

This is a two part assignment. It is slightly open ended and you can make assumptions. Please mention that in the README in the submission folder. Please have a project directory structure (with src/inc/lib/etc) and a non trivial Makefile.

Part A:

In this assignment, you will need to simulate different page-replacement algorithms and compare their performances. Your simulations should be able to take in the required parameters from a data file in the following format:

- First line of the file contains two integers in this order - the no. of pages in the reference string and the no. of page frames
- The rest of the file contains page reference string as a sequence of integers that are the page numbers (in virtual address space) accessed by the process in sequence.

The simulator will simulate the behavior of the following page replacement algorithms on the reference string and report the number of page faults generated for each algorithm:

- FIFO
- LFU
- LRU
- Clock
- N Chance

The simulator should take the following command line arguments (in this sequence): the name of the data file, a sequence of strings (max. 5) from the following sets: FF, LF, LR, CL, SC (meaning the above 5 algorithms respectively). The simulator simulates only the algorithms specified in the command line. If no algorithm is specified, all 5 are simulated.

Each of the algorithms should be implemented as a separate function that is called from the main function.

Part B

We studied about Virtual Machines in class and I did refer to the fact that they suffer from a performance slowdown. In this part, I would like you to do a study and document this performance slowdown.

Take two benchmarks --- the multi threaded program that you wrote in Assignment 2 and [Y-cruncher](#).

Measure the performance for 3 platforms

- Native Ubuntu
- Use a container --- [docker](#)
- Run it on Ubuntu on QEMU or Virtual box.

Plot it for different values of different number of digits.

Useful References:

1. <https://course.ece.cmu.edu/~ece845/sp18/docs/containers.pdf>
2. http://faculty.washington.edu/wlloyd/courses/tcss562/research_papers/T3_Hypervisors_v_s_Lightweight_Virtualization_A_Performance_Comparison.pdf