

Laboratory 8 (Deadline 17th Mar 2024, 11.59PM) (Groups of Two) assignment

Implement FIFO, LRU and Random page replacement policies:

There are fewer physical frames in main memory than logical pages. Hence all pages needed in the system may not be in physical memory. In such a case, swap space can be used when needed to bring in a swapped out page in main memory and a victim frame is swapped out. A system has a few addressable pages (say 60 in the entire system), few frames in main memory (say 20) and a few disk blocks (say 60) for swap space. The 3 numbers will be provided as command line args to your program. Implement FIFO, LRU and Random page replacement policies which selects a victim page based on the arrival of the page into main memory. When a requested page is found to be not available in main memory, the event is termed as a **page fault**. You are also supposed to throw an error if you require more blocks in the swap space than it is available.

Fourth argument to your executable file will be filename. In this file the sequence of page requests will be given. You will start with a nil map of logical pages to frames and build it on demand as and when requests arrive.

A sample page requests file is given

https://drive.google.com/file/d/1m-hoGYk_J9TpKpWwc6lInL4aETrzUQvG/view?usp=share_link

You need to try with 5 different files and 5 different configurations(i.e. Different number of pages and frames)

You are required to plot graphs for the number of page faults (on y-axis) against the total number of frames (on x-axis) for the different request patterns. Use matplotlib to plot the graph.

NOTE: The idea is that you will choose five different configurations (by varying the no. of frames and the no. of pages) and you will have five different request pattern files. For each of these request patterns, for the five different configurations, you need to plot the page fault graph.

Submit: a single zip file (format: <roll-number1>_<roll-number2>_lab8.zip) with all required source files and the completed report with all the required analysis. The evaluator will simply unzip the submission, and run the file with different process data files. If the desired output is not seen, you will not be awarded any marks.