Page Replacement

Page replacement takes the following approach. 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). We can now use the free frame to hold the page for which the process faulted. We modify the page-fault service routine to include 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 page to the disk; change the page and frame tables accordingly.
3. Read the desired page into the (newly) free frame; change the page and frame tables.
4. Restart the user process.

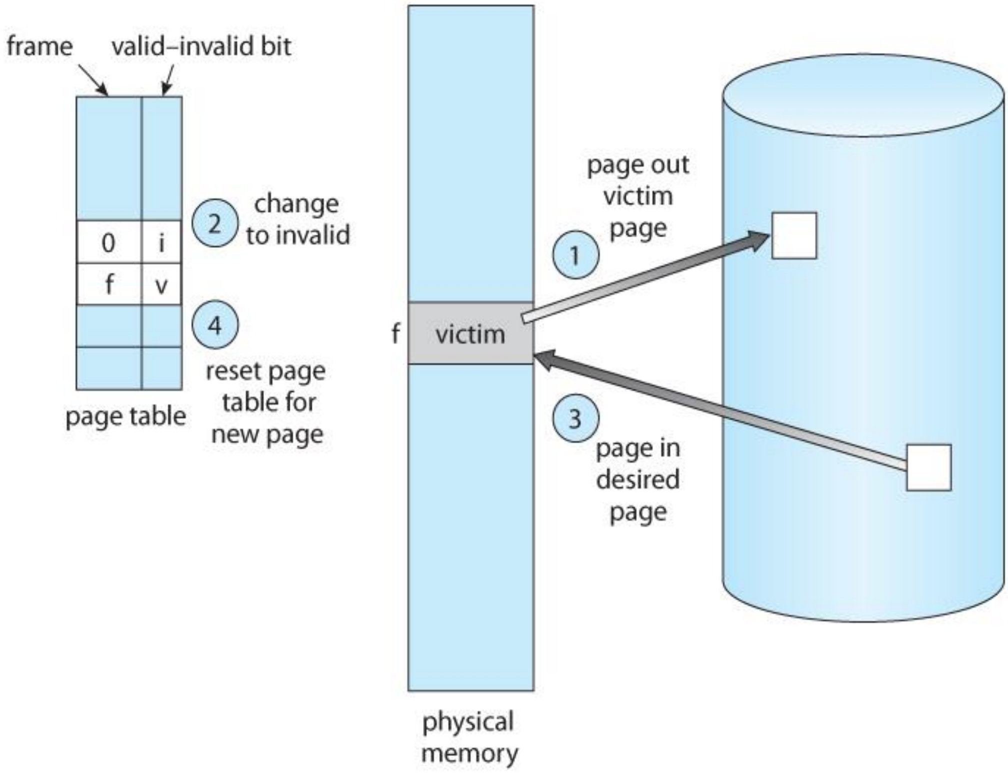


Figure 9: Page replacement.

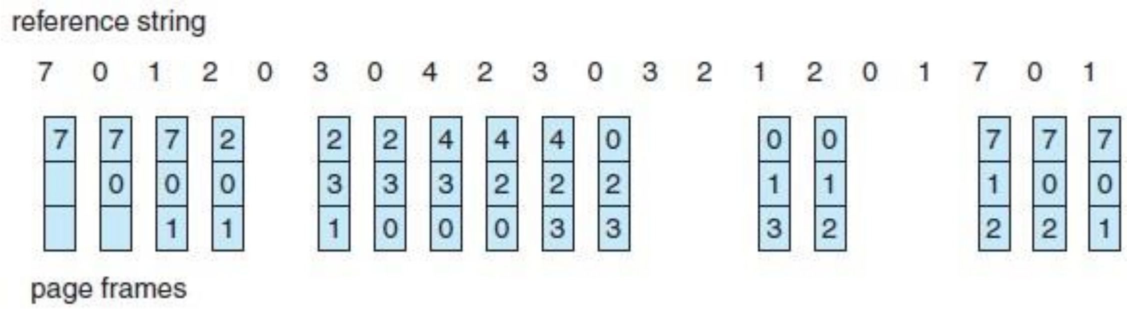
Page Replacement Algorithms

In an operating systems that use paging for memory management, page replacement algorithm are needed to decide which page needed to be replaced when new page comes in. Whenever a new page is referred and not present in memory, page fault occurs and Operating System replaces one of the existing pages with newly needed page. Different page replacement algorithms suggest different ways to decide which page to replace. The target for all algorithms is to reduce number of page faults.

1. FIFO Page Replacement

The simplest page-replacement algorithm is a FIFO algorithm. A FIFO replacement algorithm associates with each page the time when that page was brought into memory. When a page must be replaced, the oldest page is chosen. Notice that it is not strictly necessary to record the time when a page is brought in. We can create a FIFO queue to hold all pages in memory. We replace the page at the head of the queue. When a page is brought into memory, we insert it at the tail of the queue.

<u>Example:</u> Suppose three pages can be in memory at a time per process. Process references pages: 7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1 , what is the number of page fault?



The number of page fault = 15

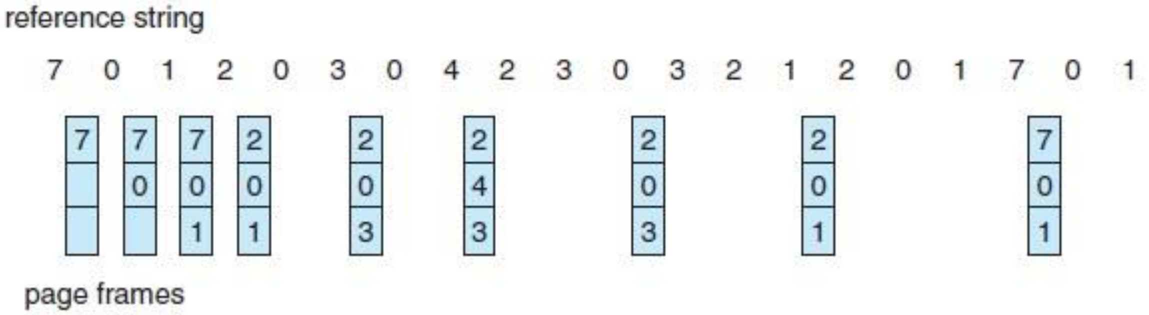
2. Optimal Page Replacement

In this algorithm, pages are replaced which are not used for the longest duration of time in the future. Use of this page-replacement algorithm guarantees the lowest possible page fault rate for a fixed number of frames.

Example: Suppose we have the following process references pages:

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

What is the number of page fault if we use three page frames?

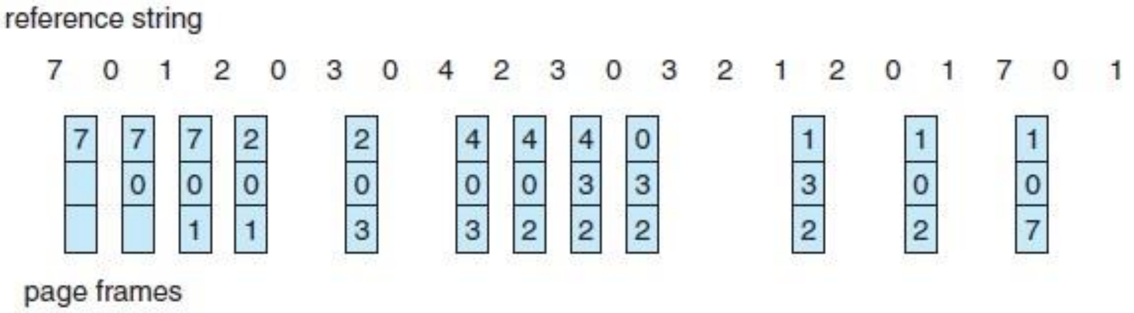


The number of page fault = 9

3. LRU Page Replacement

In least recently used (LRU) replacement algorithm page will be replaced which is least recently used. When a page must be replaced, LRU chooses that page that has not been used for the longest period of time.

Example: Suppose three pages can be in memory at a time per process. Process references pages: 7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1 , what is the number of page fault?



The number of page fault = 12