

جامعة الأنبار

كلية علوم الحاسوب وتكنولوجيا
المعلومات

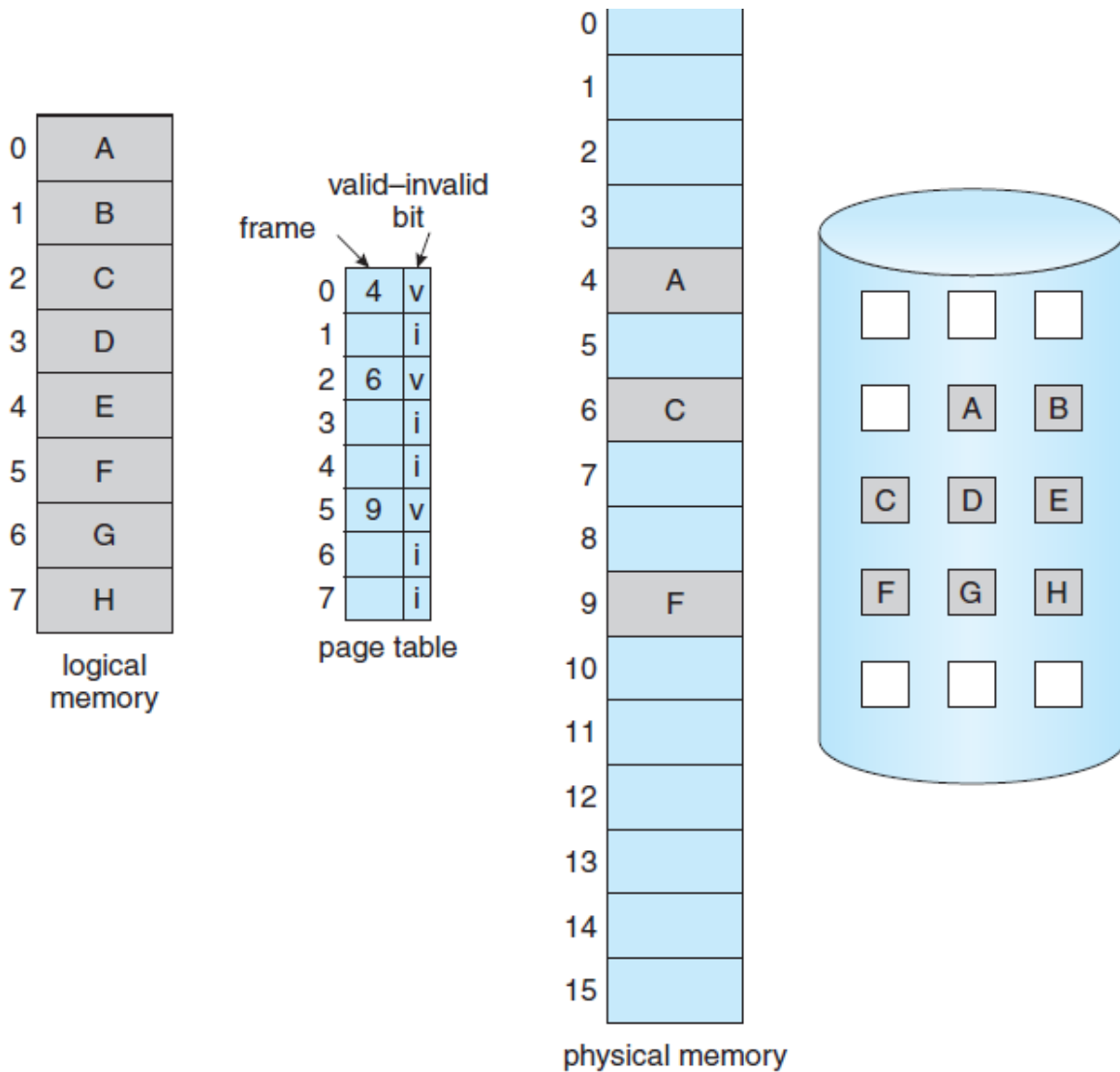
قسم أنظمة شبكات الحاسوب

المرحلة الرابعة

Operating System

التدريسي: أ.م.د. عمر منذر حسين

- Virtual memory is a technique that allows the execution of processes that may not be completely in memory.
- The main visible **advantage of this scheme is that programs can be larger than physical memory.**
- Virtual memory is a technique for using the secondary storage to extend the limited size of the physical memory beyond its actual physical size.
- Those parts of the program that are currently active are brought to the main memory while those parts that are not active will be stored on the magnetic disk.
- First, we discuss implementing virtual memory through **demand paging.**
- Bring a page into memory only when it is needed (Pager) not swapper.
- Less I/O needed, Less memory needed, Faster response, More users.
- when The **valid–invalid** bit is set to “valid,” the associated page is both legal and in memory.
- If the bit is set to “invalid,” the page either is not valid (that is, not in the logical address space of the process) or is valid but is currently on the disk.



- During address translation, if valid–invalid bit in page table entry is 0 (invalid) >> **page fault**, will lead to trap OS
- Get empty frame.
- Swap page into frame.
- Reset tables, validation bit = 1.
- Restart instruction
- During address translation, if valid–invalid bit in page table entry is 0 (invalid) >> **page fault**, will lead to trap OS
- Get empty frame.

- Swap page into frame.
- Reset tables, validation bit = 1.
- Restart instruction

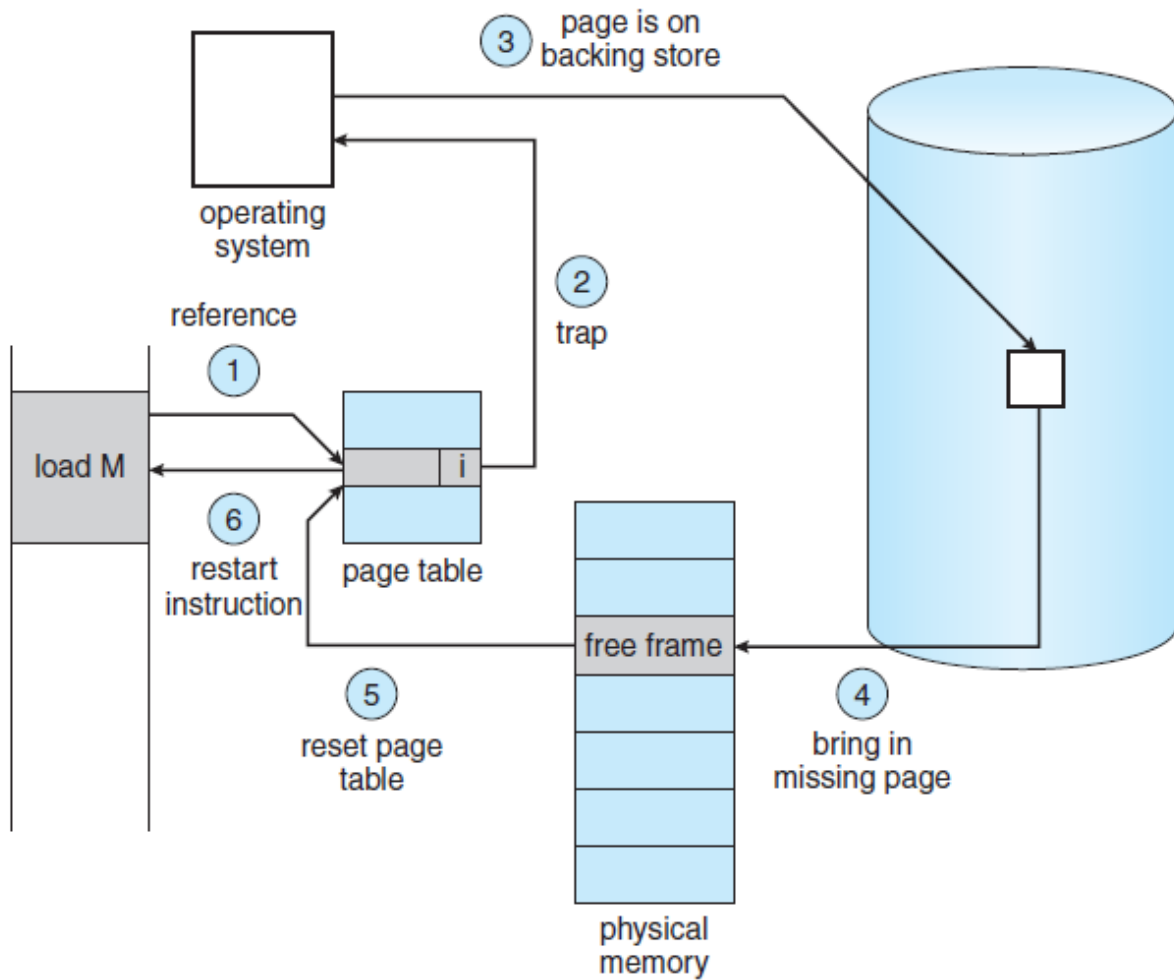
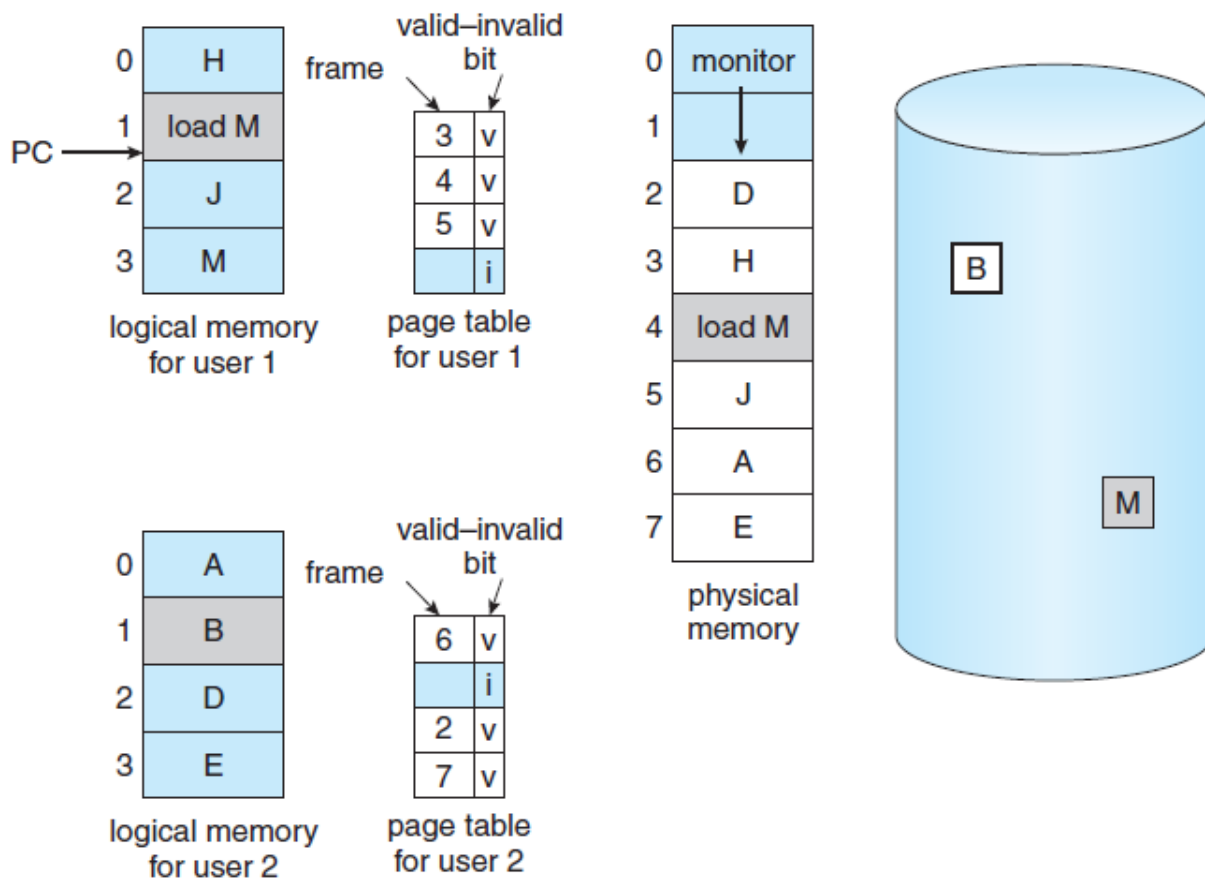


Figure 9.6 Steps in handling a page fault.

- **Page replacement**
- The operating system determines where the desired page is residing on the disk but then finds that there are no free frames!!!



•

Basic page replacement

- If no frame is free, we find one that is not currently being used and free it. We can free a frame by writing its contents to swap space and changing the page table to indicate that the page is no longer in memory.

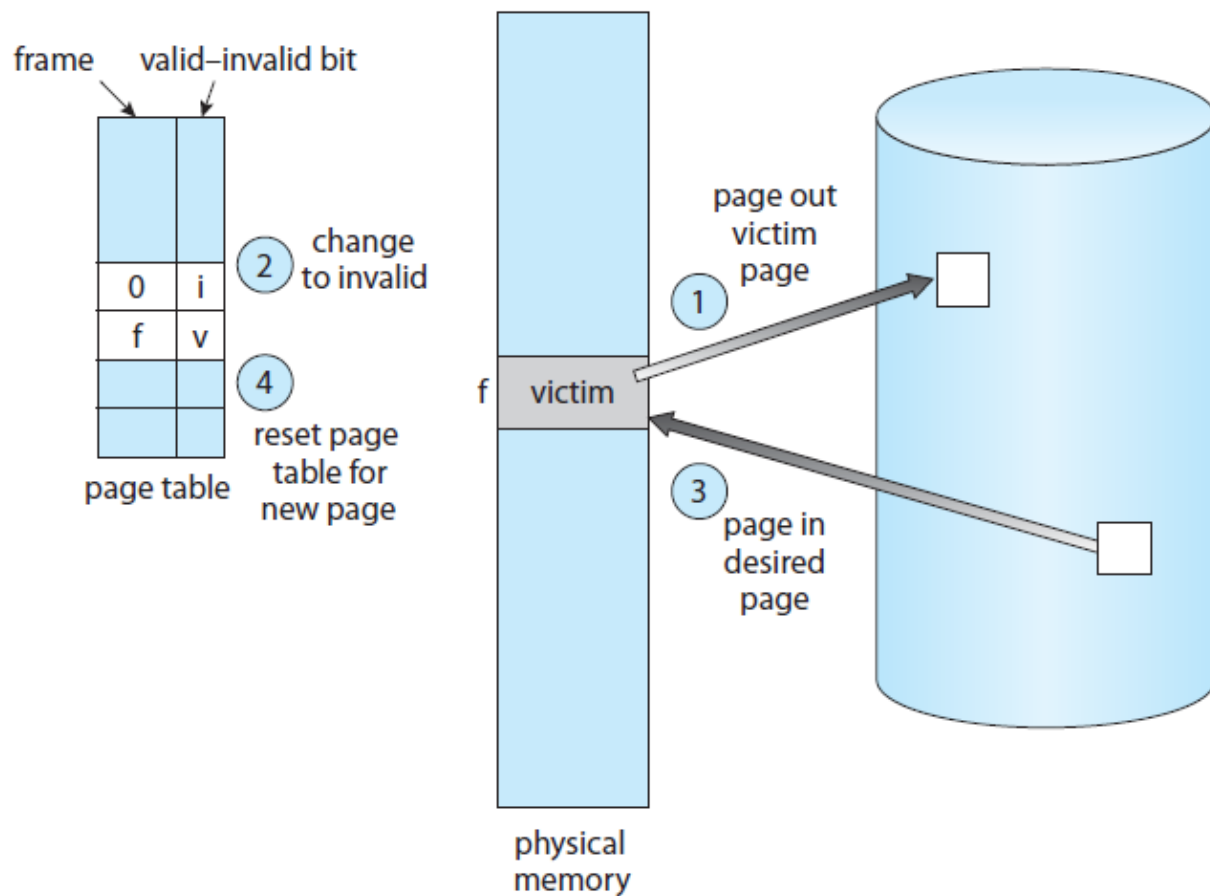


Figure 9.10 Page replacement.

- 1. Find the location of the desired page on the disk.
- 2. Find a free frame:
 - a. If there is a free frame, use it.
 - b. If there is no free frame, use a page-replacement algorithm to select a victim frame.
 - c. Write the victim frame to the disk; change the page and frame tables accordingly.
- 3. Read the desired page into the newly freed frame; change the page and frame tables.
- 4. Continue the user process from where the page fault occurred.

- Use modify (**dirty**) bit to reduce overhead of page transfers – only modified pages are written to disk.
- If **Dirty bit is set** this mean the page has been modified and need to rewrite on the disk.
- If **Dirty bit is reset** this mean the page has not been modified and we need not write the memory page to the disk: it is already there.

Replacement Algorithms (Policies)

- When a process needs a nonresident page, the operating system must decide which resident page is to be replaced by the requested page. The technique used in the virtual memory that makes this decision is called the **replacement policy**.
- There exists a number of possible replacement mechanisms:
- **Random Replacement:** page is selected randomly for replacement.
- **First-In-First-Out (FIFO) Replacement:** the page that was loaded before all the others in the main memory is selected for replacement.
- **Least Recently Used (LRU) Replacement:** The page that has not been referenced for the longest time while residing in the main memory is selected for replacement.