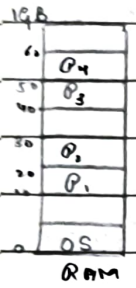
* "Memory Consist of array of bytes (or) words each with their own address".
* It is concerned with managing memory

__/__/____

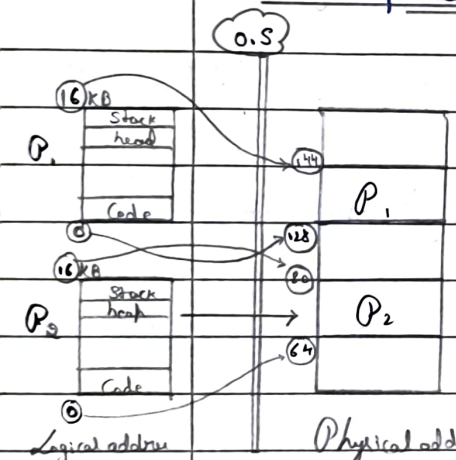
Memory Management Techniques:

Contiguous Memory Allocation.

* In Multi-programming environment, we have multiple Process in the main memory (Ready Q) (RAM) to keep the CPU Utilization high and to make Computer responsive to users.
* To Realize this Ice in Preference, however, we must keep several processes in the memory; that is, we must Share the main memory. As a result, we must manage main memory for all the different Processes.



* Isolation & Security of Memory Processes: (How it is Provided?)
  ① P_1 & P_2 will be allotted with base at RAM called 'Physical Memory'.
  1) P_1 → B: 128 and offset: 16
  2) P_2 → B: 64 and offset: 16
  if P_2 want to access some address of Random value (eg: 129)
  (0)
  ∴ 0 + 129 => 129 (addr)
  * actual physical addr => 64 + 129 = 193 (not belong to P_2)
  Physical addr ∴ it will thrown an error Saying out of bound.



Logical (LA) Address V/s Physical (PA) Address Space:

1) Logical Address:

* An address generated by CPU.
* L.A is basically the address of an instruction/data used by a Process
* user Can access L.A of a Process user have indirect access to 'PA' via 'LA'
* L.A → doesnot Exist physically → 'Virtual Address'.
* The Set of all logical addresses that are generated by any Program is suffixed to as 'Logical Address Space'.
* Range: 0 to max.

<('f' address → they are logical address)>

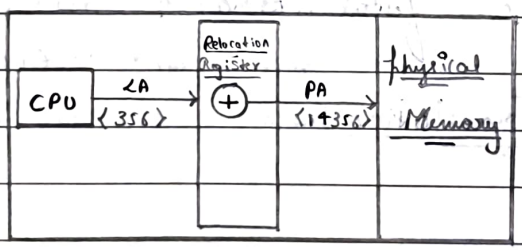
__/__/____

ii) Physical Address:

* An address loaded into the 'memory-addr register' of physical memory
* User can never access 'PA' of Program
* 'PA' is in the memory unit. It's location in the 'main memory physically'.
* 'PA' can be accessed by user indirectly but not directly.
* The set of all physical Address corresponding to L.A is commonly known as 'physical Address Space'.
* It is computed by 'Memory Management Unit (MMU)'
* Range: (R+0) to (R+Max) for base value 'R'.

⊙ MMU: The runtime mapping from virtual to physical address is done by a hardware device called the MMU.

* The user's Program mainly generates the logical Addr, & user thinks that Program is running in this LA, but Program is mainly needs physical memory in order to complete its Execution.



→ How O.S Manages the isolation & protect (Memory Mapping & Protection):

a) OS Provides this Virtual Address Space (VAS) Concept.

b) To Seperate memory Space, we need the ability to determine the range of legal (only allotted P.A) address that the Process may access and to ensure that the Process can access only these legal address.

c) The relocation register contains value of smallest physical address (Base address [R]); the limit register contains the range of logical address (Eg: Relocation = 10000 & and limit = 74600)

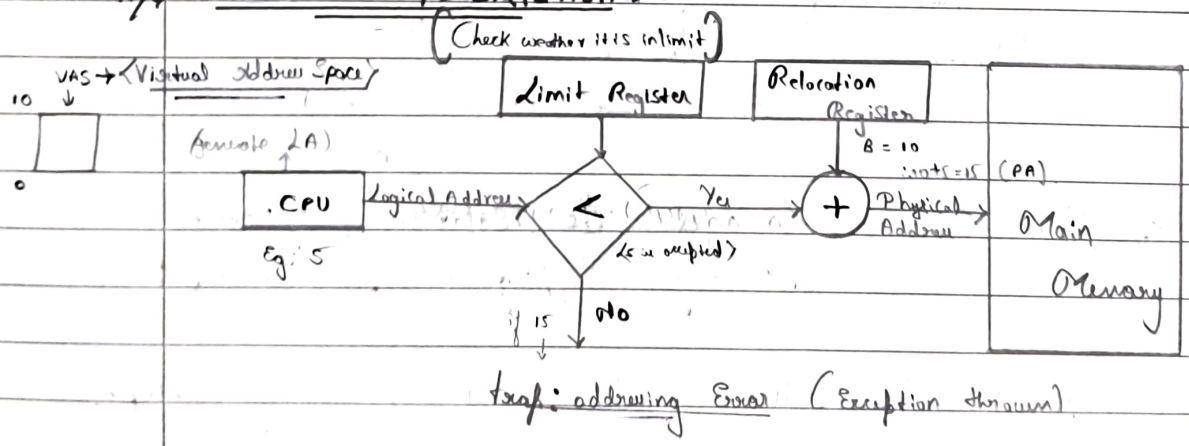
d) Each logical address must be less than the limit register.

e) MMU maps the logical address dynamically by adding the value in the relocation register.

f) When CPU Scheduler selects a process for execution, the dispatcher loads the relocation & limit registers with the correct values as part of context switch. Since every address generated by the CPU (logical address) is checked against these registers, we can protect both OS & other user's programs and data from being modified by running process.

g) Any attempt by a program executing in user mode to access the OS memory or other user's memory results in a trap in the OS, which treat the attempt as a 'fatal Error'.

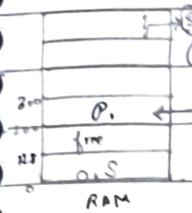
h) **Address Translation :-**



__/__/____

*) Allocation Method on Physical Memory:
   a) Contiguous Allocation.
   b) Non-Contiguous Allocation.

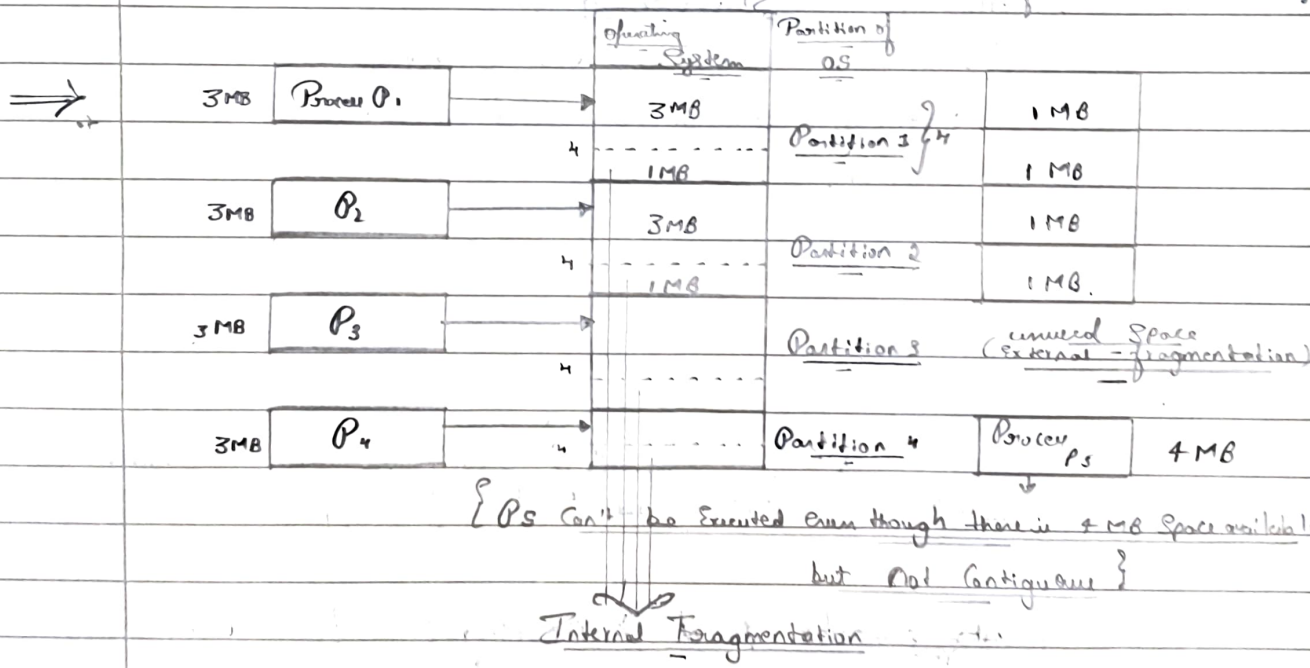
I) Contiguous Allocation :-



* In this Scene, each Process is Contained in a Single Contiguous block of memory.
* Continuous Memory allocation.
* P_1 is not Partitioned & alloted.

a) Fixed Partitioning :-

* The main memory is divided into Partitions of equal (or) different Sizes.



b) Limitations :-

Internal Fragmentation : If the Size of Process is lesser than the total Size of Partition then Some Size of Partition gets wasted and remain unused. This is wastage of memory and Called 'Internal Fragmentation'

__/__/____

2) External Fragmentation :- The total unused Space of various Partitions Cannot be used to load the Process even though there is Space available but not in contiguous form.

3) Limitation on Process Size :- If the Process Size is larger than the Size of maximum Sized Partition then that Process Cannot be loaded into the memory. Therefore, a limitation Can be imposed on the Process Size that is it Cannot be larger than the Size of largest Partition.

4) Low degree of Multiprogramming :- In fixed Partioning, the degree of multiprogramming is fixed & very low because of the Size of Partitions Cannot be varied according to the Size of Processes.

b) Dynamic Partitioning :

*In this Sequence, the Partition Size is not declared initially. It is declared at the time of Process loading.

			O.S	Partition of O.S
⇒ 5MB	Process P_1	→	P_1 (5MB)	Partion 1
2MB	Process P_2	→	P_2 (2MB)	Partion 2
3MB	Process P_3	→	P_3 (3MB)	Partion 3
4MB	Process P_4	→	P_4 (4MB	Partion 4

{ Process Size = Partition Size }

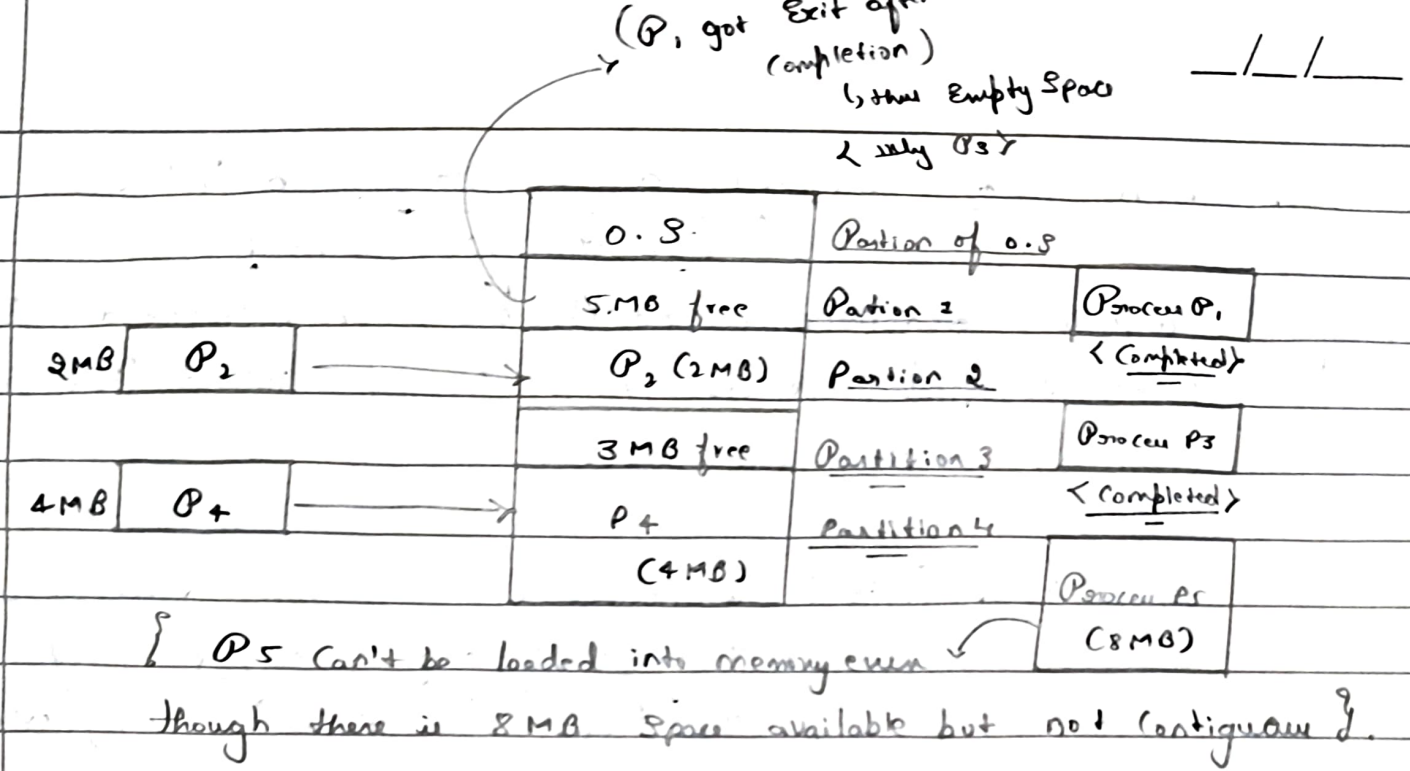
① Advantages over fixed Partitioning :

1) No internal fragmentation 2) No limit on Size of Process
3) Better degree of Multi-Programming.

② Limitation :-

External fragmentation.

__/__/____


(P1 got Exit after completion)
(thus Empty Space by P3)
0.8
Portion of 0.8
5MB free
Portion 1
Process P1
<Completed>
2MB
P2
P2 (2MB)
Portion 2
3MB free
Partition 3
Process P3
<Completed>
4MB
P4
P4 (4MB)
Partition 4
Process P5
(8MB)
{ P5 Can't be loaded into memory even though there is 8MB Space available but not Contiguous }.