

Programming Assignment 2: Page Replacement Algorithm

CSC 4103: Operating Systems, Spring 2020

Due: April 16th, Thursday (by 11:59 PM)

Total Points: 10

Instructions: Compile and test-run your code on the classes server. Submit your work as instructed and verify your submission. The verify command will display your submission date/time. Include your name, email, and classes login ID in all source code files. **Late submissions** will be penalized at the rate of 10% per day late and no more than 3 calendar days late.

How to Submit (Online Class): You need to complete your submission in two steps.

(1) Submit your code to the classes server. In order to connect to the classes server, you will need to use VPN. The following link contains a brief instruction for using VPN:

<https://www.csc.lsu.edu/~fchen/class/csc4103-sp20/files/VPN.pdf>

(2) Submit a simple note on Moodle. This note should include necessary information, such as your name, classes login, and myLSU email.

Note: The UNIX time of your submission on the classes server will be used to determine the exact submission time. So the submission time is the time when you upload your code.

Objective

To implement a page replacement algorithm used by operating systems.

Background

If the total memory demand exceeds the physical memory capacity, the operating system needs to replace pages from memory based on the locality principle. The CLOCK (i.e., Second-Chance) algorithm is an approximation of the well-known Least Recently Used (LRU) algorithm, which evicts the least recently accessed pages. The CLOCK algorithm uses a reference bit (RB) associated with each page to efficiently estimate the recency of page references and identify the victim pages for eviction.

Programming Task

Write a program that implements the CLOCK page replacement algorithm. Use the page reference sequence in the input file to drive your program. Report the number of page faults and time cost incurred by the page faults (see details below). The number of available page frames can be set as specified by an input parameter of your program. Assume that demand paging is used and the page frames are initially free. More details about the CLOCK algorithm can be found in the lecture notes (Lecture #9, Page 40) and the textbook.

Programming Language

C/C++ or Java.

Detailed Requirements

- (1) The program should accept the following input parameters to specify:
 - a. The memory size (the number of available page frames)
 - b. The input page reference file that contains the page references.
 - c. Hit cost (time needed to access the page resident in memory)

- d. Swap-in cost (time needed to swap a page from swap space into memory)
- e. Swap-out cost (time needed to swap a page out of memory to swap space)

Example: `$./clock 20 pageref.txt 1 10 20`

This example command means the system has 20 free physical page frames, using references in pageref.txt as input, accessing a page in memory takes 1 time unit, swapping in a page takes 10 time units, swapping out a page takes 20 time units.

Page Reference File Format: 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. Each line of the input file thus specifies a page reference, either read or write. Two sample input reference files (pageref.txt and pageref-small.txt) are posted online.

- (2) The program should generate the following output information:
 - a. The total number of page references.
 - b. The total number of page faults on READ.
 - c. The total number of page faults on WRITE.
 - d. The total number of time units for accessing pages in memory.
 - e. The total number of time units for swapping in pages.
 - f. The total number of time units for swapping out pages.
- (3) Page fault costs: The program needs to calculate the “cost” of page faults in terms of time units. In the example above, upon a page hit (the page is found in memory), the reference cost is 1 time unit; upon a page miss (the page is not found in memory), it takes 10 time units to load (swap in) the page to memory, and then access the page in memory. When evicting a victim page, if the victim page is modified (i.e., the page has been written in memory), it takes 20 time units to write (swap out) the modified victim page to the swap space. So you will also need a modify bit to keep track of the page status. We assume the pages are all initially in the swap space and the memory is free.
- (4) Your submission should include an README file to clearly explain how to compile and run your code, such as the command, parameters, and the expected input and output, and any other 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 the programming languages as specified above, and be compilable and runnable in classes server’s Linux environment. Windows code will NOT be accepted.
- (2) Submit your work as instructed before the deadline and verify your submission as instructed below. The verify tool will display your submission date/time.
- (3) Late submissions will be penalized by 10% per day late and no more than 3 days late.

Submitting Your Work

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

Name:	Your Name (Last, First)
Project:	PA-2 (Page Replacement Algorithm)
File:	filename
Instructor:	Feng Chen

Class: cs4103-sp20
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, and then you create your program or source code in **prog2**.

Note that do 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