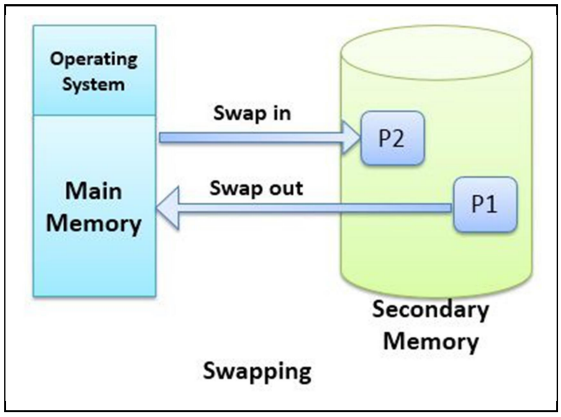
4. Next Fit: Next fit is similar to the first fit but it will search for the first sufficient partition from the last allocation point.

Logical and Physical address space



Sr. No.	Basis of Comparison	Logical Address	Physical Address
1.	Basic	It is the virtual address generated by CPU.	The physical address is a location in a memory unit.
2.	Variation	Keeps on changing.	Always stays the same.
3.	Visibility	The user can view the logical address of a program.	The user can never view physical address of program.
4.	Access	The user uses the logical address to access the physical address.	The user can not directly access physical address.
5.	Generation	The Logical Address is generated by the CPU.	Physical Address is Computed by MMU.

Swapping



Swapping is mechanism in which a process can be swapped temporarily out of main memory (or move) to secondary storage (disk) and make that memory available to other processes. At some later time, the system swaps back the process from the secondary storage to main memory. Swapping is used for improving the performance of the system. In this the process those are waiting for some input and output are transferred to the physical memory from they are running and the processes those are ready for the execution will be execute by the CPU.

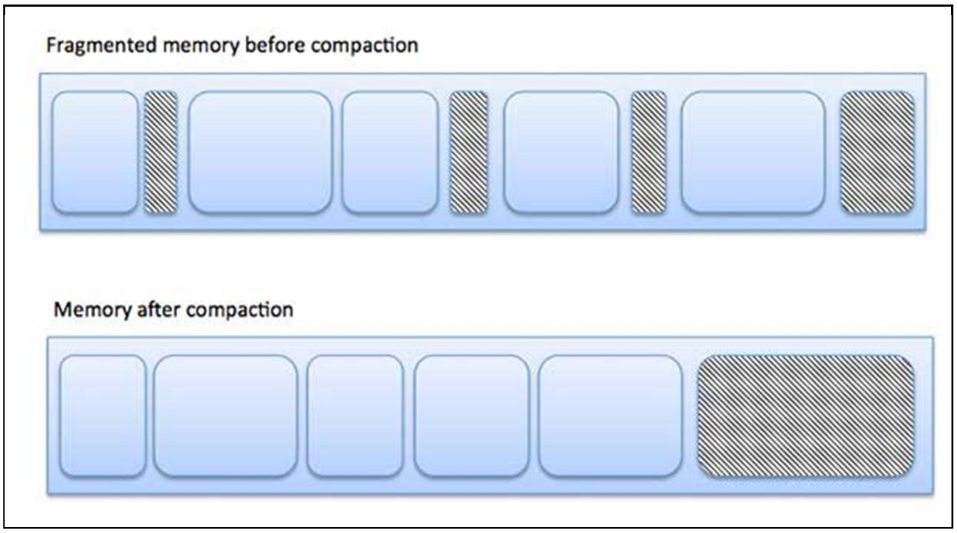
Fragmentation

Fragmentation occurs in a dynamic memory allocation system when most of the free blocks are too small to satisfy any request. It is generally termed as inability to use the available memory.

Types of Fragmentation

External Fragmentation: External Fragmentation happens when a dynamic memory allocation algorithm allocates some memory and a small piece is left over that cannot be effectively used.

Internal Fragmentation: Internal fragmentation is the space wasted inside of allocated memory blocks because of restriction on the allowed sizes of allocated blocks.



Compaction

Compaction is a process in which the free space is collected in a large memory chunk to make some space available for processes. Compaction refers to combining all the empty spaces together and processes. Compaction helps to solve the problem of fragmentation, but it requires too much of CPU time.

Paging

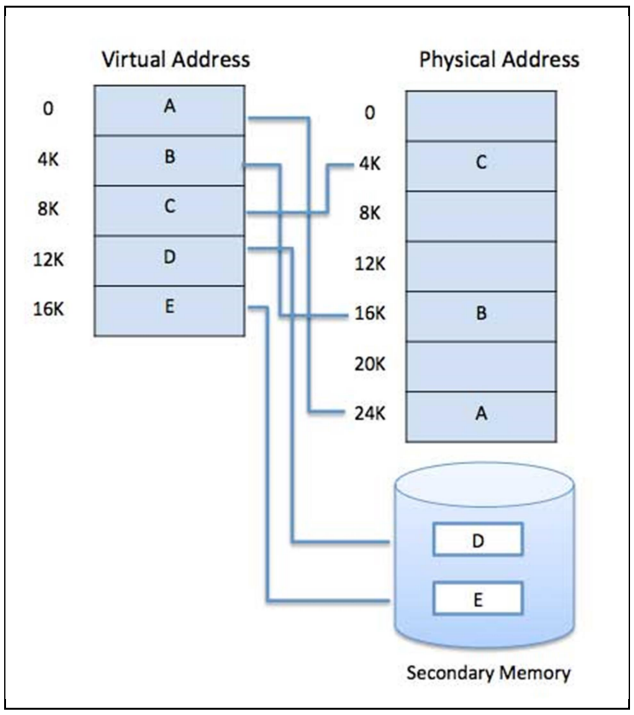
Paging is a memory management technique in which the memory is divided into fixed size pages. Paging is used for faster access to data. When a program needs a page, it is available in the main memory as the OS copies a certain number of pages from your storage device to main memory. Paging allows the physical address space of a process to be noncontiguous.

Segmentation

Segmentation is another memory management technique in which each job is divided into several segments of different sizes, one for each module that contains pieces that perform related functions. Each segment is actually a different logical address space of the program.

Segmentation memory management works very similar to paging but here segments are of variable-length where as in paging pages are of fixed size.

Virtual Memory



A computer can address more memory than the amount physically installed on the system. This extra memory is actually called virtual memory.

Virtual Memory is a space where large programs can store themselves in form of pages while their execution and only the required pages or portions of processes are loaded into the main memory. Virtual Memory is implanted by Demand Paging.

Demand Paging

Demand paging is a type of swapping done in virtual memory systems. In demand paging, the data is not copied from the disk to the RAM until they are needed or being demanded by some program. The data will not be copied when the data is already available on the memory.

Pure Demand Paging

In pure demand paging, even a single page is not loaded into memory initially. Hence pure demand paging causes a page fault. Page fault is the situation in which the page is not available whenever a processor needs to execute it.

Page Replacement Algorithm

Page replacement algorithms are the techniques using which an Operating System decides which memory pages to swap out, write to disk when a page of memory needs to be allocated.

Page Fault

A page fault is a type of interrupt, raised by the hardware when a running program accesses a memory page that is mapped into the virtual address space, but not loaded in physical memory.

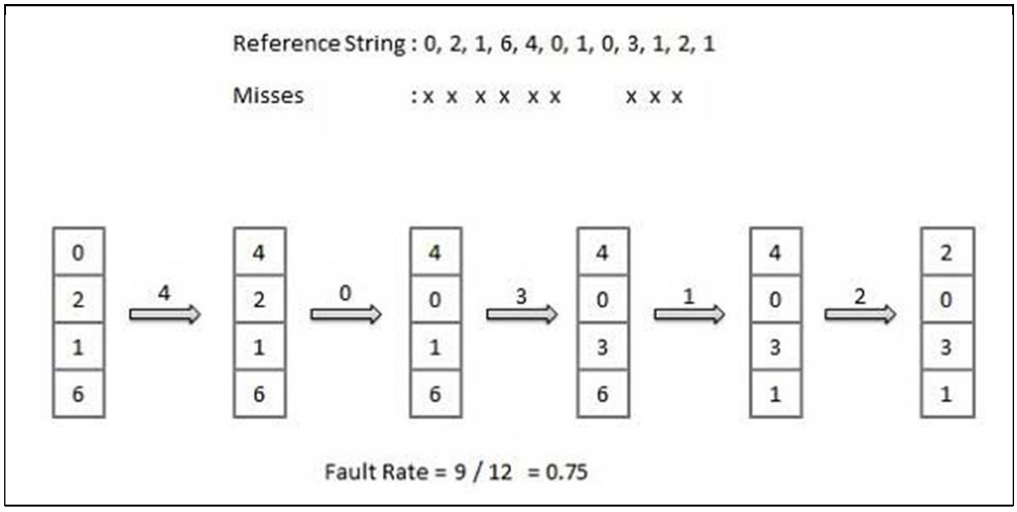
Reference String

The string of memory references is called reference string. Reference strings are generated artificially or by tracing a given system and recording the address of each memory reference.

Types of Page Replacement Algorithm

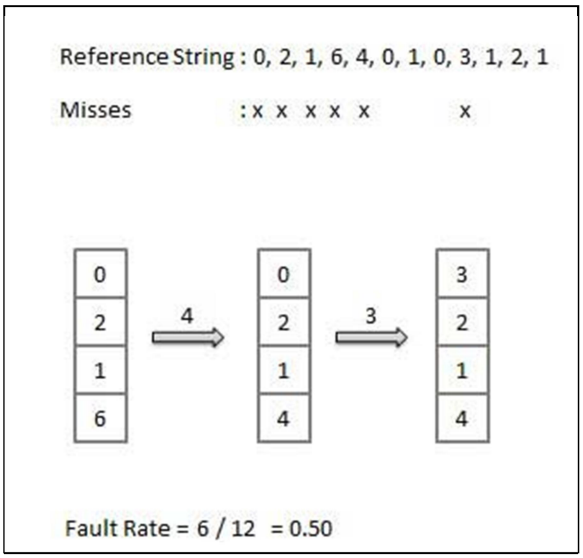
First In First Out

This is the simplest page replacement algorithm. In this algorithm, operating system keeps track of all pages in the memory in a queue, oldest page is in the front of the queue.



Optimal Page replacement

In this algorithm, pages are replaced which are not used for the longest duration of time in the future.



Least Recently Used

In this algorithm page will be replaced which is least recently used.