CS 314: Operating Systems Laboratory

Assignment 8

Ashwin Waghmare
210010060

Tejal Ladage
210010026

1) **FIFO**

In the First in First Out (FIFO) replacement policy, when a page fault happens and the physical memory is at capacity, the oldest page—meaning the one that was initially brought into memory—is replaced. While FIFO is a straightforward policy, it may be prone to thrashing.

To implement this a vector is used and when a new page comes it is pushed at back of vector and when a page fault occurs and physical memory is full then the first page in vector is removed to create space for new page.

2) **LRU**

In the Least Recently Used (LRU) replacement policy, when a page fault arises and the physical memory is full, the page that has been least recently used is evicted to make room for the new page. LRU tends to perform better than FIFO when there's a locality of reference in the sequence of page requests. It strikes a good balance between performance and complexity.

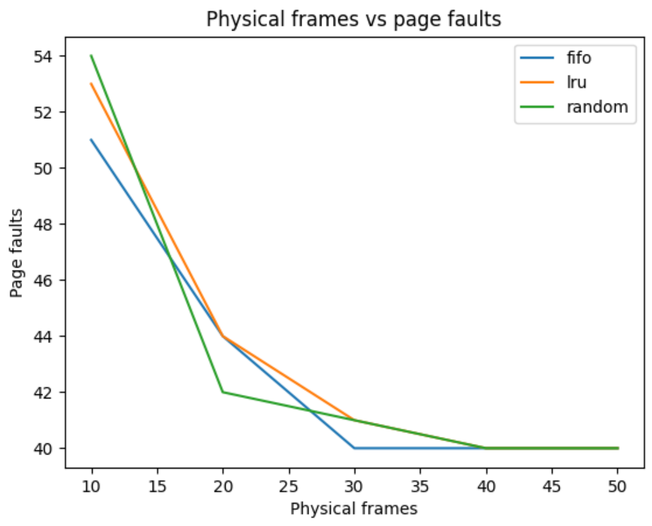
To execute this process, a vector is utilized to store pages in physical memory. When a new page is introduced, it is added to the end of the vector. If a previously accessed page is requested again, it is relocated to the end of the vector. If the physical memory reaches its capacity, the page at the front of the vector is evicted to accommodate the requested page.

3) **Random**

In Random replacement policy when physical memory is full, a page is randomly removed to make space for new page.

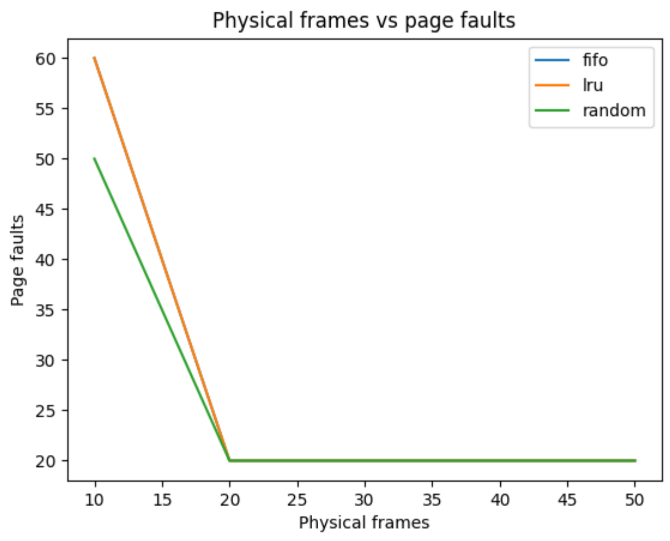
To implement this approach, a vector is employed to manage pages within physical memory. When a new page arrives, it is appended to the end of the vector. If the physical memory reaches its limit, a page is chosen randomly and removed to create space for the incoming page.

a) Results for req1.dat



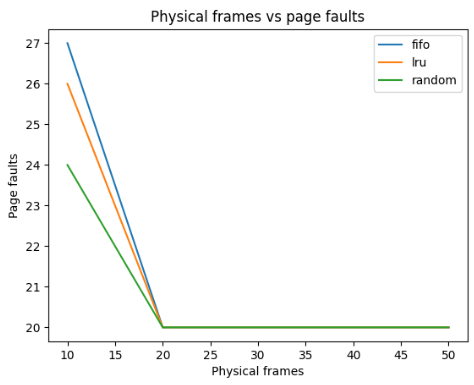
From the results we can see that FIFO is working better that LRU. As the number of frames in main memory increases replacement policy doesn't affect the efficiency in this case after 40 all replacement polices give the same result. Here FIFO works better than LRU.

b) Results for req2.dat



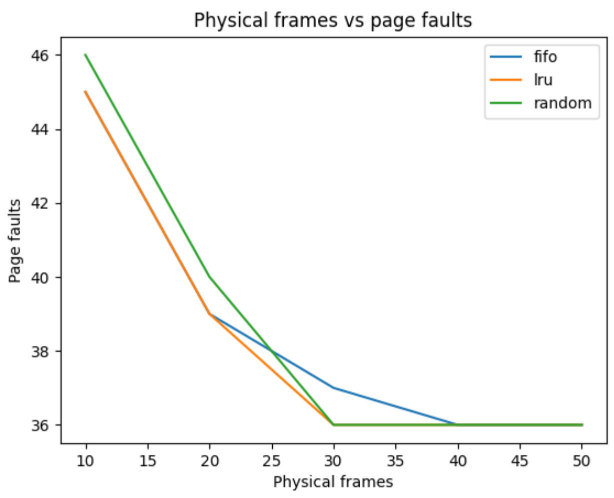
This workload is an example for looping sequential. Under a looping sequential workload, FIFO and LRU kick out older pages. Because of the workload's cyclical nature, these older pages will be viewed more quickly than the pages that the policies wish to preserve in cache. Here we can see that FIFO and LRU page faults decreases linearly as the number of frames are increasing.

c) Results for req3.dat



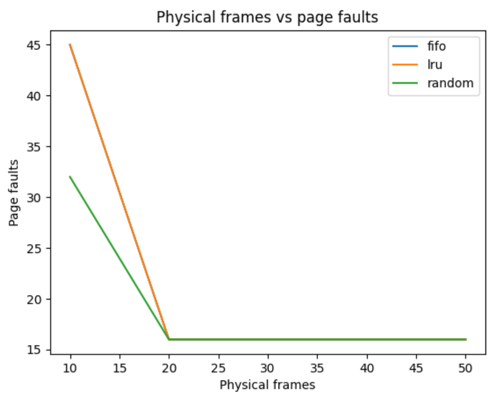
This sequence is similar to 80-20 workload which exhibits locality. In this workload, LRU gives better results than FIFO. After a particular point, in this case after 20, replacement policy doesn’t affect LRU, FIFO, Random and all give the same results.

d) Results for req4.dat



In this workload, page requests are generated randomly. We can see from the above graph that Random replacement policy gives much better results than FIFO and LRU. Also, as number of frames increase, the page faults are decrease.

e) Results for req5.dat



This sequence is an example for Be-lady’s anomaly which states that at a point on as number of frames increases number of page faults for FIFO replacement policy. For this sequence we can observe spike in page faults from 14 page frames to 15 page frames.