*)→ What is Virtual Memory? // Demand Paging // Page faults.

⊙ **Virtual memory**: It is a dechnique that allows the Execution of processes that are not Completely in the memory. It Provides user an illusion of having a very big main memory. This is done by treating a part of 2° memory as the main memory (Swap-Space).

* Advantage of this is; Programs Can be larger than Physical Memory.

* Programmer is Provided Very large Virtual memory when only a Small physical memory is available.

_/_/__

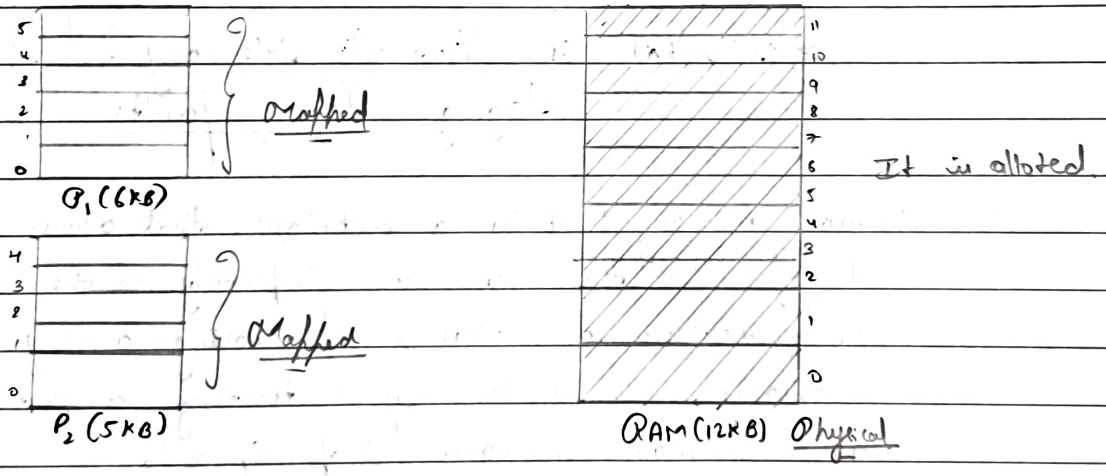
→ It is required that instructions must be in physical memory to be Executed. But it limits the Size of a Program to the Size of physical memory. In fact, in many Cases, the entire Program is not needed at the Same time. So, we want an ability to Execute a Program that is only Partially in memory would give many benefits:

ⓐ A Program would no longer be Constrained by the amount of physical memory that is available.

ⓑ Because each user Program Could take less physical memory, more Programs Could be run at the Same time, with a Corresponding 'Ies in CPU utilization and throughput'.

ⓒ Running a Program that is not entirely in memory would benefit 'both the System and the user'.

⇒



① If P3 (5 KB) arrives → we require 5 frame → but only 1 is available

② V.M → comes into action when we have limited physical memory.

⊛ So, we load only needed Pages of Process like P1's 2 Pages P2's 3 Pages thus we get enough empty frames. To load P3, P4 Processes.

⊛ i.e; it keeps the not-needed Pages of Process in a block of hard disk ⊟ i.e Reserved & Ready to use.

__/__/____

(after trap generation)

- 3rd Page (after analyzing 2 Pages) → will be loaded to RAM by the request by freeing a frame i.e. already Completed its work. Not-needed Program are Stored in 'Swap (Space) back' at the backend.

- Swap Space → Small Space at disk.

→ 'Demand Paging' (DP) is a popular method of virtual memory management.

★ In DP, the Pages of a Process which at least used, get Stored in 2° memory.

★ A Page is Copied to the main memory when its demand is made, or Page faults occur. There are Various 'Page replacement algorithum' which are used to determine the Pages which will be replaced.

★ Rather than Swapping the entire Process into memory we use '**Lazy Swapper**'. A lazy Swapper never Swaps a Page into memory unless that Page will be needed.

How Demand Paging works?

Pager — Similar to "Swapper" (entire process)

a) When a Process is to be Swapped-in, the Pager guesses which Pages will be used.

b) Instead of Swapping in a Whole Process, the Pager brings only those Pages into memory. This, it avoids reading into memory Pages that will not be used anyway.

c) Above way, OS ↓ces the Swap time and the amount of physical memory needed.

d) The Valid-invalid bit Scheme in the Page table is used to distinguish b/w Pages that are in memory and that are on the disk.

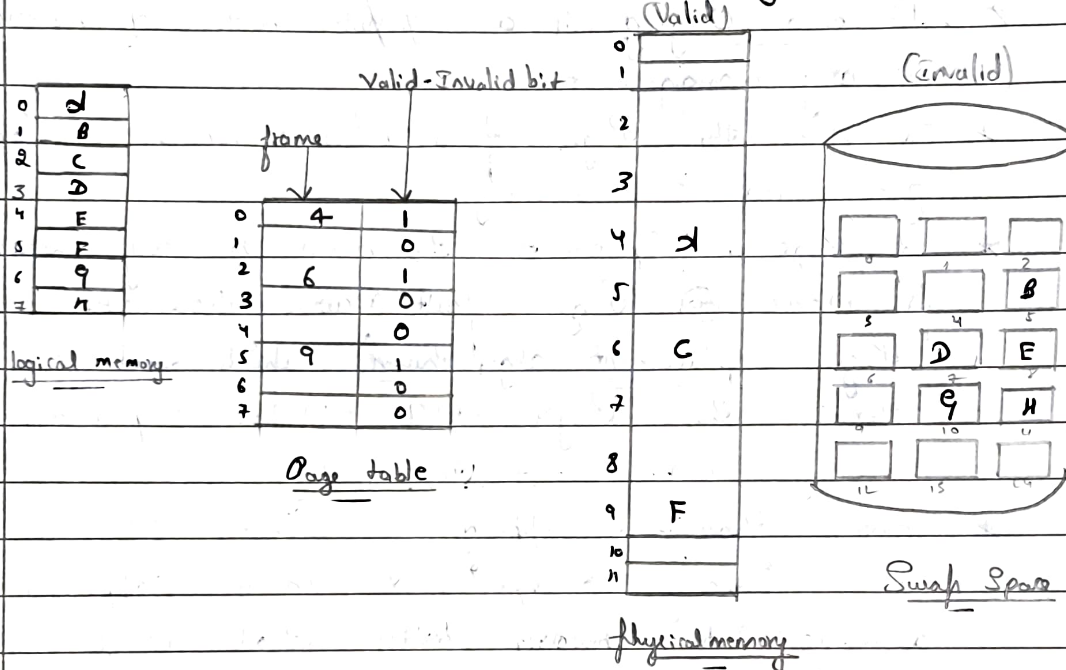
(RAM + Swap Area) => Virtual Memory.

__/__/____

i) Valid-Invalid bit '1' => assosicated Page is both legal & in memory

ii) Valid-Invalid bit '0' => Page either is not valid (not in LAS of Process) or is valid but is Currently on disk.

⊙ Page table when Some Pages are not in memory



0	d
1	B
2	C
3	D
4	E
5	F
6	g
7	h

logical memory

	frame	Valid-Invalid bit
0	4	1
1		0
2	6	1
3		0
4		0
5	9	1
6		0
7		0

Page table

e) If a Processe never attempts to access Some invalid bit Page, the Process will be Executed Succesfully without even the need of Pages Present in the Swap Space.

f) what happens if the Process tries to access a Page that was not bought into memory, access to a Page marked Invalid Causes 'Page faults'. Paging hardware noticing invalid bit of a demanded Page will Causes 'a trap to the O.S'.

Procedure to handle Page faults :-

i) Check an internal table (in PCB Process) to determine whether the reference was Valid or an invalid memory access.

<generate addr that does not Exist>

ii) If ref. was invalid Process throws Exception. If ref. is Valid, Pager will Swap-in the Page.

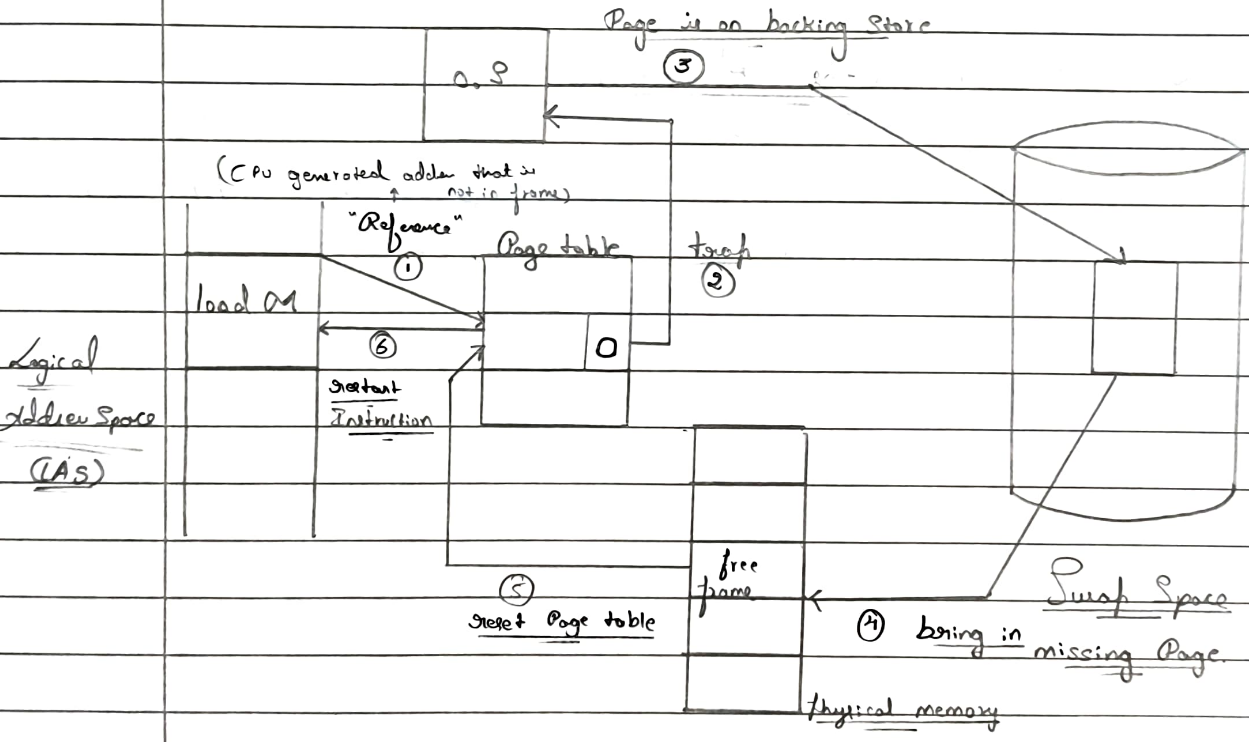
iii) we find a free-frame (from free-frame list)

iv) Schedule a disk Operation to read the desired Page into the newly allocated frame.

v) when disk read is Complete, we modify the Page table that, the Page is now in memory.

vi) Restart the instruction that was interrupted by the trap. The Process Can' now access the Page as through it had always been in memory

Steps in Handling a Page faults



⊙ Advantages of VM :-

1) degree of multi-Programming will be ↑ced
2) user Can run large apps with less Physical memory.

__/__/__

*) Pure Demand Paging :

1) In Extreme Case, we Can Start Executing a Process with no Pages in memory. when OS Sets the instruction Pointer to the first instruction of the Process, which is not in the memory. The Process immediately faults for the Page and Page is brought in the memory.

2) Never bring a Page into memory until it is required.

*) we use 'locality of reference' to bring out reasonable Performence from demand Paging.

*) Disadvantages of VM :

1) System Can become Slower as Swapping takes place
2) 'Thrashing' may occur.