CSCI 3120 Operating Systems

Bonus Assignment: Page Replacement Algorithm Evaluation

Due: 16:00, Dec. 13, 2019

- **Teaching Assistants:**
  - Lauchlan Toal (lc790935@dal.ca),
  - Patricia Kibenge-MacLeaod (p.kibenge@dal.ca)
  - Hui Huang (huihuang@dal.ca)
- **Help Hours at CS Learning Center (2nd Floor of Goldberg Building):**
  - Tuesdays: 1pm-3pm, Hui Huang
  - Thursdays: 2pm-4pm, Patricia Kibenge-MacLeaod
  - Fridays: 12pm-2pm, Lauchlan Toal

1. Assignment Overview

In this assignment, you need to write a C program to implement a page replacement algorithm (specifically, FIFO) and evaluate its performance in terms of the number of resulting page faults.

2. Important Note

There is a zero-tolerance policy on academic offenses such as plagiarism or inappropriate collaboration. By submitting your solution for this assignment, you acknowledge that the code submitted is your own work. You also agree that your code may be submitted to a plagiarism detection software (such as MOSS) that may have servers located outside Canada unless you have notified me otherwise, in writing, before the submission deadline. Any suspected act of plagiarism will be reported to the Faculty's Academic Integrity Officer in accordance with Dalhousie University's regulations regarding Academic Integrity. Please note that:
1) The assignments are individual assignments. You can discuss the problems with your friends/classmates, but you need to write your program by yourself. There should not be much similarity in terms of coding.
2) When you refer to some online resources to complete your program, you need to understand the mechanism and write your own code. In addition, you should cite the sources via comments in your program.

3. Detailed Requirements

1) Overview: Demand Paging is a widely-used memory management scheme. With demand paging, only part of the pages belonging to a program are loaded into physical memory, other

pages are kept on the hard drive. When a page on the hard drive needs to be used, a free frame in physical memory is found and this page is moved into the free frame. If there is no free frame, there will be a page fault. In this scenario, a page replacement algorithm is used to find a victim frame, then the page is moved into the location of the victim frame.

There exist a variety of different page replacement algorithms, whose performance varies in terms of resulting page faults. We can evaluate an algorithm by running it with a reference string and computing the number of resulting page faults. The lower the number of page faults, the better the algorithm. In this assignment, you need to write a C program to implement the first-in, first-out (FIFO) algorithm and evaluate its performance in terms of the number of resulting page faults.

2) Input File: To evaluate the performance of a page replacement algorithm, we need to know both the number of frames in physical memory and the reference string. These two parameters are provided via a file named "Input.txt", which is in the same directory as your program. The first row of this file includes an integer that indicates the number of frames in physical memory. The second row of this file includes the reference string: a series of page numbers that are separated by commas. A sample "Input.txt" is provided in this assignment. This sample file includes the following number of frames and reference string (note that the TA will use a different file to mark your assignment):

```
3
7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1
```

3) Output Output: The output of your program should be placed in a file named "Output.txt", which is in the same directory as your program. "Output.txt" should be composed of two parts.

The first part includes a series of rows. Each row shows the pages that have been loaded into the frames after each page access. In each row, the paper numbers should be separated using tabs; if a frame does not include a page (i.e. the frame is empty), then a letter "N" should be used as a placeholder for the frame. Obviously, the number of rows in the first part is equal to the number of page numbers in the reference string. For the sample "Input.txt", the first six rows should look like:

```
7	N	N
7	0	N
7	0	1
2	0	1
2	0	1
2	3	1
```

Note that, for the sample "Input.txt", when page 0 is loaded into physical memory for the first time, a page fault occurs and it leads to the second row in the example output. When page 0 is loaded into physical memory for the second time, it does not lead to a page fault because page 0 is already in physical memory. However, for clarity purposes, the current pages in physical memory are still displayed in the output. Specifically, the fifth row in the example output shows the pages in physical memory after page 0 is loaded into physical memory for the second time.

The second part of "Output.txt" should indicate the number of resulting page faults. For the sample "Input.txt", this part should look like:

Number of Page Faults: 15

Note that:

a) The detailed analysis of the scenario corresponding to the sample "Input.txt" is included in the lecture notes for Chapter 10. You can refer to the lecture notes to have a deep understanding of FIFO page replacement algorithm.
b) For clarity purposes, there should be an empty line between the first and the second part of "Output.txt".

4) Additional Information:

a) For simplicity, we assume that all the page numbers in the reference string belong to the same program.
b) We assume that all the frames in physical memory are initially empty.
c) The number of frames in physical memory is in the range of 1 to 5.
d) The reference string includes a series of page numbers. There could be 1 to 20 page numbers in the reference string.
e) Each page number in the reference string is an integer, which is in the range of 1 to 10.

5) Readme File: You need to complete a readme file named "Readme.txt", which includes the instructions that the TA could use to compile and execute your program on bluenose.

6) Submission: Please pay attention to the following submission requirements:

a) You should place "Readme.txt" in the directory where your program files are located.
b) Your program files and "Readme.txt" should be compressed into a zip file named "YourFirstName-YourLastName-ASN-Bonus.zip". For example, my zip file should be called "Qiang-Ye-ASN-Bonus.zip".
c) Finally, you need to submit your zip file for this assignment via brightspace.

Note that there is an appendix at the end of this document, which includes the commands that you can use to compress your files on bluenose.

4. Grading Criteria

The TA will use your submitted zip file to evaluate your assignment. The full grade is 20 points. The details of the grading criteria are presented as follows.

- "Readme.txt" with the correct compilation/execution instructions is provided [1 Point]
- The number of frames in physical memory and the reference string are retrieved from "Input.txt", which is in the same directory as your program. [1 Point]

- The output is placed in “Output.txt”, which is in the same directory as your program. [1 Point]
- FIFO page replacement algorithm is properly implemented. [8 Points]
- The first part of “Output.txt” is correct. [6 Points]
- The second part of “Output.txt” (i.e. the number of resulting page faults) is correct. [2 Points]
- Proper programming format/style (e.g. release the memory that is dynamically allocated when it is not needed any more; proper comments; proper indentation/variable names, etc.). [1 Point]

Please note that when “Readme.txt” is not provided or “Readme.txt” does not include the compilation/execution instructions, your submission will be compiled using the standard command `gcc -o A5 A5.c (or your filename)`, and you will receive a zero grade if your program cannot be successfully compiled on bluenose.

5. **Academic Integrity**

At Dalhousie University, we respect the values of academic integrity: honesty, trust, fairness, responsibility and respect. As a student, adherence to the values of academic integrity and related policies is a requirement of being part of the academic community at Dalhousie University.

1) What does academic integrity mean?

Academic integrity means being honest in the fulfillment of your academic responsibilities thus establishing mutual trust. Fairness is essential to the interactions of the academic community and is achieved through respect for the opinions and ideas of others. Violations of intellectual honesty are offensive to the entire academic community, not just to the individual faculty member and students in whose class an offence occur (See Intellectual Honesty section of University Calendar).

2) How can you achieve academic integrity?

- Make sure you understand Dalhousie’s policies on academic integrity.
- Give appropriate credit to the sources used in your assignment such as written or oral work, computer codes/programs, artistic or architectural works, scientific projects, performances, web page designs, graphical representations, diagrams, videos, and images. Use RefWorks to keep track of your research and edit and format bibliographies in the citation style required by the instructor. (See http://www.library.dal.ca/How/RefWorks)
- Do not download the work of another from the Internet and submit it as your own.
- Do not submit work that has been completed through collaboration or previously submitted for another assignment without permission from your instructor.
- Do not write an examination or test for someone else.
- Do not falsify data or lab results.

These examples should be considered only as a guide and not an exhaustive list.

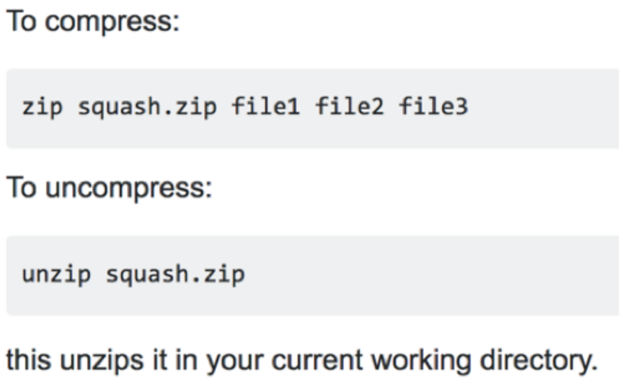
3) What will happen if an allegation of an academic offence is made against you?

I am required to report a suspected offence. The full process is outlined in the Discipline flow chart, which can be found at: http://academicintegrity.dal.ca/Files/AcademicDisciplineProcess.pdf and includes the following:
a. Each Faculty has an Academic Integrity Officer (AIO) who receives allegations from instructors.
b. The AIO decides whether to proceed with the allegation and you will be notified of the process.
c. If the case proceeds, you will receive an INC (incomplete) grade until the matter is resolved.
d. If you are found guilty of an academic offence, a penalty will be assigned ranging from a warning to a suspension or expulsion from the University and can include a notation on your transcript, failure of the assignment or failure of the course. All penalties are academic in nature.

4) Where can you turn for help?

- If you are ever unsure about ANYTHING, contact myself.
- The Academic Integrity website (http://academicintegrity.dal.ca) has links to policies, definitions, online tutorials, tips on citing and paraphrasing.
- The Writing Center provides assistance with proofreading, writing styles, citations.
- Dalhousie Libraries have workshops, online tutorials, citation guides, Assignment Calculator, RefWorks, etc.
- The Dalhousie Student Advocacy Service assists students with academic appeals and student discipline procedures.
- The Senate Office provides links to a list of Academic Integrity Officers, discipline flow chart, and Senate Discipline Committee.

Appendix: How to Use Zip and Unzip on Bluenose



To compress:

```
zip squash.zip file1 file2 file3
```

To uncompress:

```
unzip squash.zip
```

this unzips it in your current working directory.