

## Programming Assignment 4 (Virtual Memory) [50 pts]

COMP 4270: Operating Systems

Spring 2019

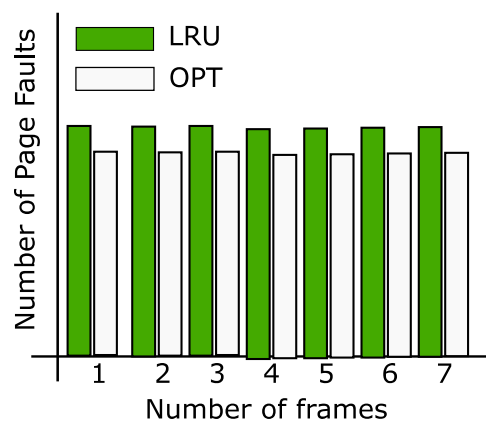
Due: 5/2/2019

No late submission will be accepted for this assignment.

### Task 1 – Implementation of Page Replacement Algorithms (25 pts)

Write a program that implements *the LRU algorithm* based on the counter-based implementation method, and *the optimal page replacement algorithm* presented in class. In this task, you will examine the impact of the number of frames on both the LRU and optimal algorithms by varying the number of frames. We will also investigate how the LRU algorithm performs in terms of the number of page faults in comparison with the optimal algorithm.

Generate a random page-reference string where page numbers range from 0 to 9. The length of the reference string should be 10 (i.e., consisting of 10 digits). Apply the random page-reference string to each algorithm, and record the number of page faults incurred by each algorithm. Vary the number of frames from 1 to 7 and repeat the experiment with the same reference string. Draw a graph that illustrates the results. The x-axis, and y-axis of the graph should be the number of frames, and the number of page faults, respectively. For example, your graph may look like this.



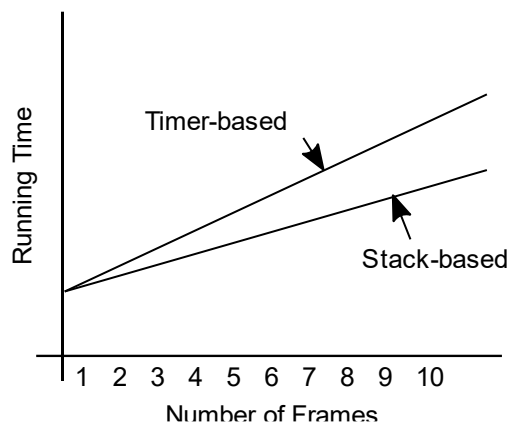
Briefly discuss your findings. Submit both the code and report that includes your graph.

## Task 2 – Performance Evaluation of the LRU Algorithm (25 pts)

In this task, you will evaluate the performance of the two different versions of the LRU algorithm that we covered in class, i.e., the timer-based and the stack-based algorithms. Specifically, you will measure and compare the running times of the timer-based LRU algorithm and the stack-based LRU algorithm. Since you have already implemented the LRU algorithm based on the timer-based method in Task 1, in this task, implement the algorithm based on the stack-based method in order to compare the running times of the two algorithms.

Generate a random-reference string with length 500 (i.e., consisting of 500 digits) where page numbers range from 0 to 20 (e.g., 0 2 18 12 20 1 ...). Measure the running times of both timer-based and stack-based LRU algorithms with the random-reference string as input. Vary the number of frames from 1 to 10 and repeat the experiment with the same reference string.

Here the running time is defined as the amount of time the algorithm takes to complete processing the random-reference string. Draw a graph that illustrates the results. The x-axis and y-axis of the graph should be the number of frames, and running time, respectively. For example, your graph would look like this.



Briefly discuss your findings and submit both your code and report that includes your graph.

## References

[1] Operating Systems Concepts 9<sup>th</sup> Edition by Silberscharz, Yale, and Gagne, Wiley