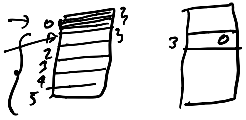
DEMAND PAGING IN VIRTUAL MEMORY SYSTEMS

- Loading entire executable code or all the pages at once during program execution code has certain disadvantages
- A better strategy would be to load pages into the physical memory only when they are needed. This technique is called demand paging
- With demand Paging, pages are only loaded when they are demanded during execution.
- Similar to Paging with swapping. We swap in and out the whole process with swapper
- In demand paging, we swap in and out the pages with Pager

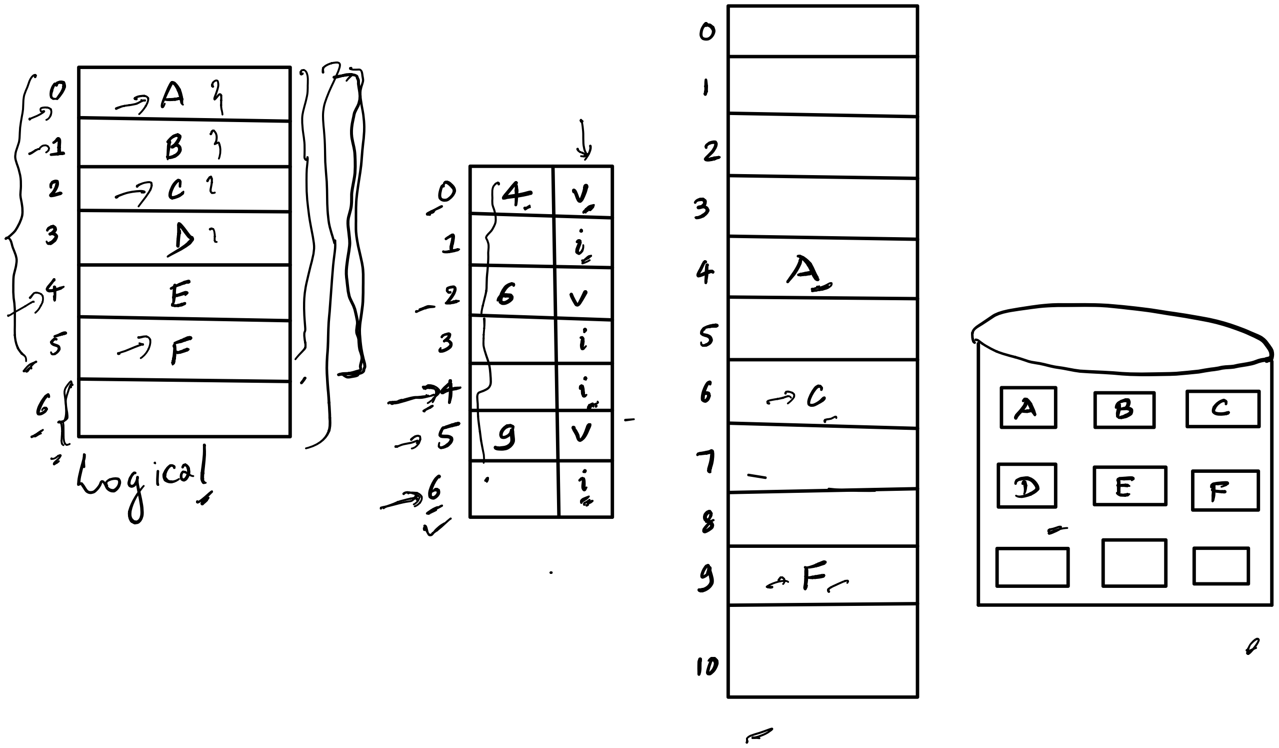
HOW DEMAND PAGING WORKS?

- When the process has been created(ready state), the pager will load some pages for the process in memory depending on some algorithms(frame allocation algorithms)
- When in running state, the CPU will start executing the instructions.
- Since initially only some pages were loaded, two cases are possible for the pages associated with the instruction



    - The page exists in the main memory
    - The page doesn't exists in the main memory but exists on the disk. This is also called as page fault.
- In case of page fault, the kernel(pager) needs to bring the page into the main memory from the disk and restart the instruction.
- How does the kernel know about page fault?
- An extra bit called valid/invalid bit is kept with the page table

- The Paging hardware(MMU) while accessing the page table will notice that the invalid bit is set and it will cause a trap to the kernel.



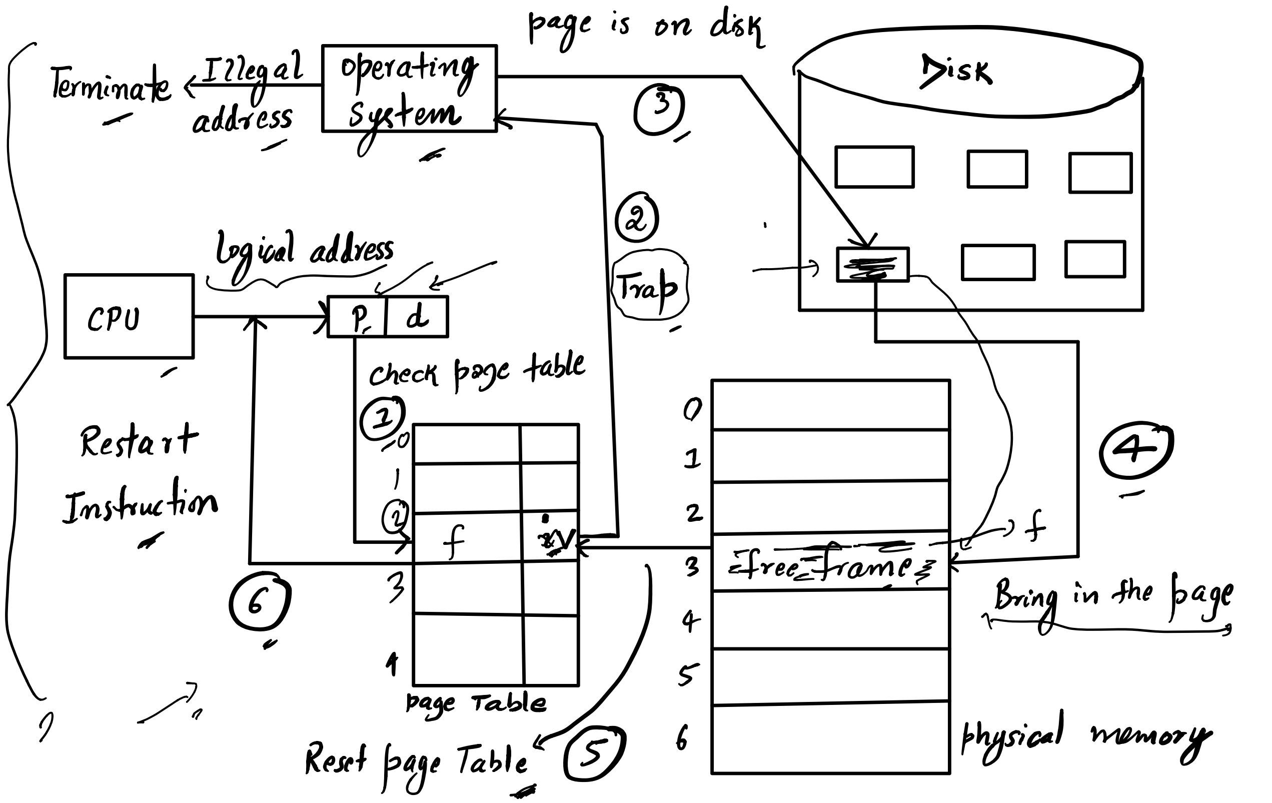
- If the valid/invalid bit is set to valid, it means that the page is legal and is in the memory
- If the valid/invalid bit is set to invalid, there can be two possibilities
    - The page is not legal - not in logical address space of process
    - The page is legal but it is on the disk and not in the main memory

HOW PAGE FAULT IS HANDLED BY THE KERNEL

0	0
1	1
2	0
3	1
4	

- The process is first interrupted by the CPU and its registers values are saved to the PCB
- First check if the the page is within the logical address space of the process using an internal table that is kept in PCB
- If the page is not valid, the kernel will terminate the process.
- If the page is valid, the kernel will find a free frame from the list of free frames. What happens if no frames are free?
- The kernel will then schedule a disk operation to read the desired page into the allotted frame

- When the disk operation has completed, the kernel modified the page table to indicate that the page is in memory
- The kernel restarts the instruction that was interrupted


page is on disk
Terminate
Illegal address
operating System
Disk
3
2
Trap
logical address
CPU
p
d
check page table
1
0
1
2
3
4
f
iv
page Table
Restart Instruction
6
Reset page Table
5
0
1
2
3
4
5
6
free frame
f
4
Bring in the page
physical memory