CS 3413

Assignment 7

Due Date: November 6th, 2020 at 9:30 am

ASSIGNMENT IS TO BE COMPLETED INDIVIDUALLY BY ALL STUDENTS!

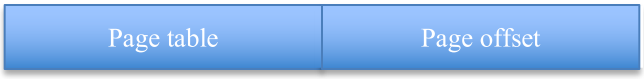
Your solution is to be written in C and submitted via D2L.

This question consists of writing a program that translates logical to physical addresses for a virtual address space of size 2^{32} bytes. Your program will read from stdin a list of logical addresses, translate each logical address to a physical address and then print the physical address that would be accessed in memory. However, your simulation can only have n pages loaded in memory at any given time! The goal is to simulate the steps involved in translating logical to physical addresses.

Design your simulation with the following parameters:

- A page is 2^{12} = 4 kilobytes. (Note, this means a frame is also 2^{12} in size).
- Therefore, the page table has 2^{20} entries. You can use a simple array to implement this table.
- 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 simple single level of page table mapping (page table is 20 bits, page offset is 12 bits)



Your program should print for each logical address read:

```
    logical address -> physical address
```

It should also keep track of the number of page faults that occur and print the statistic at the end.

Notes:

- Running your program with a sample file can be done as:
  ```
        ./a.out 10 < sample_logical_input.txt
  ```
- For this question you are to use a simple First-in First-out (FIFO) page replacement policy. Since you do NOT have to worry about writing to the page you DO NOT need to save the page when you replace it!
- Be aware of data type sizes!
- Hint … maybe in the next assignment we will change the page replacement algorithm ☺