

# CS 3413

## Assignment 8

Due Date: November 23<sup>rd</sup>, 2020 at 9:30 am

---

---

**ASSIGNMENT IS TO BE COMPLETED INDIVIDUALLY BY ALL STUDENTS!**

**Your solution is to be written in C and submitted via D2L.**

This question consists of writing a program that can be used to measure how well each of the page replacement policies of First-in First-Out (FIFO), Least Recently Used (LRU) and Optimal work. Also, in this assignment we will deal with the oversized page table from the previous assignment! Your program will read from stdin a list of memory operation types and logical addresses. For example:

```
r 12345678
w 87654321
```

The first line indicates we are reading from logical address 12345678 and the second line indicates we are writing to address 87654321. For each line in the input, we are to translate each logical address to a physical address. If the page that we are trying to access is not loaded into a frame then we have to load it using one of the page replacement policies from above that is specified on the command line (-F for FIFO; -L for LRU; and -O for optimal). However, your simulation can only have  $n$  pages loaded in memory at any given time that is also specified on the command line. For example:

```
./a.out -L 10 < sample_addr_requests.txt
```

would process the contents of “sample\_memory\_requests” with 10 frames available and using the LRU replacement policy.

The goal is to simulate the steps involved in translating logical to physical addresses. Design your simulation with the following parameters:

- Addresses are 32 bits and provided as an unsigned integer.
- A page is  $2^{12} = 4$  kilobytes. (Note, this means a frame is also  $2^{12}$  in size).
- The page table and page directory are both 10 bits.
- Your process has been allocated  $n$  frames.  $n$  is a parameter to your program on the command line.

For the address translation you are to use the page directory, page table and page offset using the tree-based structure to reduce the size of the mapping from logical to physical. NO LARGE SINGLE PAGE TABLE!!!



Your program should print the physical address for each logical address to aid in testing!  
Also, you should print the following statistics at the end:

- How many minor page faults occurred?
- How many major page faults occurred?
- How many page hits occurred?
- How many pages are you currently storing in swap?