

**CS 5323 – Operating Systems**  
**Assignment 4**  
**Due: April 18, 2022, 11:59 pm**

Page replacement algorithms are an essential part of completing the illusion of infinite virtual memory to the user by masking the need for reading pages from the disk for every execution. In this assignment, you will implement two algorithms First In First Out (FIFO) and Least Recently Used (LRU) and measure their performance on three benchmark datasets that are actual traces of the disk operations performed when using real-world applications such as unzipping a file using bzip and compiling a C program using gcc. Each file will have two values, the first correspond to the address of the frame being referenced and the second corresponds to the type of operation (**Read** or **Write**). Your program must take in three arguments – the file name, the algorithm type and the number of frames in memory for the process. Your output must be the number of disk reads and disk writes performed and the contents of the frames at the end of the simulation.

Example usage:

```
./virtual_memory <trace file> <nframes> <fifo|lru>
```

You can use the test.trace file, which provides the frame number and operation type, as a sanity check. Your output will be:

```
Contents of page frames
1  0  7
Number of Reads: 12
Number of Writes: 0
```

**Things to remember:**

1. You must implement each of the algorithms using a proper data structure such as linked list or bitmap or hash table. In your report you must be able to defend your choice. You should be able to provide the exact contents of the frames in the memory at any given time.
2. Each algorithm requires you to select a victim page using some heuristic. Make sure you have a mechanism to implement this heuristic.
3. A disk write only occurs when there is an update on a page in memory to make it “dirty”.

The submission will need to contain the following:

1. Your code file.
2. A README file describing how to compile and run your program. Any requirements that we need to be aware of.
3. A report containing details about your implementation and your analysis of each algorithm and how the performance changes as you increase the number of frames in memory. You must provide details about how you implemented each algorithm and what data structures were used and why it was used. You must also provide an observation as to whether Belady’s anomaly is observed and when it was observed.

Make sure your name is in the code in the comments! Again, note this is an individual project and must be your own code. If you use any other code, it must be acknowledged in comments.