Lab Assignment 3 (30 points)

Simulating Page Replacement Algorithms

Due date: 2 pm on November 28 (Wednesday)

1. Goal of this programming assignment

The primary goal of this lab assignment is to exercise four paging replacement algorithms and evaluate the effectiveness of each algorithm which affects the performance of memory management.

2. Lab assignment description

Write a C or C++ program that simulates the operation of 4 page replacement algorithms used in a virtual memory management system:

1. FIFO (First-in-First-out) algorithm
2. Least Recently Used (LRU) algorithm
3. Most Frequently Used (MFU) algorithm
4. Optimal algorithm

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

2.1 Specification

In this lab assignment, you will be calculating the page fault rate of the each different page replacement algorithms for the given page reference string for one process only. Your program should accept three arguments, which are the number of page frames of the physical memory, the names of an input file, and 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 algorithms four times, the first time with 128 page frames, the second time with 256 page frames, the third time with 512 page frames, and the 4th 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:

It is a single number that indicates the number of frames in the primary memory. So a number 128 will indicate that there are 128 frames of primary memory for this process. Your program will be tested with four different size of primary memory: 128, 256, 512 and 1024 frames.

2.2.2 input file:

The file contains the memory page request sequence string for only one process. The each line represents a page number which is being accessed by the process. Assume a process can have up to 4096 pages, and the pages being numbered from 0-4095. This implies that no number in the reference stream will fall outside that range. There will be exactly 10,000 references in the page reference string. In other word, the input file has 10,000 lines. For your test, you may use my sample input file from the class web-site.

2.2. Output details:

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

==========

Page Replacement Algorithm Simulation (frame size = 128)

==========

		Page fault rates				
Algorithm	Total page faults	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
MFU	2967	0.291	0.283	0.281	0.250	0.260
Optimal	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 Linux OS environment
(2) Running Environment: Your program should be compiled at **csegrid** and be able to be tested without errors.

4. Deliverables

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

5. Evaluation criteria

(1)	Deliverables	
i.	Submitting all required deliverables	2 point
ii.	Sample Output	1 point
(2)	Completeness for each algorithms	22 points
i.	FIFO	5 points
ii.	LRU	6 points
iii.	MFU	5 points
iv.	OPT	6 points
(3)	Report- Discussion and Analysis document	5 points

i. Draw a graph about the relationship between the page fault rate and size of frame – 2 points
ii. Discuss the relationship between the page fault rate and size of frame – 3 points

6. How to turn in my work

Please do the followings 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
  - e.g.: If a student name is "Bill Clinton" and his ID is 999-34-5678, then his tar file name is "cli5678.tar".
- Once you create the tar file, and compress the tar file using 'gzip'.
  - Do not know how to tar or zip your files?
    - Check "man tar" at Unix prompt
- Upload **your gzipped file to class Canvas.**