OPERATING SYSTEM
IT-42033

CHAPTER – 9
Q&A

9.11

- FIFO: Find the first segment large enough to accommodate the incoming segment. If relocation is not possible and no one segment is large enough, select a combination of segments whose memories are contiguous, which are “closest to the first of the list” and which can accommodate the new segment. If relocation is possible, rear range the memory so that the first N segments large enough for the incoming segment are contiguous in memory. Add any leftover space to the free-space list in both cases.
- LRU: Select the segment that has not been used for the longest period of time and that is large enough, adding any leftover space to the free space list. If no one segment is large enough, select a combination of the “oldest” segments that are contiguous in memory and that are large enough. If relocation is available, rearrange the oldest N segments to be contiguous in memory and replace those with the new segment.

9.12

- A page fault occurs when an access to a page that has not been brought into main memory takes place. The operating system verifies the memory access, aborting the program if it is invalid. If it is valid, a free frame is located and I/O is requested to read the needed page into the free frame. Upon completion of I/O, the process table and page table are updated and the instruction is restarted.

9.24

- a. Stack—good.
- b. Hashedsymboltable—not good.
- c. Sequential search—good.
- d. Binary search—not good.
- e. Purecode—good.
- f. Vector operations—good.
- g. Indirection—not good.

9.29

- The costs are additional hardware and slower access time. The benefits are good utilization of memory and larger logical address space than physical address space.

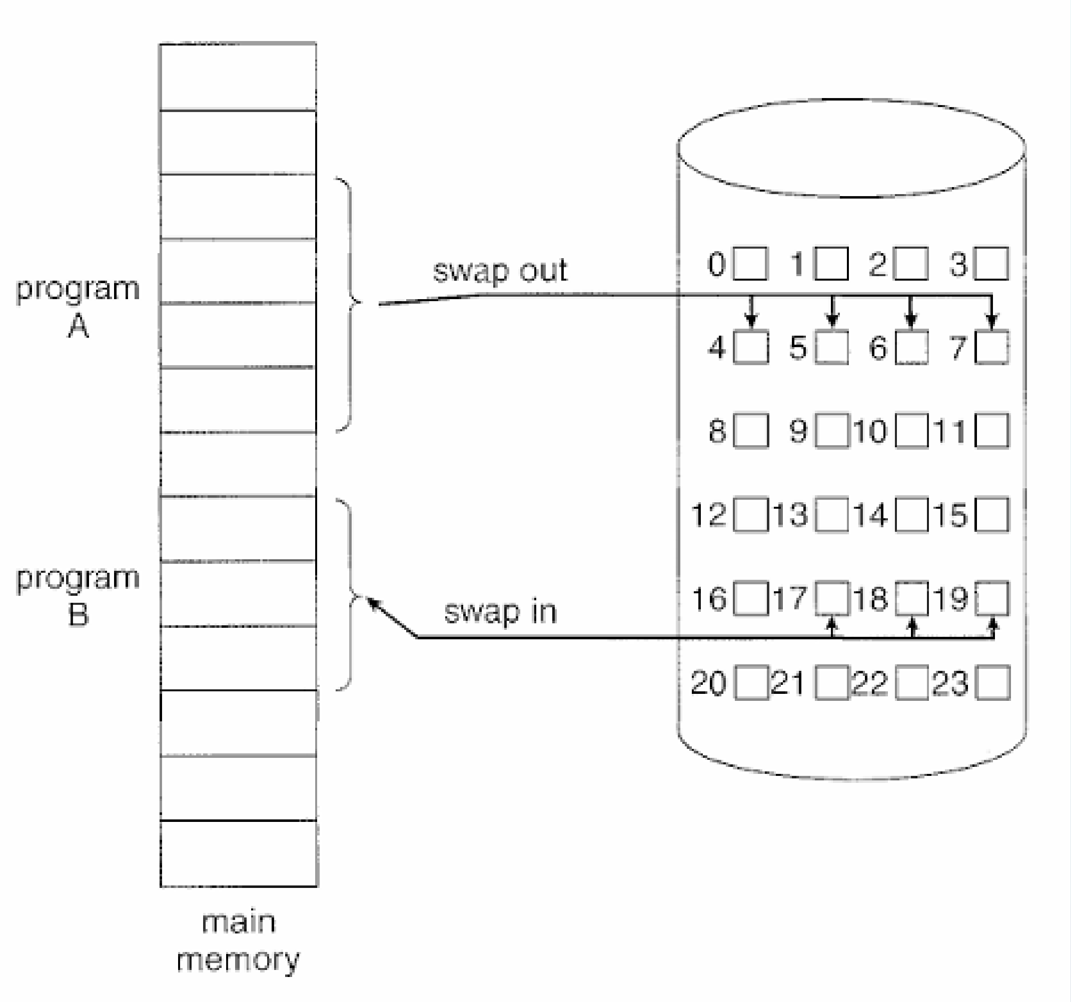
Q. Operating system use base and limit registers, but it has to be modified the machine to provide a page table. Can the page tables be set up to simulate base and limit registers? How can it be? Why can it not be?

- The page table can be set up to simulate base and limit registers provided that the memory is allocated in fixed-size segments. In this way, the base of a segment can be entered into the page table and the valid/invalid bit used to indicate that portion of the segment as resident in the memory. There will be some problem with internal fragmentation

Explain Demand Paging illustrating with transfer of a paged memory to contiguous disk space.

- Loading to the entire program in physical memory at program execution time. A problem with this approach is needed the entire program in memory. Suppose a program starts with a list of available options from which the user is to select. Loading the entire program into memory results in loading the executable code for all options. An alternative strategy is to load pages only as they are needed. This technique is known as demand paging.

Demand Paging (Cont…)

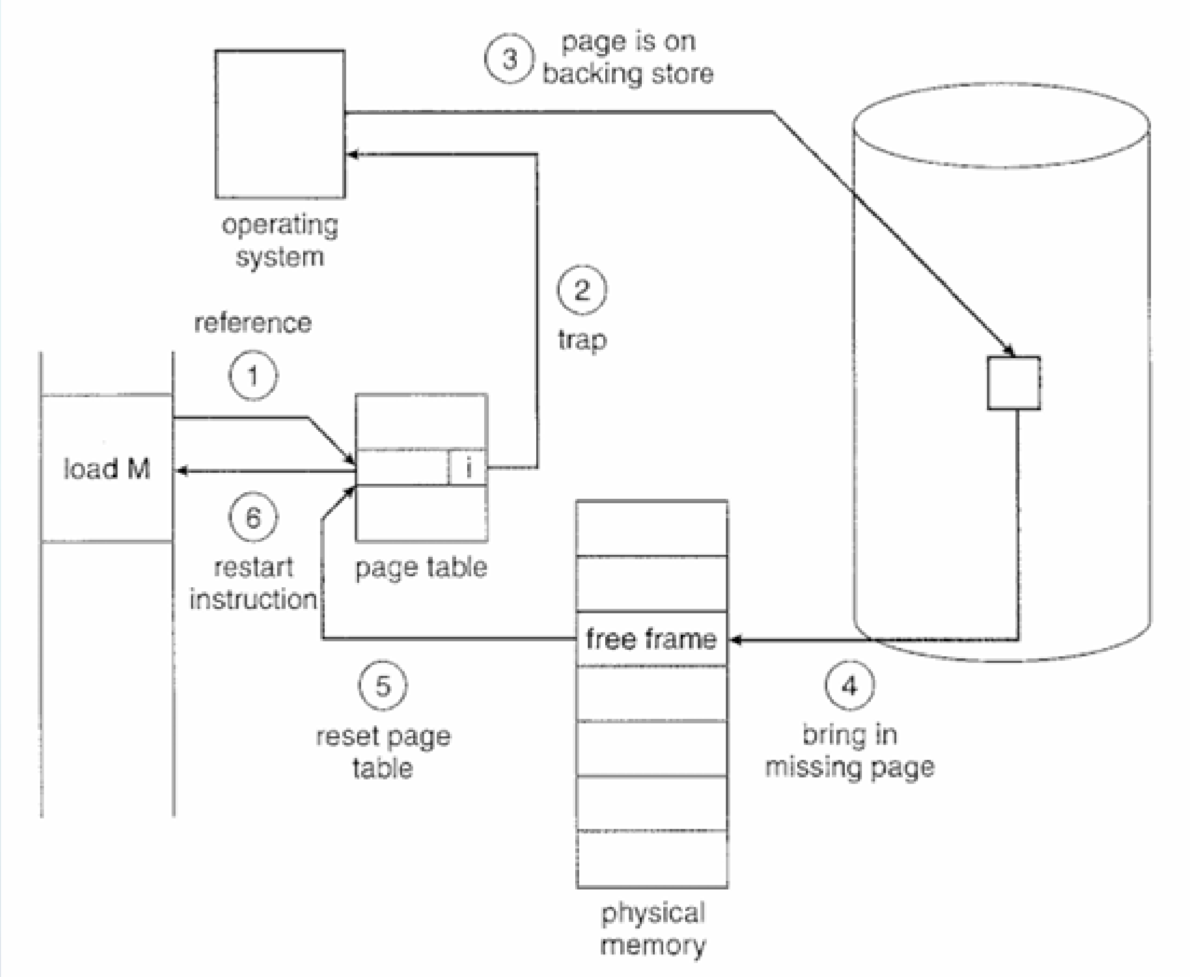


- Transfer of a paged memory to contiguous disk space

Q. Explain and illustrate steps in handling a page fault.

1. The procedure checks an internal table for this process to determine whether the reference was a valid or an invalid memory access.
2. If the reference was invalid, we terminate the process. If it was valid, but we have not yet brought in that page, we now page it in.
3. It finds a free frame.
4. It is scheduled a disk operation to read the desired page into the newly allocated frame.
5. When the disk read is complete, we modify the internal table kept with the process and the page table to indicate that the page is now in memory.
6. And it restarts the instruction that was interrupted by the trap. The process can now access the page as though it had always been in memory.

Steps in handling a page fault (Cont…)



Q. In computing the effective access time, it must be known how much time is needed to service a page fault. What are they causing page fault?

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:

   a. Wait in a queue for this device until the read request is serviced.

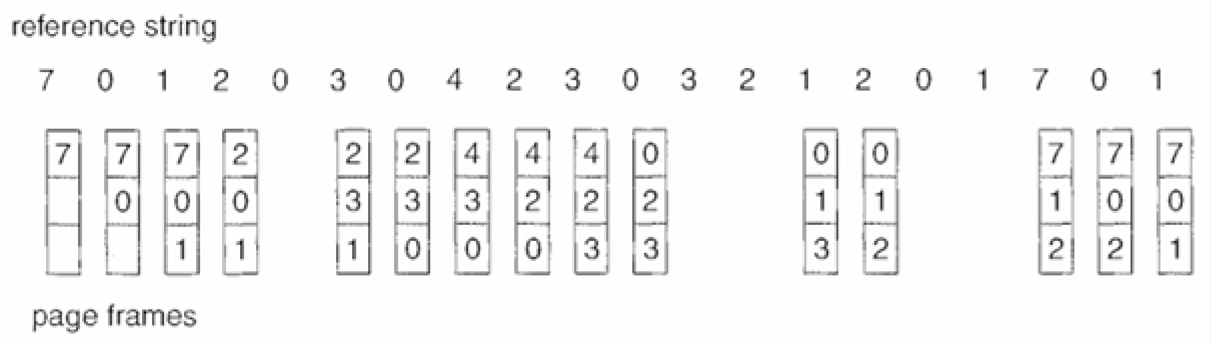
   b. Wait for the device seek and/ or latency time.

   c. 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/0 subsystem.
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 that the desired 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.

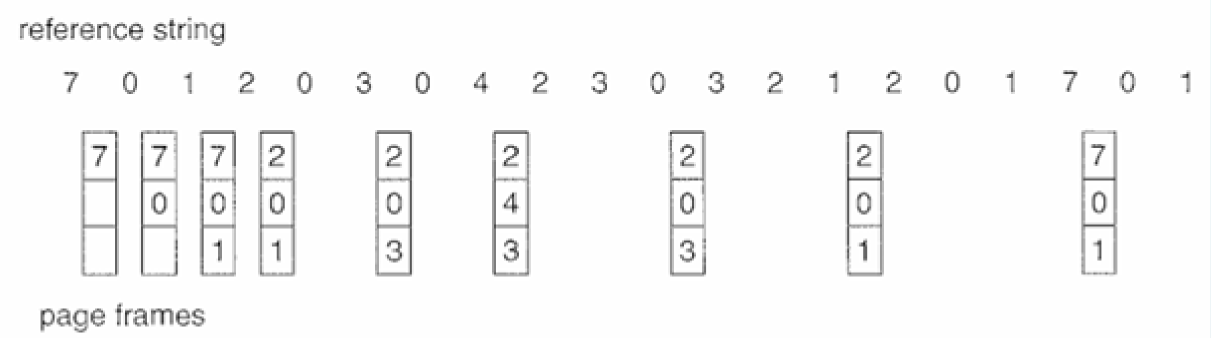
Q. In the following page reference string, three frames are initially empty:

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

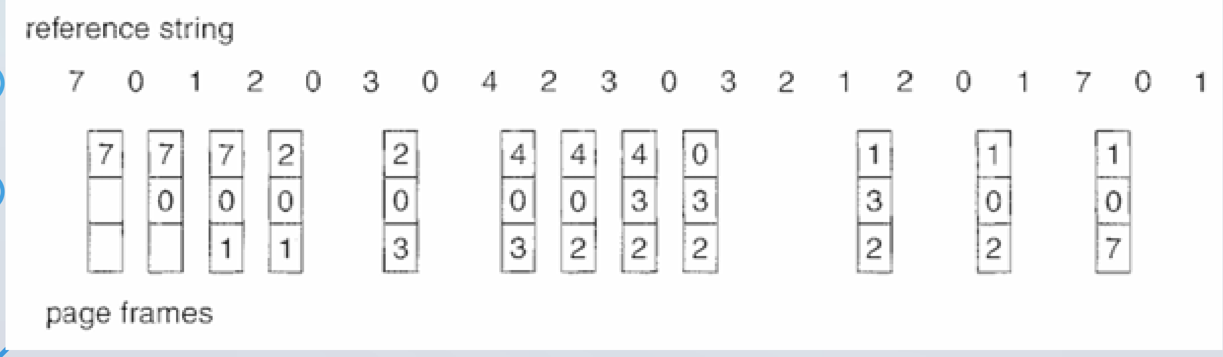
Consider the page fault using (a) FIFO page-replacement algorithm, (b) Optimal page-replacement algorithm (c) LRU page-replacement algorithm



- FIFO page-replacement algorithm



- Optimal page-replacement algorithm



- LRU page-replacement algorithm

References

Images: Internet

Source: Operating System Concepts (8th Edition)