**5th homework assignment for Computer Operating Systems – Paging**

This problem is worth 15 points.

Write a program that implements the FIFO, LRU, and Optimal page replacement algorithms presented in chapter 8 of your text. First generate a random page-reference string (this should be 20 entries long) where page numbers range from 0 to 9. Apply the random page-reference string to each algorithm, and record the number of page faults incurred by each algorithm. Implement the replacement algorithms so that the number of page frames goes from 1 to 7 and you must compute the page fault for each of these frame numbers. Record the number of page faults with each of these different page frames numbers and each of the different algorithms. Assume that demand paging is used. Remember to count the first time a page comes in, as this is a page fault in demand paging.

Then do the same procedure, except use the following page-reference string instead of the random one:

0,7,0,1,2,0,8,9,0,3,0,4,5,6,7,0,8,9,1,2 and then lastly do it with the page-reference string: 7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1

The sample output has the correct solutions for LRU and FIFO but nothing for Optimal.

Have separate clearly marked classes, functions, or methods for LRU, FIFO, and Optimal replacement algorithms. Also have comments within your code.

Make certain to have your name, date, assignment number, and a brief description of the program at the top of your main method.

Your output should be in the following format (repeated seven times, one for each number of page frames, and the set of seven repeated three times, one for each of the three page-reference strings):

For x page frames, and using string page reference string nnnnnnnnnnnnnn:

FIFO had ### page faults.

LRU had ### page faults.

Optimal had ### page faults.

Where ### is the number of page faults, x is the number of page frames, and nnnnnn is the page-reference string.

Note that I have placed a sample output in the assignment file that you can use to partially check your code and see the required output format a little more clearly. It does not have the output for the optimal case.

Remember to remove or turn off all of any debug output before you generate the output to send to me. Do not make the program interactive, it should run through the three strings with no human interaction.

Your name and the assignment number must appear at the top your output file from your program.

Place the following in the Dropbox for assignment #5: a zip file with a copy of your source code, output file from your run, and your observations file. Do not use screen shots for your data file.

Place your observations (how do the different methods compare for the page-replacement strings) in another attached text or word file and make certain that your name and the assignment number are at the top of your observations file. Remember to place the approximate number of hours you spent on this assignment in your observations file.

**Look for errors**:

* More page faults than items – not clearing a counter?
* Fewer page fault than different pages or no page faults (remember this is to use demand paging, so that for 1 page frame the two required strings will each have 20 page faults for the 1 page frame case).
* Hand check your results.
* Compare with the results in the sample output.
* Optimal has more page faults than one of the other methods (this should never happen).
* For one page frame, all methods should give the same results.

For additional information on paging, you can check.

www.tutorialspoint.com/operating\_system/os\_virtual\_memory.htm