Operating Systems

(Memory management: Demand paging, page replacement algorithms: FIFO, LRU)
Slides Set #18

By Prof K R Chowdhary

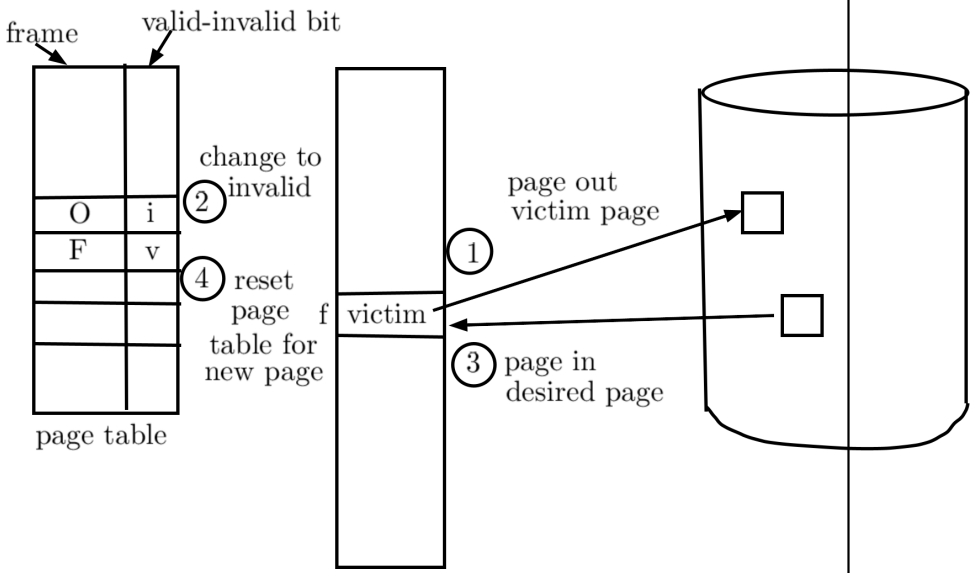
JNV University

January 15, 2024

Page Replacement

- Say a process has 10 pages, and it actually uses only half of them, then demand paging saves the I/O necessary to load the five pages that are never used. Advtg.?
- If we increase our degree of multiprogramming, we are over-allocating memory.
- Further, the fact is that system memory is not used only for holding program pages, but...
- In over-allocation of memory, a page may result result may see no frame available:
  - The operating system has several options at this point.
  - The operating system could instead swap out a process, freeing all its frames.

Basic Page Replacement



Page replacement approach:
If no frame is free, we find another that is not currently being used, and free it.

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

Basic Page Replacement ...

- Such pages (read-only) cannot be modified; thus, they may be discarded when desired.
- Page replacement is basic to demand paging. It completes the separation between logical memory and physical memory.
- We must solve two major problems to implement demand paging: develop a 1. frame-allocation algorithm, and 2. a page-replacement algorithm.
- We evaluate an algorithm by running it on a particular **string of memory references** and computing the number of page faults.
- For a given page consider only the page number, and not the entire address. If there is a reference to page p, then any references to p page will never cause page fault.

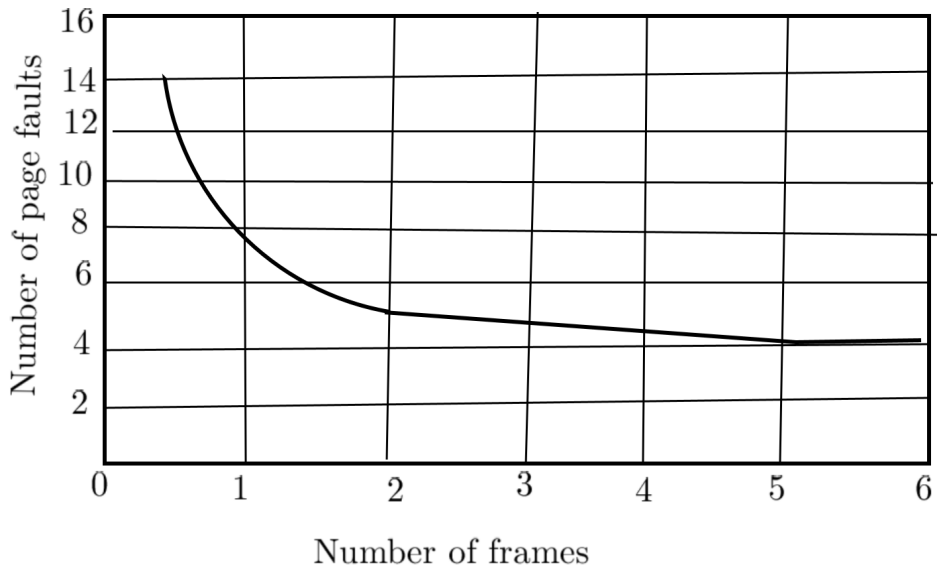
Basic Page Replacement ...

- For example, if we trace a particular process, consider that we record the following address sequence:

  0100, 0432, 0101, 0612, 0102, 0103, 0104, 0101, 0611, 0102, 0104, 0101, 0103, 0610, 0102, 0103, 0104, 0101, 0609, 0102, 0105

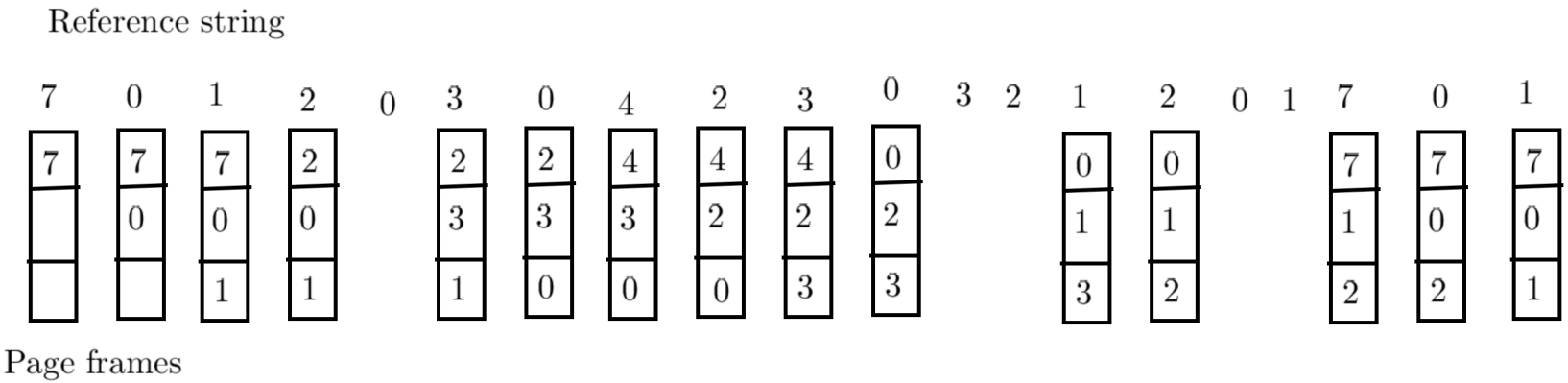
  At 100 bytes per page, this sequence is reduced to string:

  1, 4, 1, 6, 1, 6, 1, 6, 1, 6, 1



FIFO Page Replacement

- The simplest page-replacement algorithm is a first-in, first-out (FIFO) algorithm.
- For our example reference string, our **three** frames are initially empty. The first three references (7, 0, 1) cause page faults.



- The FIFO page-replacement algorithm is easy to understand and program. However, its performance?
- Notice that, even if we select for replacement a page that is in active use,
- Consider the following reference string for above fig.

  1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

LRU Page Replacement

- If the optimal algorithm is not feasible, perhaps an approximation of the optimal algorithm is possible.
- LRU replacement associates with each page the time of that page's last use.
- The result of applying LRU replacement to our example reference string is shown in Figure
- The LRU policy is often used as a page-replacement algorithm and is considered to be good.
  - Counters.
  - Stack.