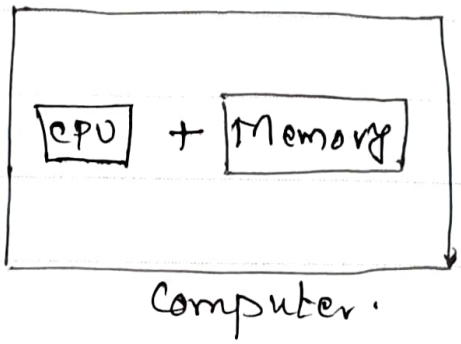
UNIT-III

Basic of Memory Managment

→ The capability of a computer depends on CPU and memory.



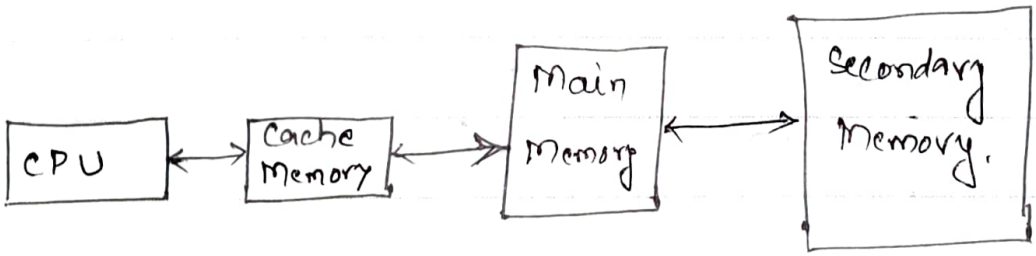
Three criteria as far as memory concern

→ Size (↑)

→ Access time (↓)

→ Per unit Cost (↓)

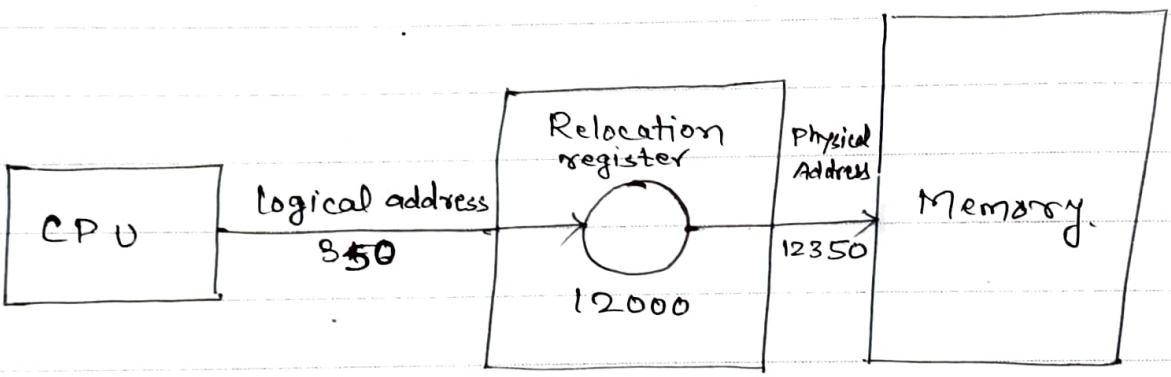
* Size is contradictory in nature with access time and per unit cost.



Logical address and Physical address

Parameter	Logical Address	Physical Address.
Basic	It is the virtual address generated by CPU	The physical address is a location in a memory unit
Address space	Set of all logical addresses generated by CPU in references to a program is refered as logical Address space	Set of all physical addresses mapped to the corresponding logical addresses is referred as physical address space.

Visibility	The user can view the logical address of a program.	The user can never view physical address of program.
Access	The user uses the logical address to access the physical address.	The user can not directly access physical address.
Generation	The Logical Address is generated by the CPU	Physical address is computed by MMU



MMU (Memory Managment Unit)

Address Binding

(i) Compile time
(ii) Load time
(iii) Execution time

→ If there is a process P1 and we want to execute that then first that process should be brought into main memory.

Memory Allocation Policy

(1) CONTIGUOUS POLICY :

In contigious Policy, executing process must be loaded entirely in main-memory in contigious fashion. It can be divided of two types :-

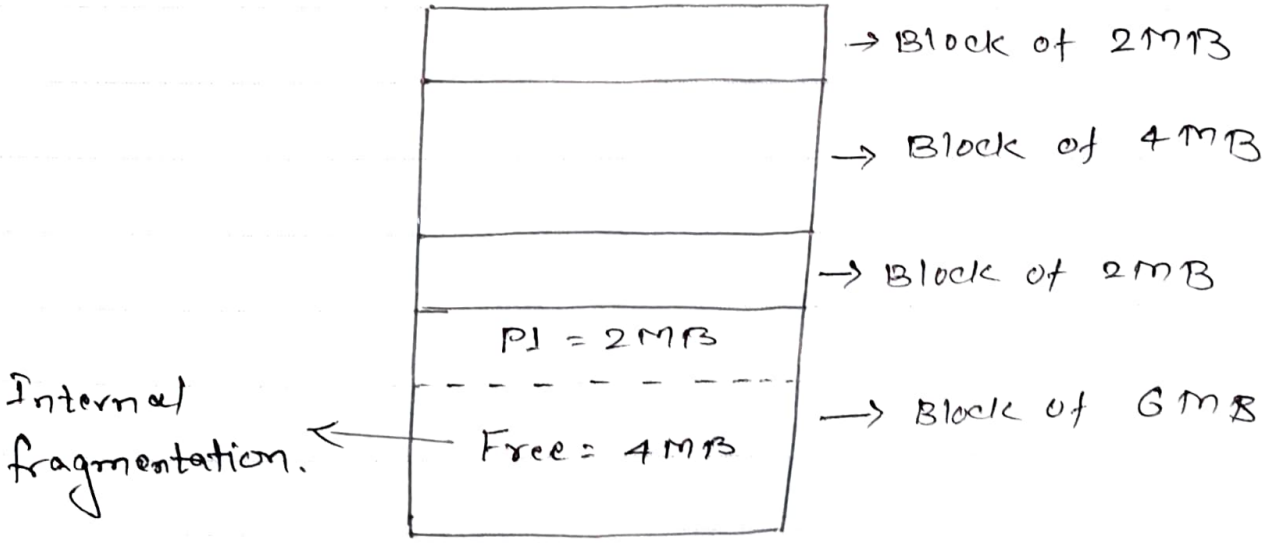
(a) Fixed Size Partitioning / Static Partitioning

→ Oldest and simplest technique.

→ In this scheme, first we divide the main memory into certain number of partitions (Non-overlapping). i.e number of partitions (Non-overlapping) is fixed.

→ Size of each partition may or may not be same.

→ Here partition are made before execution or during system configure.



Fixed Size Partitioning

Advantages:-

(i) Easy to implement

(ii) Little OS overhead.

Disadvantages:-

(i) Internal Fragmentation

(ii) External Fragmentation

(iii) Limit Process size

(iv) Limitation on degree of multiprogramming.

(b) Variable Size Partitioning / Dynamic Partitioning

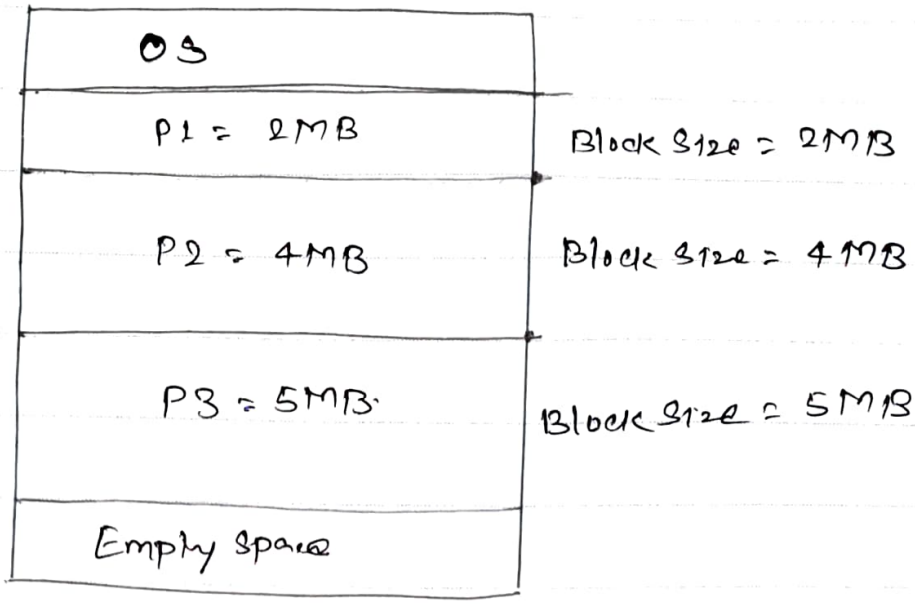
→ Initially RAM is empty and partitions are made during run-time according to process's need instead of partitioning during system configure.

→ 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 utilisation of memory.

→ No. of Partition in memory is not fixed and depends on the number of incoming process and main memory size.

→ Partition size = Process size.



Dynamic Partitioning.

Advantages:-

(i) No. internal fragmentation

(ii) No. restriction on degree of multiprogramming

(iii) NO Limitation on the size of the process.

Disadvantages :-

(i) Difficult implementation

(ii) External fragmentation (e.g. if P1 and P3 has completed and an empty space P4 is running then P5 of 6MB cannot be allocated in memory in above figure)

Memory Allocation Algorithm for Contigious Policy

(1) FIRST FIT :- Allocate the first hole that is big enough. Searching can start either at begining of the set of holes or where the previous first-fit search ended. We

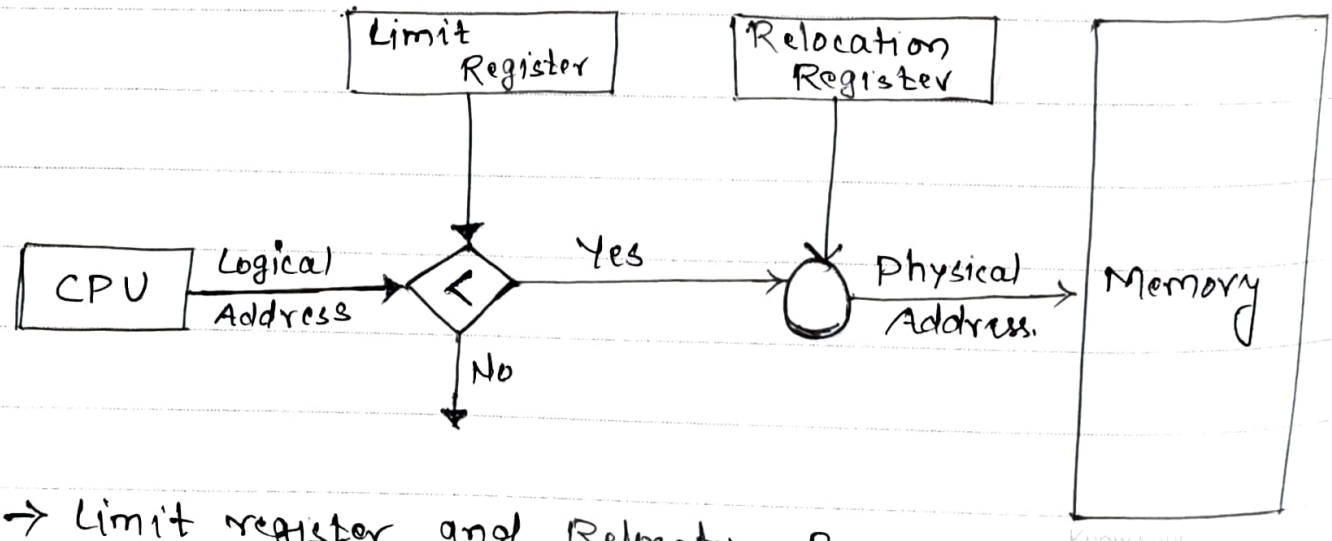
can stop searching as soon as we find a free hole that is large enough.

<2> BEST FIT :- Allocate the smallest hole that is big enough. We must search the entire list, unless the list is kept ordered by size. This startegy produces the smallest leftover hole (internal fragmentation)

<3> WORST FIT :-

Allocate the largest hole. Again we must search the entire list unless it is sorted by size. This strategy produces the largest leftover hole, which may be more useful than a smaller leftover hole from a best-fit approach.

MEMORY PROTECTION:



→ Limit register and Relocation Register exist in PCB of a process

→ Limit register store the number of instructions in process.

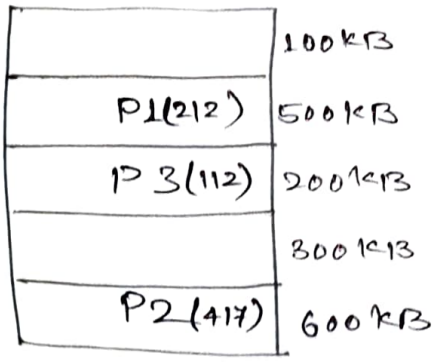
→ Relocation register store the base address.

Q: Given memory partitions of 100KB, 500KB, 200KB, 300KB and 600KB (in order). How would each of the first-fit, best-fit and worst fit algorithms place processes of 212KB, 417KB, 112KB, and 426KB (in order)? Which algorithm makes the most efficient use of memory.

Sol^n

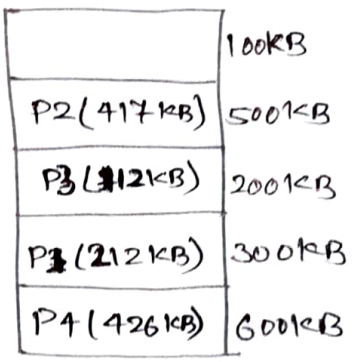
Let P1 = 212KB, P2 = 417KB, P3 = 112KB, and P4 = 426KB

First Fit

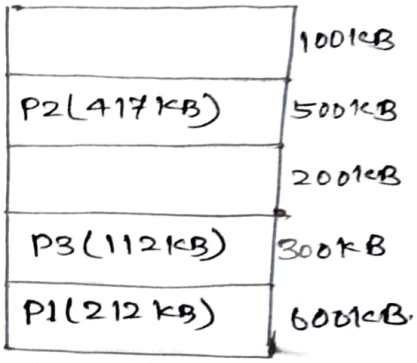


P4 cannot placed.

Best fit



Worst fit



P4 cannot be placed.

So Best fit is most efficient here.