Operating Systems Lab 8

Divy Jain (210010015), Karthik Hegde (210010022)

March 17, 2024

Problem

Implement FIFO, LRU and Random page replacement policies. We are given a fixed number of addressable frames in the storage (disk), the main memory size or pages in main memory and a swap space for storing excess or unused pages. Find the number of page faults in each of the policies for different configurations.

Solution

The basic idea behind the policies is as follows.

FIFO (First-In-First-Out): This policy replaces the oldest page in memory.

LRU (Least Recently Used): This policy replaces the least recently used page in memory.

Random: This policy randomly selects a page from memory to replace.

Configurations

We design our configurations while testing as follows

- there are total 100 total frames
- at most 50 frames are used
- main memory size is varied from 10 to at most 50
- swap is large enough to accommodate all page requests
- there are at least 5 configurations for each request

Requests

There are total 5 different requests based on the policies.

req_random.dat This page access order consists random values. Total frames are 100, swap space is 50, and main memory is varied from 10 to 25 in steps of 3. Here random policy initially performs better but fails as the memory size increases.

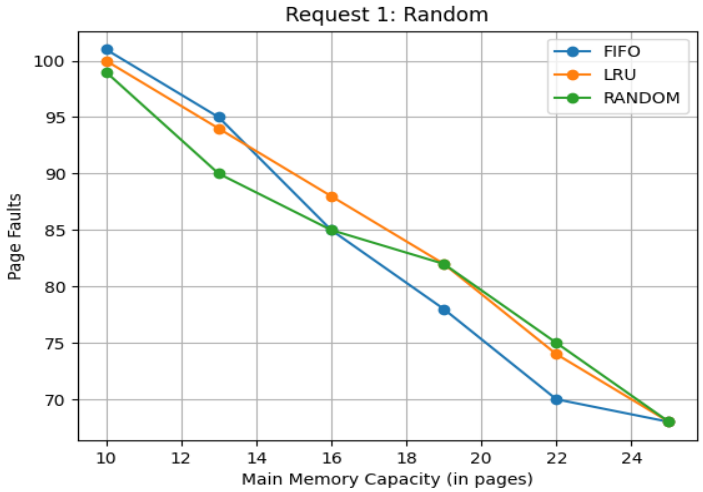

Figure 1: Request 1 Memory 10 13 16 ... 25

req_fifo.dat This page access order consists values such that it favours FIFO over LRU. Total frames are 100, swap space is 50, and main memory is varied from 10 to 25 in steps of 3. The numbers or pages in this request order are picked such that the page evicted by LRU's recent access update are accessed again, favouring FIFO.

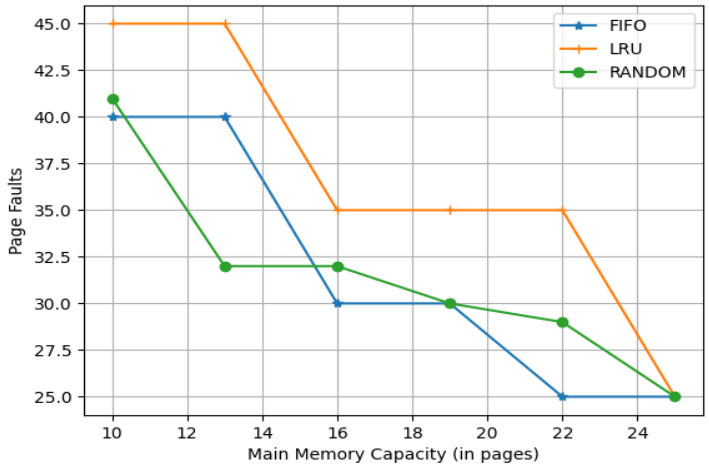

Figure 2: Request 2 Memory 10 13 16 ... 25

req_lru.dat The pages are accessed and reused in the reverse order. For example,1 1, 1 2 2 1, 1 2 3 3 2 1, 1 2 3 4 4 3 2 1 and so on in increasing length. Total frames are 100, swap space is 50, and main memory is varied from 25 to 50 in steps of 5. Here LRU policy performs better than FIFO and random.

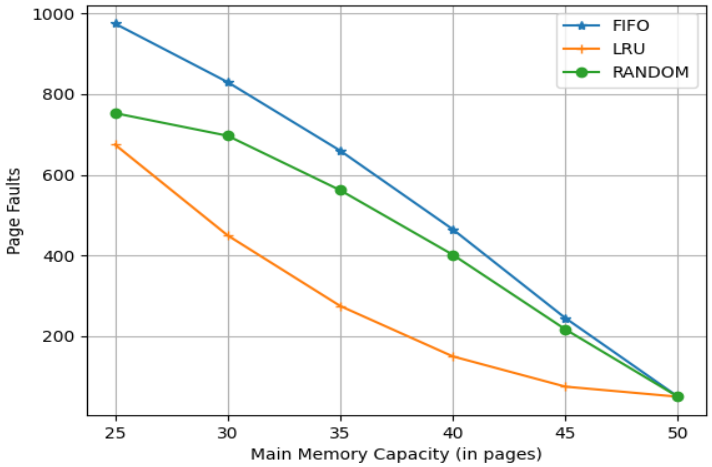

Figure 3: Request 3 Memory 25 30 35 .. 50

req_both.dat The pages are accessed in increasing length sequences like 1 to 10, 1 to 15, 1 to 20 and so on. Total frames are 100, swap space is 50, and main memory is varied from 10 to 50 in steps of 5. Here both LRU and FIFO have same page faults, but random preforms better than both.

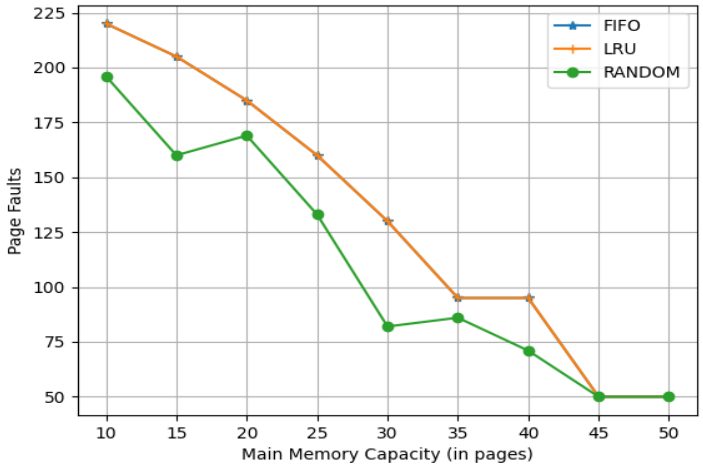

Figure 4: Request 5 Memory 10 15 20 .. 50

req1.dat This request order is same as the given sample input file. Total frames are 60, swap space is 50, and main memory is varied from 10 to 30 in steps of 5. We observe FIFO perform better than LRU and random in this memory range.

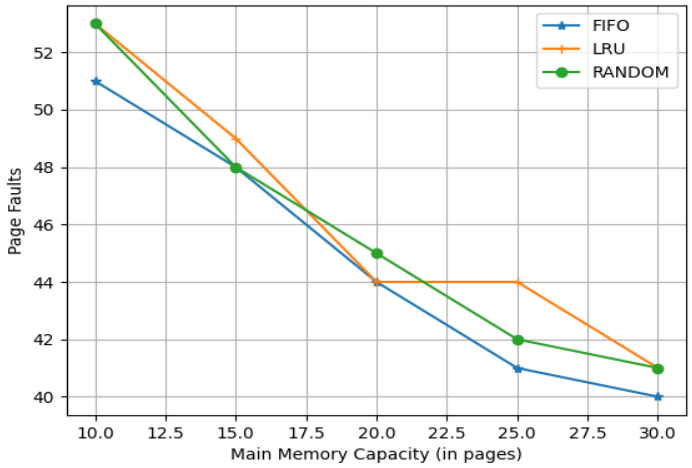

Figure 5: Request 5 Memory 10 15 20 25 30

Conclusion

We observe that the number of page faults depends on main memory size. The number of page faults decreases as the memory size is increased. Other than that they also depend on the eviction policy and page order. In req_fifo.dat page order cases FIFO performs better and in req_lru.dat page order LRU perform better. In other cases there is not a large difference in FIFO and LRU and they perform equally well. Howerver, in real world scenario LRU is known to perform better.