

Programming Assignment 2: Page Replacement Algorithm

CSC 4103: Operating Systems, Spring 2017

Due: March 26th, Sunday (by 11:59 PM)

Total Points: 10

Objective

To implement page replacement algorithms used by operating systems.

Background

If total memory requirements exceed the physical (main) memory, it is necessary to replace pages from memory to free frames for new pages. Two replacement algorithms are Least-Recently-Used (LRU) and its approximation, the CLOCK algorithm. Refer to the lecture notes and the textbook for more information about the two algorithms.

Programming Task

Write a program (in C/C++ or Java) that implements the LRU and CLOCK page-replacement algorithms. Use the page reference sequence in the provided input file on our course website (<http://www.csc.lsu.edu/~fchen/class/csc4103-sp17/assignments/pageref.txt>). Report the number of page faults and latency incurred by each algorithm (see details below). Implement the replacement algorithms so that the number of available page frames can vary as specified (e.g., 20). Assume that demand paging is used, i.e., page frames are initially free.

Requirements:

- (1) The program should accept three input parameters to specify:
 - a. The algorithm ("LRU" or "CLOCK")
 - b. The cache size (the number of page frames for allocation)
 - c. The input file that contains the page-reference string.

Example: `$./replace CLOCK 20 pageref.txt`

Note: A sample input file can be downloaded from our course website. The program should be tested using the supplied input file. The format of the input file: Each line has two fields. The first field is either "R" or "W", meaning a read or a write to the page, respectively. The second field is the page number. Thus, each line of the input file gives a reference to a page, either read or write.

- (2) The program should generate the following output information:
 - a. The total number of page references.
 - b. The total number of page misses.
 - c. The total number of time units for page misses.
 - d. The total number of time units for writing back the dirty (modified) page.
- (3) The program should calculate the "cost" of page faults in terms of time units. In particular, upon a page hit (the page is found in memory), the reference cost is 0; upon a page miss (the page is not found in memory), it takes 5 time units to load the page in. When deciding to evict a victim page, if the victim page is modified (i.e., the page has been written in memory), it takes 10 time units to write the victim page out.

- (4) Your submission should include an README file to clearly explain how to compile and run your code, such as the commands, arguments, and expected input and output, and any necessary information. In the README file, please also provide your full name, LSU ID, and email address.

Before you submit:

- (1) Compile and test-run your code on the classes.csc.lsu.edu server. Your code should be written in programming languages as specified above, and be compilable and runnable in the classes server's Linux environment. Windows code will NOT be accepted.
- (2) Submit your work as instructed before the deadline and verify your submission, which will display your submission date/time.
- (3) Late submissions will be penalized by 10% per day late and no more than 3 days.

Submitting Your Work

All files you submit must have a header with the following:

Name:	Your Name (Last, First)
Project:	PA-2 (Page Replacement Algorithms)
File:	filename
Instructor:	Feng Chen
Class:	cs4103-sp17
LogonID:	cs4103xx

You need to use the server “**classes.csc.lsu.edu**” to work on the assignment. You can login to your account in the server using SSH. Create a directory **prog2** (by typing **mkdir prog2**) in your home directory in which you create your program or source code.

Note that you should NOT include any directory in **prog2**.

Make sure that you are in the **prog2** directory while submitting your program. Submit your assignment to the grader by typing the following command:

```
~cs4103_chf/bin/p_copy 2
```

This command copies everything in your prog2 directory to the grader's account. Check whether all required files have been submitted successfully:

```
~cs4103_chf/bin/verify 2
```