

**Lab Assignment 3 (30 points)****Simulating Page Replacement Algorithms****Due date: 11:59 pm on December 5 (Friday)****1. The goal of this programming assignment**

The primary goal of this lab assignment is to implement three paging replacement algorithms and evaluate their effectiveness, which affects memory management performance.

**2. Lab assignment description**

Write a C or C++ program that simulates the operation of three-page replacement algorithms used in a virtual memory management system:

1. FIFO (First-in-First-out) algorithm
2. Least Recently Used (LRU) algorithm
3. Optimal (OPT) algorithm

You will simulate each algorithm on the input datasets and report its performance. A description of each of these algorithms appears in Chapter 10 of our text.

**2.1 Specification**

In this lab assignment, you will calculate the page fault rate of each different page replacement algorithm for the given page reference string for one process only. Your program should accept three arguments: 1) **the number of page frames of the physical memory**, 2) **the name of an input file**, and 3) **an output file**. The input file contains the page reference string in the order of page request, and the output file stores the display events and the analysis results. You will run each algorithm four times: 1) **the first time with 128 page frames**, 2) **the second time with 256 page frames**, 3) **the third time with 512 page frames**, and 4) **the 4<sup>th</sup> time with 1024 frames**.

Sample usage is `prompt> memsim frame_size input.dat output.dat`  
where

1. `frame_size`: no. of page frames in the primary memory
2. `input.dat`: refers to the input file and
3. `output.dat`: the name of the output file.

## 2.2 Input details:

### 2.2.1 No. Of frames:

A single number indicates the number of frames in the primary memory. Your program will be tested with four primary memory sizes: 128, 256, 512, and 1024 frames.

### 2.2.2 input file:

The file contains the memory page request sequence string for only one process. Each line represents a page number accessed by the process. Assume a process can have up to 4096 pages, numbered 0-4095. This implies that no number in the reference stream will fall outside that range. There will be precisely 10,000 references in the page reference string. In other words, the input file has 10,000 lines. You may use my sample input file from the class website to test your program.

## 2.2. Output details:

The output file should print the page fault rates for each of the three-page replacement algorithms specified above, based on the input page reference string, at intervals of 2000 pages. The sample output format is shown below. For example, the column under 8000 lists the page fault rates for all algorithms on the first 8000 page references. Note that these are just sample values.

Page Replacement Algorithm Simulation (frame size = 128)						
Algorithm	Total page faults	Page fault rates				
		2000	4000	6000	8000	10000
FIFO	3387	0.338	0.298	0.387	0.320	0.339
LRU	3109	0.310	0.298	0.307	0.320	0.311
OPT	2500	0.276	0.298	0.287	0.220	0.250

## 3. Programming Requirements

- (1) Programming language: You have to use either C or C++ to develop your simulator on a Linux OS environment
- (2) Running Environment: Your program should be compiled on csegrid and tested without errors.

#### 4. Deliverables

- (1) Program source code includes all source program files, the Makefile, and the Readme.
- (2) Sample output
- (3) Analysis document: write up to about 2 pages to discuss the page fault ratio of three algorithms with respect to the frame size and how to decrease the page fault of FIFO, LRU, and Optimal. Include a graph of the page fault ratio for each frame size.

#### 5. Evaluation criteria

- (1) Deliverables
  - i. Submitting all required deliverables 2 points
  - ii. Sample Output 1 point
- (2) Completeness for each algorithm
  - i. FIFO 7 points
  - ii. LRU 8 points
  - iii. OPT 8 points
- (3) Report- Discussion and Analysis document 5 points
  - i. Draw a graph about the relationship between the page fault rate and the size of the frame – 2 points
  - ii. Discuss the relationship between the page fault rate and the size of the frame – 3 points

#### 6. How do I turn in my work?

Please do the following when you submit your programming assignment.

- Create a tar file that contains your written source code, makefile, and readme. DO NOT INCLUDE EXECUTABLES AND OBJECT FILES.
- Please use the following convention when you create a tar file
  - First 3 letters of your last name + last 4 digits of your student ID
  - For example, if a student's name is "Colorado Denver" and his ID is 999-34-5678, then her tar file name is "den5678.tar".
- Once you create the tar file, compress it with 'gzip'.
  - Do not know how to tar or zip your files?
    - Check "man tar" at the Linux prompt
- Upload **your gzipped file to the class Canvas.**