UNIT- 4

Memory Management

→ Program must be brought into memory & placed within a process for it to be run.

→ Input queue or job queue - collection of processes on the disk that are waiting to be brought into memory to run the program.

→ User programs go through several steps before being run.

Binding of Instructions & Data to Memory

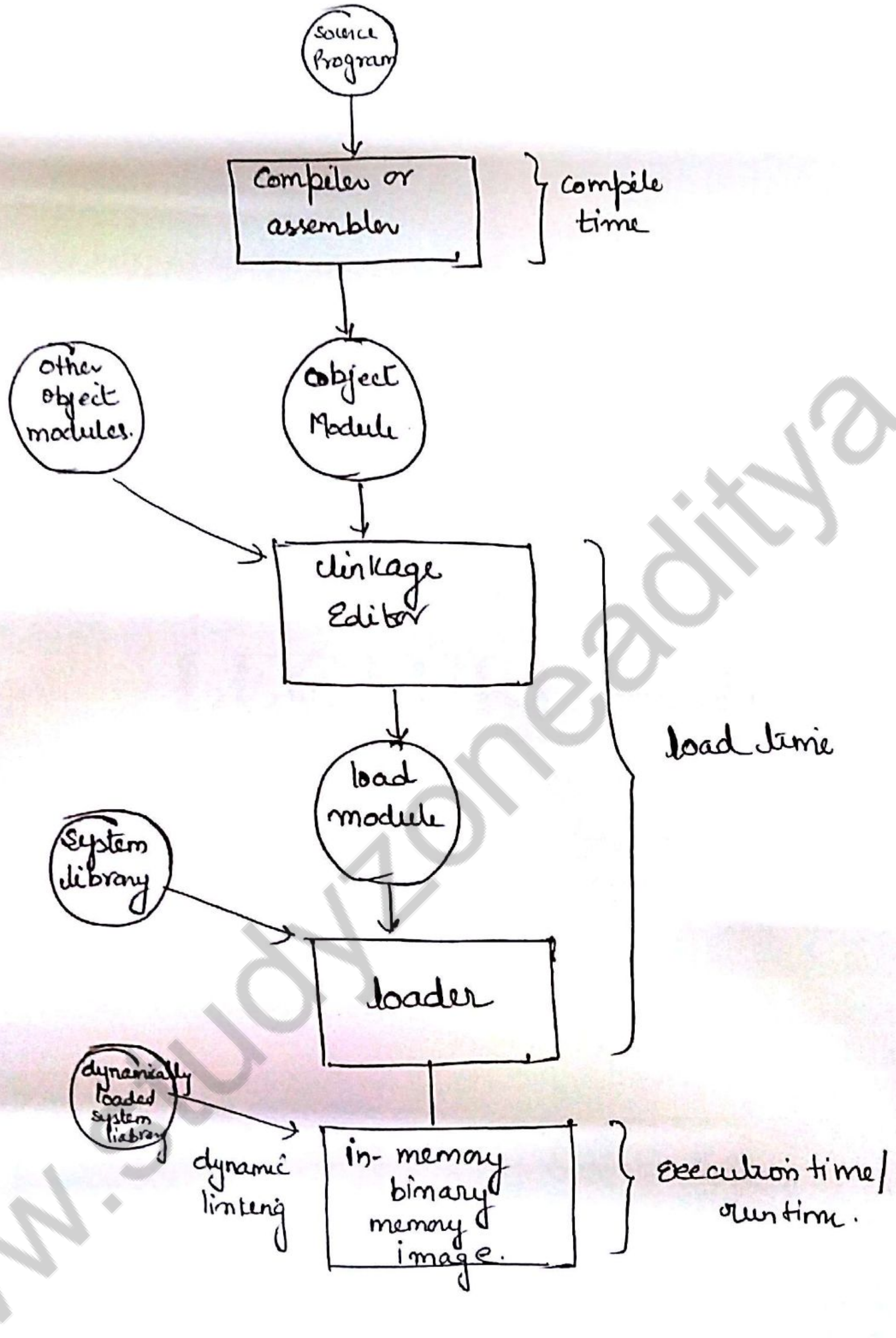
Address binding of instructions & data to memory addresses can happen in 3 different stages:-

→ Compile time :- If memory location known apriori, absolute code can be generated; must recompile code if starting location changes.

→ Load Time :- Must generate relocatable code if memory location is not known at compile time.

→ Execution time :- Binding delayed untill run time if the process can be moved during its execution from one memory segment to another. Need h/w support for address maps. (eg base limit registers & limit registers).

Multistep Processing Of a User Program.



Static vs Dynamic Loading

The choice b/w static & Dynamic loading is to be made at the time of computer program being developed. If you have to load your program statically, then at the time of compilation, the complete programs will be compiled and linked without leaving any external program or module dependency. The linker combines the object program with other necessary object modules into an absolute program, which also includes logical address.

If you are writing a Dynamically loaded program, then your compiler will compile the program and for all the modules which you want to include dynamically, only references will be provided & rest of the work will be done at the time of execution.

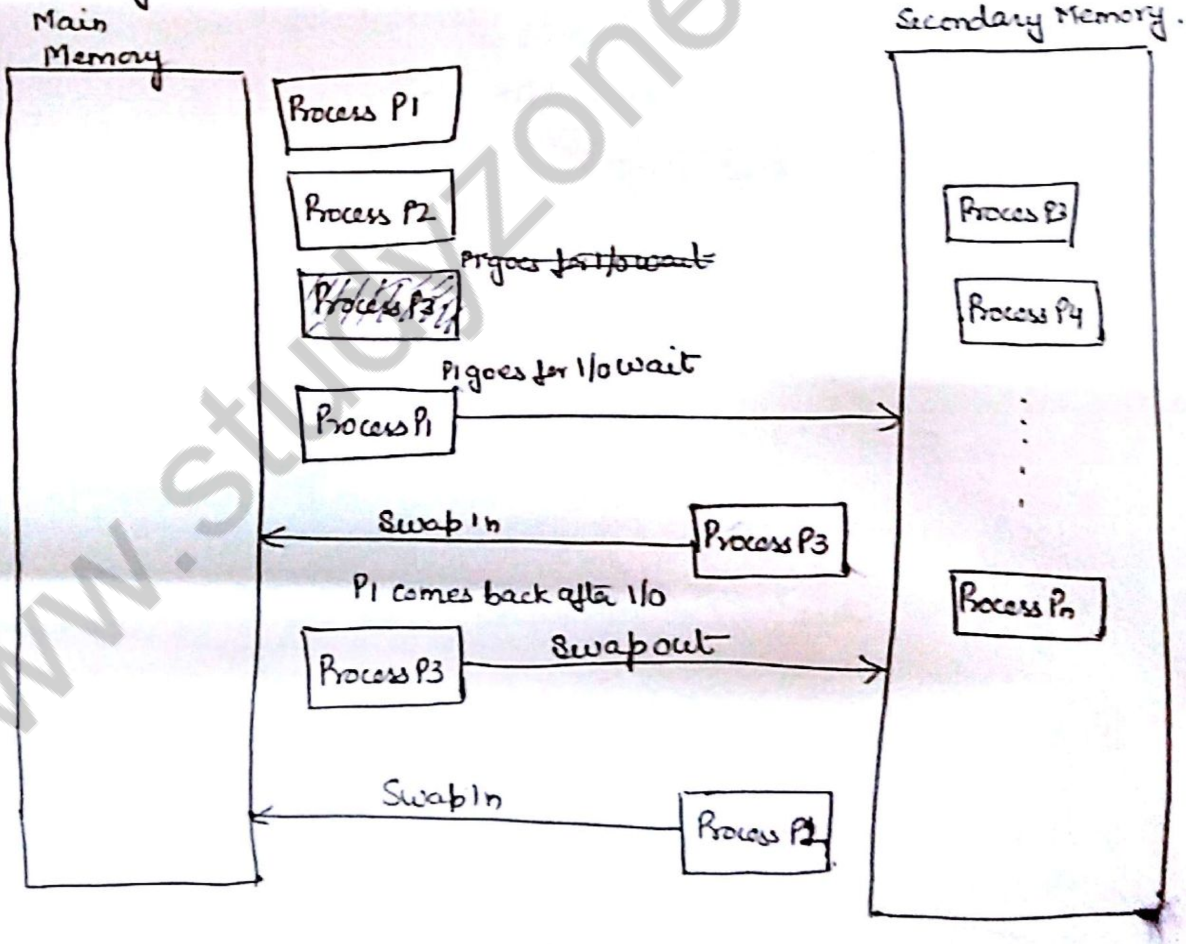
At the time of loading, with static loading, the absolute program (and data) is loaded into memory in order for execution to start.

If you are using dynamic loading, dynamic routine of the library are stored on a disk in relocatable form and are loaded into memory only when they are needed by the program.

Swapping

Swapping is a mechanism in which a process can be swapped temperoily 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.

Though performance is usually affected by swapping process but it helps in running multiple and big processes in parallel & thats the reason "Swapping is also known as a technique for memory compaction".



Single Partition Allocation

In this type of allocation, relocation-register scheme is used to protect user processes from each other, and from changing O.S. code and data. Relocation register contains value of smallest physical address whereas limit register contains range of logical address. Each logical address must be less than the limit register.

Multiple partition

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.

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 remains unused.

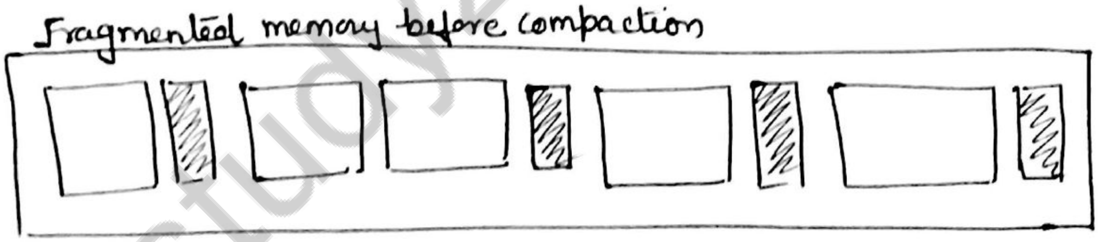
This problem is known as Fragmentation.

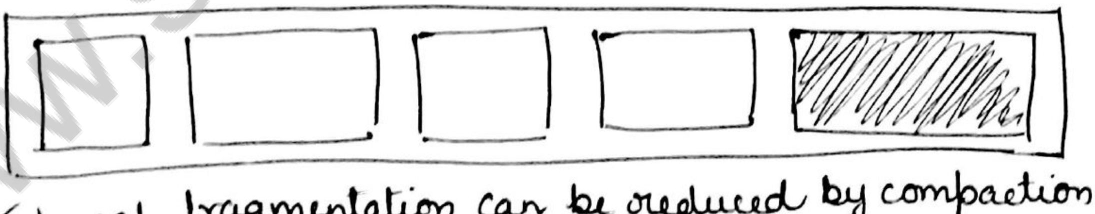
Fragmentation is of 2 types.

External Fragmentation :- 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.

Internal Fragmentation :- Memory block assigned to process is bigger. Some portion of memory is left unused, as it cannot be used by another process.

Fragmented memory before 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.

Paging:

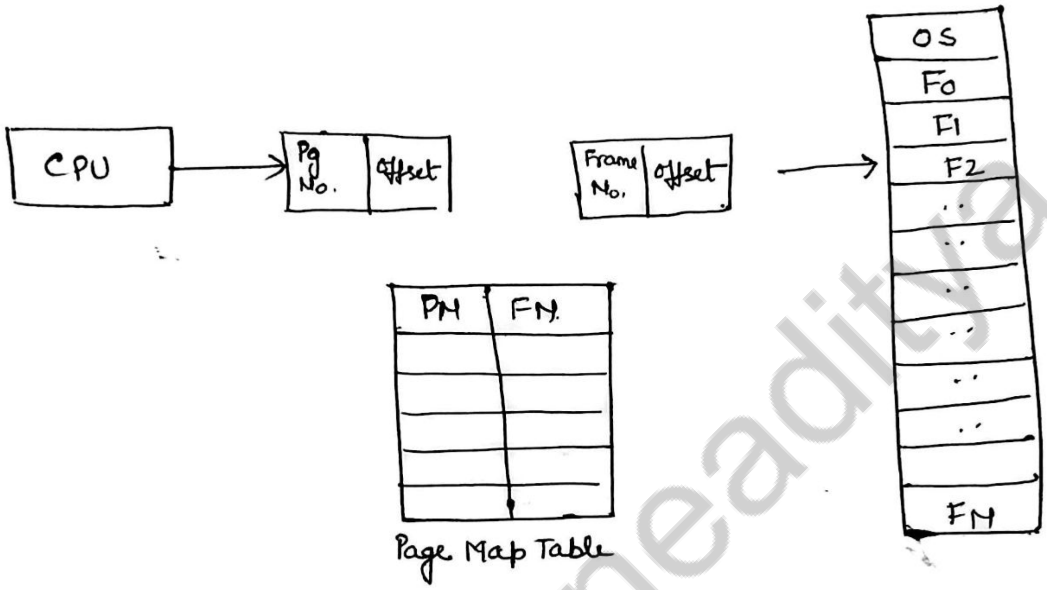
A computer can address more memory than the amount physically installed on the system. This extra memory is actually called virtual memory and it is a section of a hard that's set up to emulate the computer's RAM. Paging technique plays an important role in implementing virtual memory.

Paging is a memory management technique in which process address space is broken into blocks of same size called pages (size is power of 2, b/w 512 bytes and 8192 bytes). The size of the process is measured in the number of pages.

Similarly, main memory is divided into small fixed size blocks of (physical) memory called frames. and the size of a frame is kept the same as that of a page to have optimum utilization of the main memory and to avoid external fragmentati°.

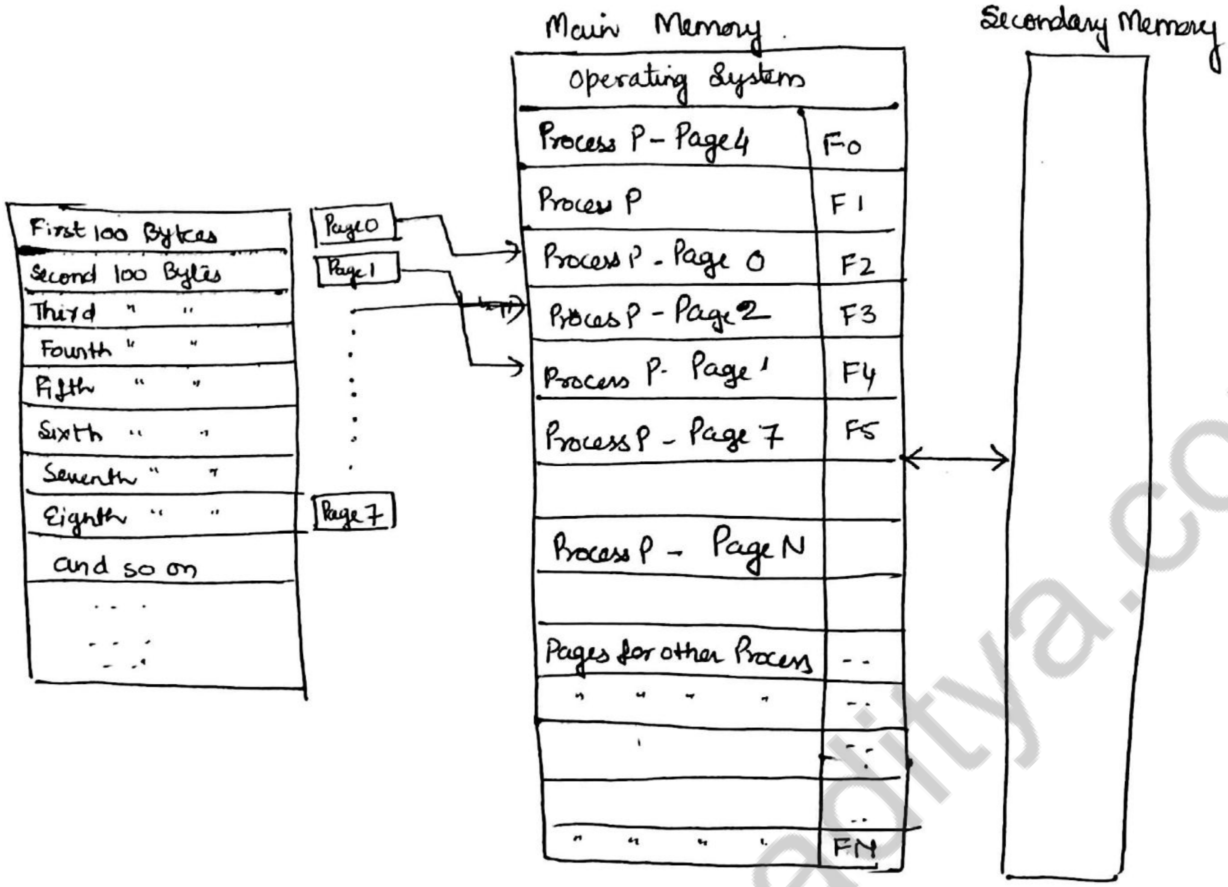
Page Map table

A data structure called page map table is used to keep track of the relation between a page map table is used to keep track of the relation between a page of a process to a frame in physical memory.



When the system allocates a frame to any page, it translates this logical address into a physical address and create entry into the page table to be used throughout execution of the program.

When a process is executed, its corresponding pages are loaded into the memory frames. Suppose you have a program of 8Kb but your memory can accommodate only 5Kb at a given pt. in time, then the page concept will come into picture. When a computer runs out of RAM, the OS will move idle, or unwanted page of memory to free up RAM for other processes & brings them back when needed by the program.



Paging Technique

Address Translation

Page address is called logical address and represented by page number and the Offset.

$$\boxed{\text{Logical Address} = \text{Page Number} + \text{page offset}}$$

Frame address is called physical address and represented by a frame number and the Offset.

$$\boxed{\text{Physical Address} = \text{Frame Number} + \text{Page offset}}$$

Advantages and Disadvantages of Paging

→ Paging reduces external fragmentation, but still suffer from internal fragmentation

→ Paging is simple to implement and assumed as an efficient memory management technique.

→ Due to equal size of the pages and frames, swapping becomes very easy.

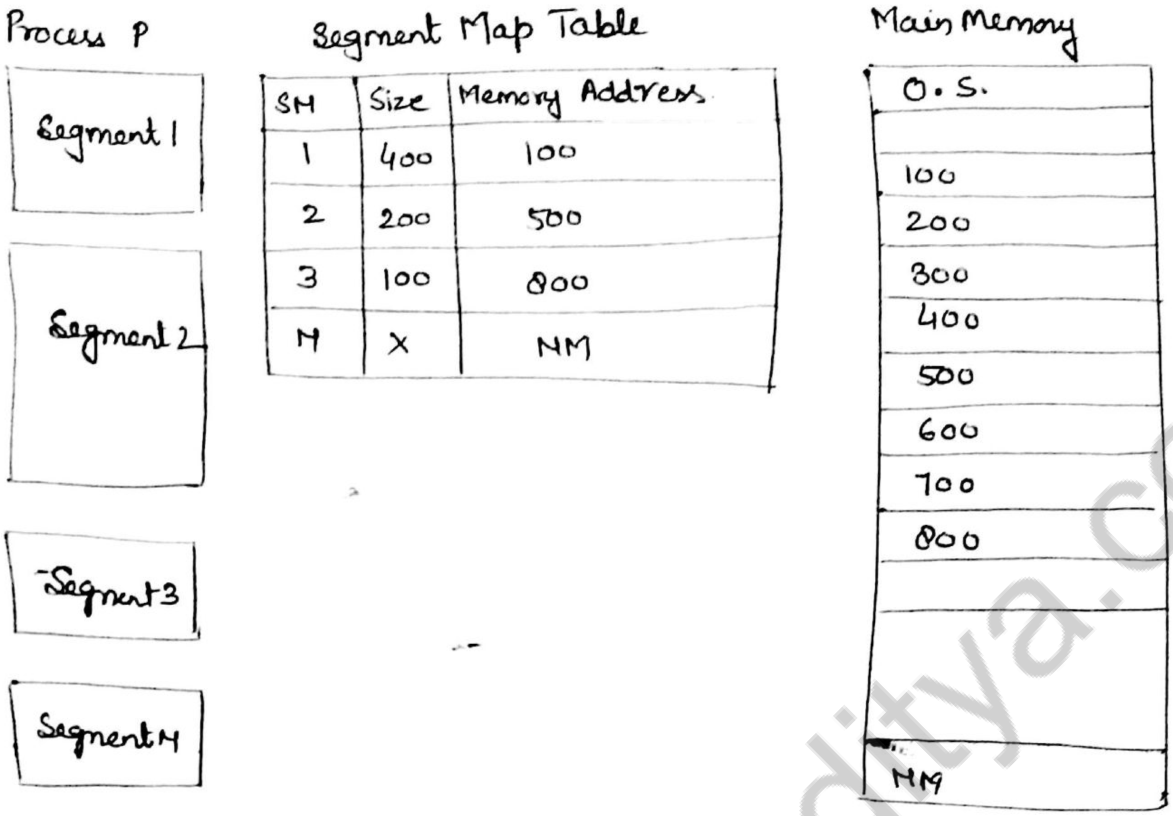
→ Page table requires extra money space, so may ~~mo~~ not be good for a system having small RAM.

Segmentation

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

When a process is to be executed, its corresponding segmentation are loaded into non-contageous memory though every segment is loaded into a contageous ~~memory~~ block of available memory.

Segmentation memory mgmt works very similar to paging but here segments are of variable length whereas in paging pages are of fixed size.



Segment Map Table

SN	Size	Memory Address
1	400	100
2	200	500
3	100	800
N	X	NM

A program segment contains the program's main memory, utility functions, data structure, and so on. The O.S. maintains a segment map table for every process and a list of free memory block along with segment numbers, their size & corresponding memory locations in main memory. For each segment, the table stores the starting address of the segment and the length of the segment. A reference to a memory location includes a value that identifies a segment & an offset.

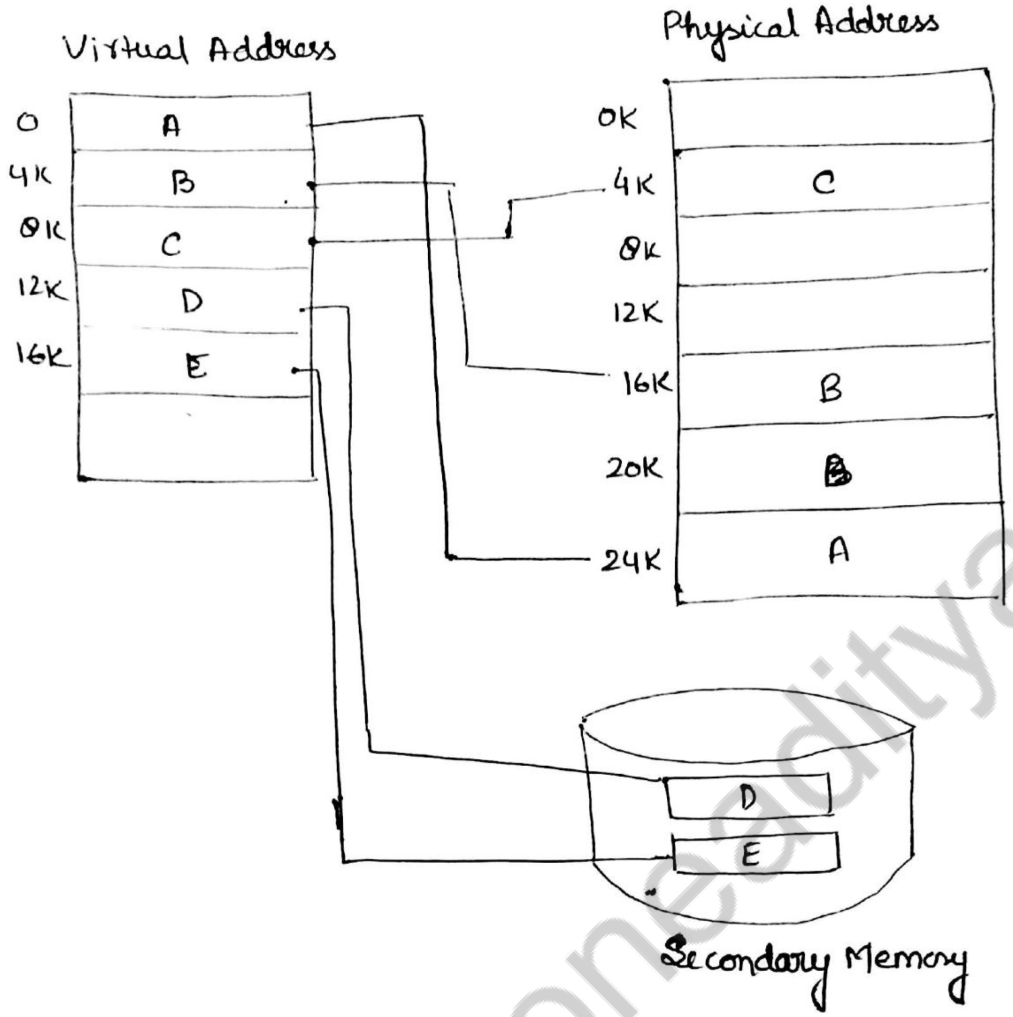
Virtual Memory

A computer can address more memory than the amount physically installed on the system. This extrat memory is actually called virtual memory and its as section of a hard disk that's set up to emulate the computer's RAM.

The main visible advantage of this scheme is that programs can be larger than physical memory. Virtual memory ~~and it is a section of a hard disk~~ serves two purposes. First, it allows us to entend the use of physical memory by using disk. Second, it allows us to have memory protection, because each virtual address is translated to a physical address.

There are several situations where entire program is not required to be loaded in the M.M.

Modern microprocessors intended for general-purpose use, a memory management unit, or MMU, is built into the h/w. The MMU's job is to translate virtual addresses into physical addresses. A basic example is given below :-



Virtual memory is commonly implemented by demand paging. It can also be implemented in a segmentation system. Demand segmentation can also be used to provide virtual memory.