

# CS314: Lab Report

## Assignment 8

Devdatt N

200010012@iitdh.ac.in

19 March 2023

## 1 Introduction

In this assignment, we analyze various page replacement algorithms with multiple parameters such as virtually addressable pages, page frames accommodatable in main memory, and swap space. However, here only the main memory size is relevant to our analysis, and others are for overflow detection.

## 2 Analysis

Here, we have used FIFO, LRU, and Random page replacement policies for different request files. They spread across being FIFO-suitable and LRU-suitable. We will look at plots constructed by varying the main memory accomodatable frames. It is to be noted that we call virtually addressable pages 'x', and main memory size 'y', and swap space as 'z'.

### 2.1 Request 1

This is the file given with the assignment itself. We run this with  $x = 60$ ,  $z = 30$  and vary  $y$  from 15 to 35. In this case, we get this particular plot: We can see that this favors FIFO more than LRU. As

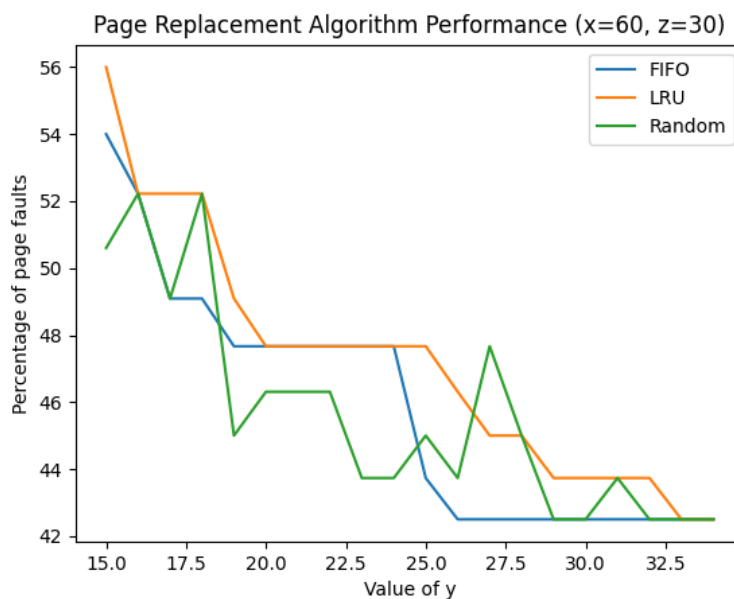


Figure 1: Results

expected, random works in a very random way but of course, the page fault percentage goes down anyway because our main memory size is increased.

## 2.2 Request 2

This is a series of randomly generated numbers. We run this with the exact same parameters as the last question, but varying  $y$  from 25 to 40. We get this particular plot: Here, FIFO and LRU run very

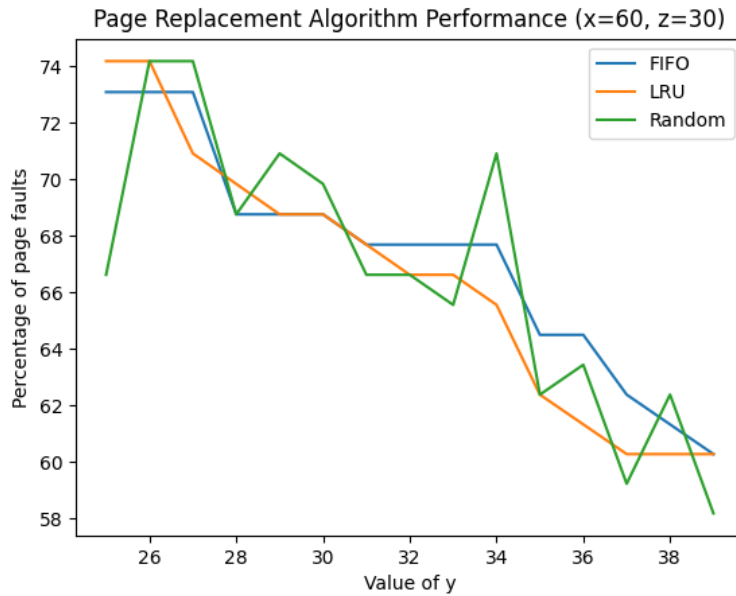


Figure 2: Results

closely, since we have used a uniform integer distribution to generate random numbers. Random, as usual does very random things.

## 2.3 Request 3

This does not really favor FIFO or LRU, since pages are seldom used again. We run with the exact same parameters as last request, and we get this plot:

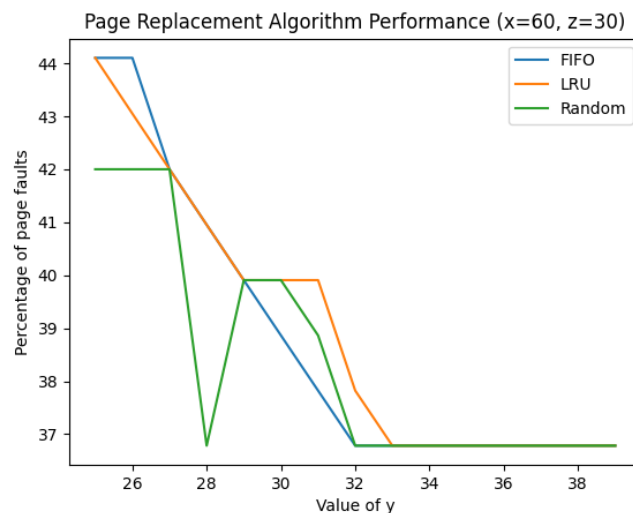


Figure 3: Results

We can see here that there is not much of a difference between FIFO and LRU page replacement policies, and even Random cannot make much of a difference.

## 2.4 Request 4

This is a series of requests favoring FIFO and not LRU. On plotting this again with the exact same parameters, we get this: This cyclically requests for 20 pages over and over again, which results in

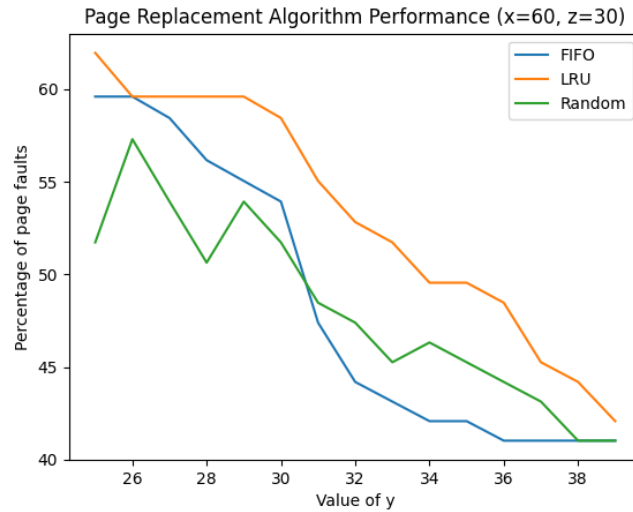


Figure 4: Results

more page faults for LRU but not for FIFO. Interestingly enough, Random performs better than LRU here in all cases and even better than FIFO here at times.

## 2.5 Request 5

This is a series of requests favoring the Least Recently Used or LRU replacement policy. This is because similar pages requested are more bound together and hence LRU works better because it does not evict the recently used pages, while FIFO does so. We run it with the exact same parameters again. Here is the plot:

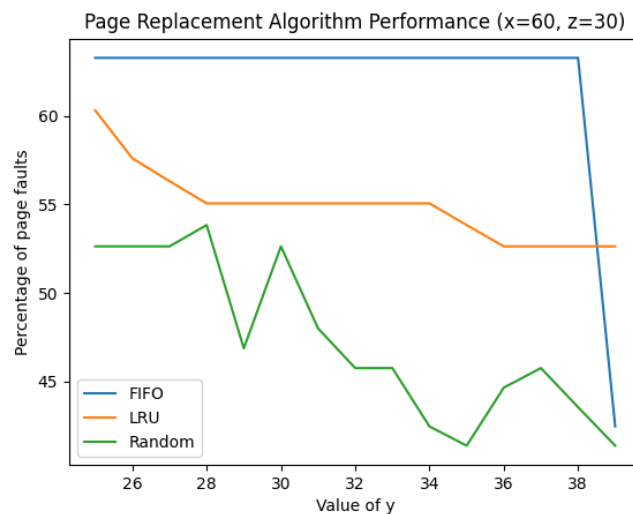


Figure 5: Results

We can see that FIFO performs quite horribly in this, much more than LRU. It is also interesting that Random performs the best in this. Random does run better almost every single time, which might mean it has to do something with the uniform int distribution and its connection with the used dataset.

## 2.6 Request 6

This is a series of 500 requests where 80 percent of requests are made to 20 percent of the pages, to simulate the 80-20 rule. LRU performs excellently in this, with parameters  $x = 100$ ,  $y$  varying from 40 to 50 and  $z = 30$ . We get this plot: This is a real-world scenario, and we can see that LRU

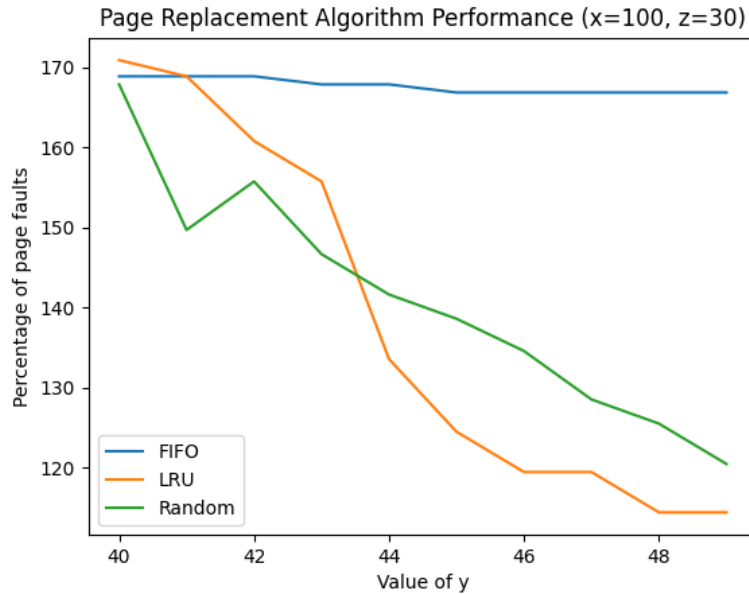


Figure 6: Results

performs much better in this, even better than Random. Thus, we can see that LRU is the best in real-world scenarios.

## 3 Conclusion

We have seen various examples, some where FIFO performed better, some where LRU performed better and one where even Random performed better. However, we can confirm that in a real world scenario, i.e, following the 80-20 rule, LRU would perform the best.