Memory Management
Lecture 7

Virtual Memory

Minakshi R.

Operating System Concepts 8th edition silberschatz Galvin

Need Of Virtual memory

- Paging & Segmentation need the entire process to reside inside the memory.
- Multi-programming: need more process should be accommodated in the memory.
- Programmer writes a program that is too large to fit in the memory.

Is that is necessary to have a whole program in the main memory before execution?

- Process are not required entirely in the main memory
- Only certain portion of the program is needed for execution
- While declaring the variable large no. of memory allocation is done that get wasted
- If a process is divided into several portion and the required portion is loaded in the memory for execution.
- So more No. of process can be accommodated in the main memory.

Overlays

- An overlay is a portion of a process.
- The overlays are stored in the disk
- The program containing overlay is called overlay structured program
- The required overlay is swapped in & later out when the memory is full.
- Swapping is done by the system
- Overlays are created by programmer & its difficult
- This method is obsolete

Virtual Memory

- A programmer is relieved from the tight constraints of memory size.
- A large process will be accommodated in the memory
- The implementation of virtual memory require hardware & software support.
- The software implementation of virtual memory system is known as virtual memory handler
- The logical address is known as virtual address
- The logical address space is known as virtual address space
- The virtual memory system requires only those pages or segments of a process in the memory that are needed at the time of execution.
- So there will be space for loading components of other processes. This approach is known as demand loading of process component.

Demand Loading of Process Component

- In the virtual memory system , only required component are loaded first in the memory
- The thumb rule of demand loading is

Never load a component of a process unless it is needed

- The component of the process that are present in the memory are called as **resident set**
- The execution of the process will be smooth if the logical address generated by the processor is in the resident set of the process

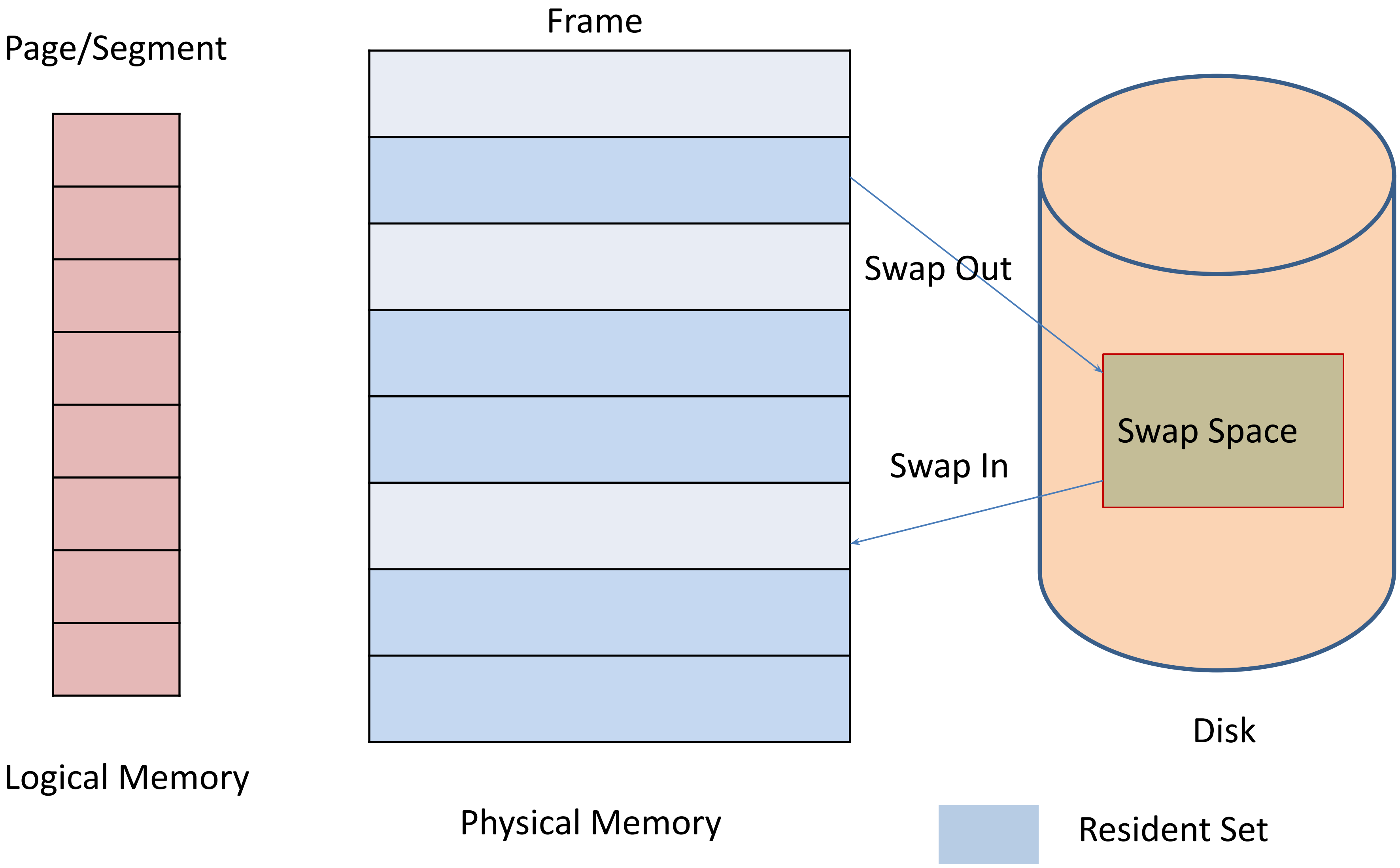
Demand Loading of Process Component

- If the processor generates a logical address that is not found in the memory, a memory access fault interrupt is generated.
- The process being executed is put in blocked state
- To resume the execution the component need to be swapped in to the memory
- For this the O.S issue a disk I/O read operation.
- Once its done the control is given to the O.S and the blocked state process is put back to the ready state.

Issues: Demand Loading of Process Component

- How will one recognize which component is in the memory & which one is not?
- How many processes will be resident in the memory (related to degree of multi-programming)
- How much main memory is allocated to the process.
- When no free space in memory and a component to be stored in memory then?
- The Virtual memory system realizes a huge memory only due to the hard disk. For this purpose a separate space called swap space is reserved in the disk.
- Swap space management: swap space required a lot of management so that the virtual memory system works smoothly.

Swap Space



Page/Segment

Frame

Swap Out

Swap Space

Swap In

Logical Memory

Physical Memory

Disk

🟦 Resident Set

Demand Paging

- Virtual Memory system is implemented using paging or segmentation
- Here paging is discussed
- To understand demand loading is replaced by demand paging
- In demand paging: only pages that are needed at any instant of the time of execution is loaded.
- It results in efficient utilization of memory
- In demand paging , an entire process is not swapped in or swapped out
- A lazy swapper, loads only those pages that are needed.
- The swapper term is used for swap in & swap out
- Here page term is used , so page in & page out.

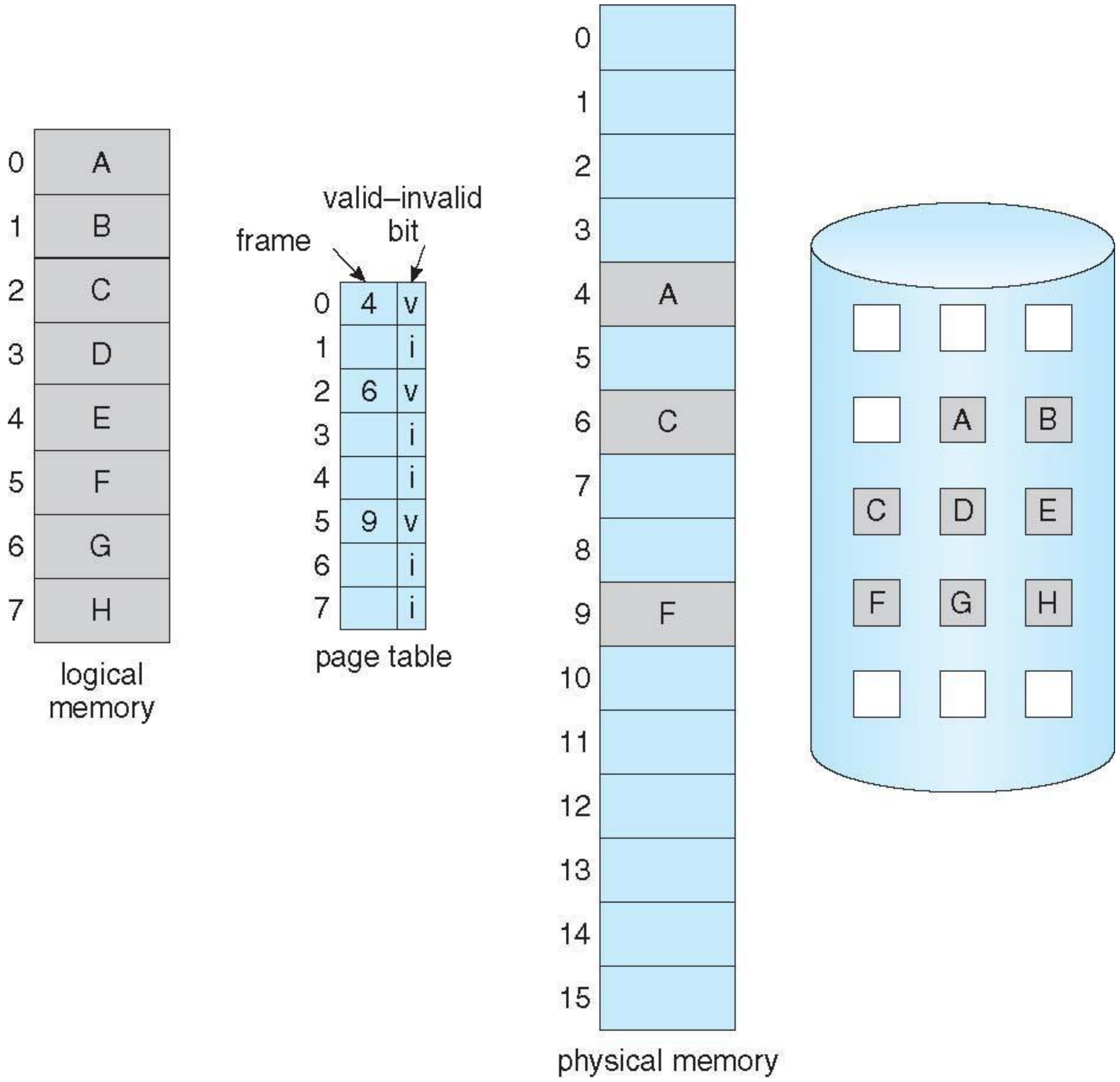
Demand Paging Problem

1. **How the demand paging recognize whether a page is present in memory or not**

Solution:

- The page table with valid or invalid bits can be used for this purpose.
- **Valid bit (1):** Contain the frame address of the page.
- **Valid bit (0):** Will not contain the frame address of the page.

Valid-Invalid Bits

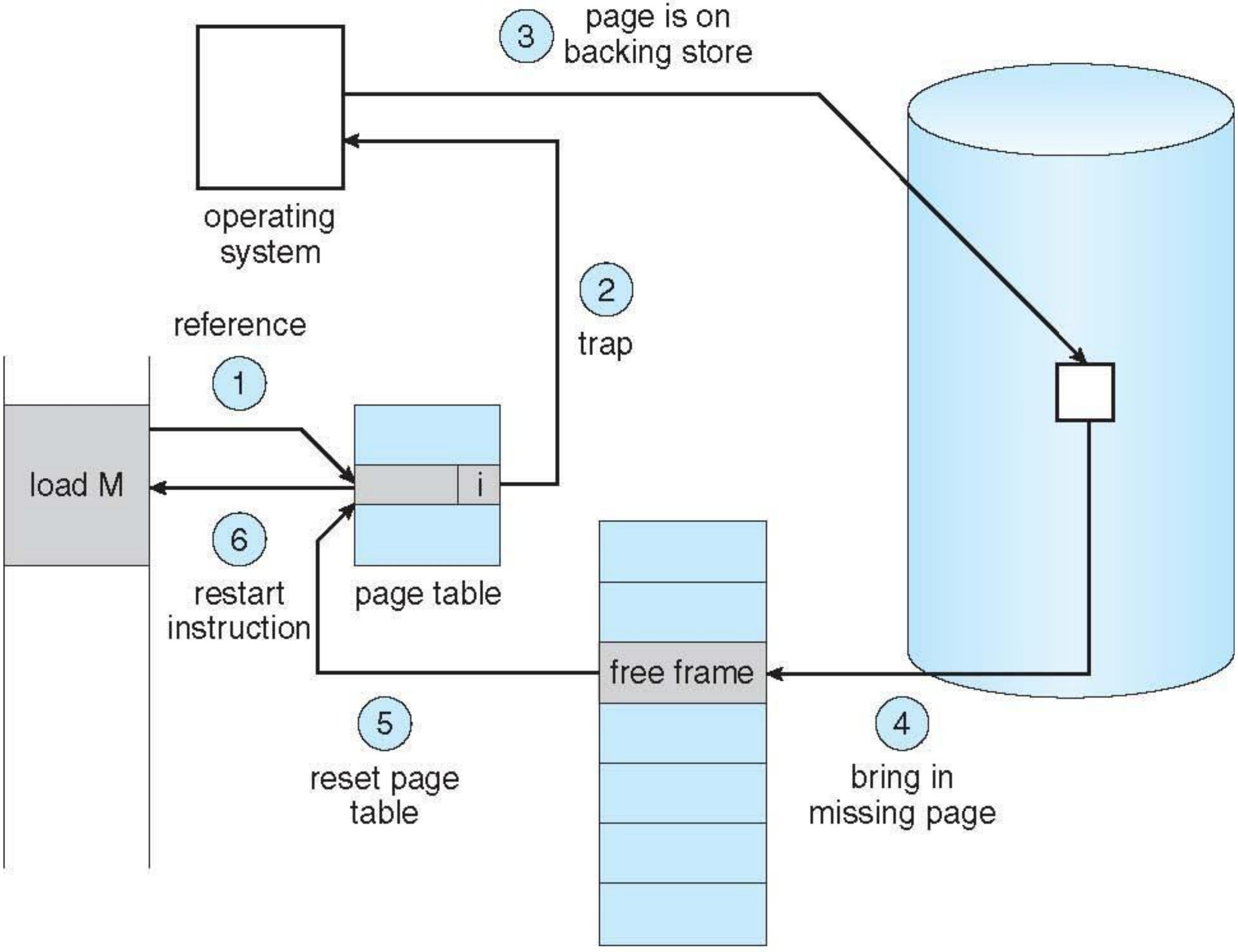


Demand Paging Problem

2. When a process execution does not get a page in the memory

- A situation will occur in demand paging when the page referenced is not present in the memory. This is known as a **page fault.**
- The page fault must be noticed by the system & be served appropriately.
- For a page-in operation, a free frame should be available.
- If the free frame is there, a disk operation to read the desired page is initiated.
- The page table entry is also updated as valid.

Steps in Handling a Page Fault



Stages in Demand Paging

1. Trap to the operating system
2. Save the user registers and process state
3. Determine that the interrupt was a page fault
4. Check that the page reference was legal and determine the location of the page on the disk
5. Issue a read from the disk to a free frame:
   1. Wait in a queue for this device until the read request is serviced
   2. Wait for the device seek and/or latency time
   3. Begin the transfer of the page to a free frame
6. While waiting, allocate the CPU to some other user
7. Receive an interrupt from the disk I/O subsystem (I/O completed)
8. Save the registers and process state for the other user
9. Determine that the interrupt was from the disk
10. Correct the page table and other tables to show page is now in memory
11. Wait for the CPU to be allocated to this process again
12. Restore the user registers, process state, and new page table, and then resume the interrupted instruction

THANK YOU