COMP 3500 – Project #4

Submission Format:

1. Send an electronic package (e.g., a tar file) with documents as detailed at the end of this file "**Documents to turn in"** to the TA (zzl0014@auburn.edu)

Due Date:

1. Friday, April 26th, 2013, 11:50pm for all students

Advices:

1. Start early and be prepared for roadblocks.

Late Penalty: Note the hefty penalty for this last project.

1. 25% per day (round up)
2. Maximum one day.

Requirements:

1. Grasp the essence of page replacement algorithms.

Project Instruction:

1. The objective of this programming project is to simulate **only two of the four** page replacement algorithms we covered in class: Optimal, Clock, FIFO and LRU. Find the computer you have used for earlier projects. You need to evaluate your programs on the same machine.

2. You create two programs, one per algorithm, to calculate the page fault rate for a randomly generated reference string. These programs can be named as *optimal.c*, *clock.c, fifo.c* or *lru.c*. All programs should accept four arguments: *num_frames*, *num_pages*, the name of an input file, and the name of an output file. For example, if the program "optimal" is compiled from optimal.c, it should run as follows:
   $ ***optimal num_frames num_pages input.dat output.opt***

   where *num_frames* denotes the number of memory frames available, *num_pages* the range of accessed pages (from 1 to num_pages), and *num_pages* must be at least 8 times bigger than *num_frames*. *input.dat* is the name of an input file, which contains more than 1000 randomly generated page references, one number per line. *output.opt* is the name of the output file.

3. For each random number (RN) from the input.dat file, your program should convert it to the page number (PN) as (RN % num_pages + 1). For example, if an RN is 1234567 and num_pages is 100, then the PN is 68.

4. You should generate an input file with 1200 random numbers and then run the two programs with the input file. You can choose the other parameters (num_pages and num_frames) yourself. For each page fault, your program should store the corresponding RN to the output file, one number per line. At the end of the execution, your program should also report the overall page fault rate for the evaluated algorithm.

5. You must create a different program for each of the page replacement algorithms. One program that implements both two algorithms will have a penalty of 20%.

Documents to Turn in:

1. Two source programs named as optimal.c, clock.c, fifo.c or lru.c.
2. A Makefile that can compile two programs when typing “make”.
3. A README file describing how to compile and run your programs.
4. One sample intput.dat file you generated and two output files produced by the replacement algorithms.