German University in Cairo
Faculty of Media Engineering and Technology

CSEN 602-Operating Systems, Spring 2018
Mini-project 4: Memory Management

Grade: 5%

In this mini-project, you will use the C programming language to simulate:

1. the FIFO page replacement algorithm
2. a modified version of the second-chance page replacement algorithms.

For a description of both algorithms, you may refer to the textbook sections 3.4.3 and 3.4.4 (in the 4^{th} edition). You should modify the second-chance algorithm to evict non-modified pages first. For example, if multiple pages have a zero R-bit ~~and the same load_time~~[1], the page with zero M-bit should be evicted first.

Implementation overview

To simplify the tracing, we will assume the maximum number of page frames available is 5, we will also assume a single process is running.

You will use a struct to represent each page that is currently in memory. The page attributes to be kept in the trust are shown below. You will also use a linked list to keep track of the order of loading pages into memory.

- page ID,
- Referenced (R) bit,
- Modified (M) bit.

Your simulation should run against an input file that lists the sequence of page accesses. The format of the file is as follows:

```
access_time    page_ID    access_type
...
```

[1]Implementations that took load_time into consideration will still be accepted.

The access_time is given in milliseconds. The access_type can be either a read or a write access. Please note that if a page is accessed for reading, its R bit is set. If a page is accessed for writing, its M-bit is set. The R bit is reset for all pages at the end of each quantum. Assume the quantum length is 20 milliseconds and that a clock interrupt takes place immediately after the end of 19^{th} millisecond and every 20 milliseconds afterwards. Clock interrupts can be simulated using simple conditions, you are not required to use timers this time.

You may assume that the memory is initially free, hence the the first 5 page are loaded into memory immediately, and without the need to evict any pages. When a modified page is evicted, it is written back to disk.

Output Format

Your simulation should print out the following:

- A notification of each page fault, its time, id of evicted page, id of loaded page.
- If the evicted page is to be written back to disk.
- The time of each clock interrupt.