Assignment Number: 6

TITLE: Page Replacement algorithms.

Problem statement- Implement the C program for Page Replacement Algorithms : FCFS, LRU and optimal for frame size as minimum three.

OBJECTIVE:
To study page replacement algorithms for virtual memory of operating system.

THEORY:
Regardless of the resident set management strategy, there are certain basic algorithms that are used for the selection of a page to replace in main memory. General page replacement algorithms include-

- Optimal
- Least recently used (LRU)
- First-in-first-out (FIFO)

The optimal policy:

It selects for replacement that page for which the time to the next reference is the longest. It can be shown that this policy results in the fewest number of page faults. Clearly, this policy is impossible to implement, because it would require the operating system to have perfect knowledge of future events. However, it does serve as a standard against which to judge real- world algorithms. Figure below gives an example of the optimal policy. The example assumes a fixed frame allocation (fixed resident set size) for this process of three frames. The execution of the process requires reference to five distinct pages. The page address stream formed by executing the program is-
2 3 2 1 5 2 4 5 3 2 5 2
which means that the first page referenced is 2, the second page referenced is 3, and so on. The optimal policy produces three page faults after the frame allocation has been filled.

The least recently used (LRU) policy:

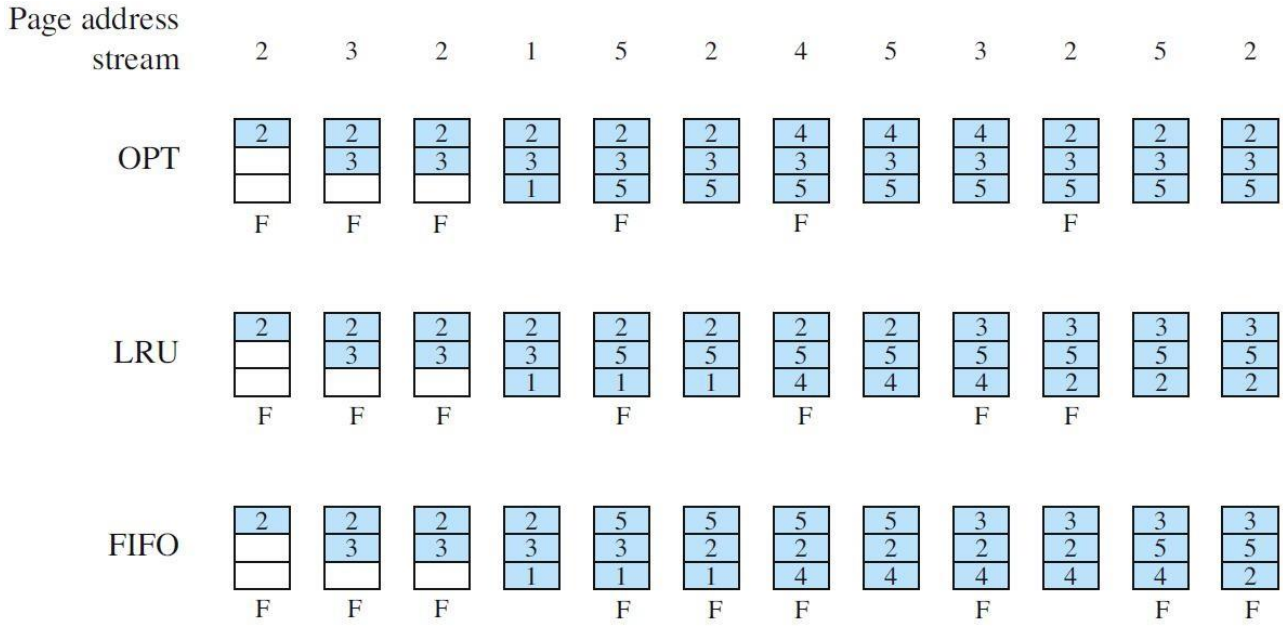
It replaces the page in memory that has not been referenced for the longest time. By the principle of locality, this should be the page least likely to be referenced in the near future. And, in fact, the LRU policy does nearly as well as the optimal policy. The problem with this approach is the difficulty in implementation. One approach would be to tag each page with the time of its last reference; this would have to be done at each memory reference, both instruction and data. Even if the hardware would support such a scheme, the overhead would be tremendous. Alternatively, one could maintain a stack of page references, again an expensive prospect.
Figure shows an example of the behavior of LRU, using the same page address stream as for the optimal policy example. In this example, there are four page faults.

The first-in-first-out (FIFO) policy:

It treats the page frames allocated to a process as a circular buffer, and pages are removed in round-robin style. All that is required is a pointer that circles through the page frames of the process. This is therefore one of the simplest page replacement policies to implement. The logic behind this choice, other than its simplicity, is that one is replacing the page that has been in memory the longest: A page fetched into memory a long time ago may have now fallen out of use. This reasoning will often be wrong, because

there will often be regions of program or data that are heavily used throughout the life of a program. Those pages will be repeatedly paged in and out by the FIFO algorithm.
In the below example, FIFO policy results in six page faults. Note that LRU recognizes that pages 2 and 5 are referenced more frequently than other pages, whereas FIFO does not.

Example:



Conclusion :

Write conclusion in your own words