CprE 308 Laboratory 6: Memory Management

Department of Electrical and Computer Engineering
Iowa State University

Spring 2019

1 Submission

Include the following in your lab report:

- Your full name and lab section in the upper right corner of the first page in large print.
- A summary of what you learned in the lab session. This should be no more than two paragraphs. Try to get at the main idea of the exercises.
- The output of the given code when everything in main is uncommented.
- Submit your summary, write-ups, and source code on Canvas. Your submission should be a single archive called `lab07-lastname-section.tar.gz`, where lastname is your last name and section is your section letter.

2 Introduction and background

In this assignment, you will write a program that simulates three virtual memory page replacement algorithms: Optimal (OPT), Least Recently Used (LRU) and First-In-First-Out (FIFO). Here is a brief description of the three algorithms that you are going to implement in your code. More details can be found in the text book.

OPT: when a page needs arrives into the memory, the OS replaces the page whose next use will occur farthest in the future. Note that this algorithm cannot be implemented in the general-purpose OS, because the pattern of future page references is usually unknown.

FIFO: This is the simplest page-replacement algorithm. When a page needs to be replaced, the page that is chosen is the one which arrived earliest into memory, among all the page frames currently in memory. FIFO performs poorly in practice since it ignores the usage history of a page and only focuses on the time of arrival. Thus, it is rarely used in its unmodified form.

LRU: The page to be swapped out is the page whose most recent access time is the earliest. LRU works on the idea that pages that have been used in the past few instructions are most likely to be used in the next few instructions too.

3 Lab Description

We assume an application that uses a total of 128 pages, some of which are on disk, and some in memory. The memory has 16 page frames, each of which can hold a single page. Initially, the memory is empty, and any page access will result in a **page fault**, which means that the page is not in the memory and must be collected from the disk. After that, for each page access, the OS first checks the memory. If the page is in memory, then it is a **page hit** where the OS access the page from the memory. If the page is not in the memory, then it results in a page fault, where the page is retrieved from the disk. On a page fault, if the memory is not full (there is an empty page frame), then the page is placed in an empty page frame. If the memory is full (i.e. all 16 page frames are occupied), then the OS should find a page to be replaced. The replacement algorithms that we consider are OPT, LRU, and FIFO: your task is to implement these three algorithms.

We maintain a structure for each page frame, which contains the (1)ID of the page in this page frame (2) most recent access time of the page, and (3)arrival time of the page. On a page hit, the OS updates the time of most recent access. On a page fault, the OS brings the new page to the memory and records the last access and memory arrival times to the same value which is the current time. The time is the index of the page in the input file. Here is the structure representing a page frame.

```
typedef struct {
    int page_id;
    int time_of_access;
    int time_of_arrival;
} PageFrame;
```

3.1 Inputs

The inputs to the program are page accesses sequences; each sequence consists of 10000 memory pages accesses. The sequence contains the IDs of the pages that are accessed by the application. The time of the access is equal to the index of the entry.

1. First sequence will be sequential (SEQ). The application accesses the pages 1,.. ,128 in order, and this sequence is then repeated.
2. The second sequence is random (RAN): there will be 10000 page IDs, each of them a random number between 1 and 128.
3. The third sequence (LR) is constructed to obey a "locality of reference pattern". With probability 0.9 the next page to be accessed is one of the 5 pages that were recently been accessed, and with probability 0.1 a page among the 128 pages will be accessed.

3.2 Task

Study the code and understand the different parts involved. Then implement the two functions "PRAlgo_OPT" for the optimal algorithm, and "PRAlgo_LRU", for the "Least Recently Used" algorithm. One replacement algorithm, "First in First Out" is already implemented in the function "PRAlgo_FIFO". You also need to implement your own custom algorithm – "PRAlgo_CUST". This algorithm should be designed by you and the goal is to beat the LRU algorithm on the Locality of reference access sequence, you may use whatever information is available to the PRAlgo_CUST function to achieve this.

This code contains three parts. The first part is the generation of the different access sequences, which we have written. The second part is the code which reads these sequences of memory pages and checks whether each page is in the memory or not. Finally, a function for deciding which page in memory should be replaced; this function is called only in case there is a page fault AND there is no space for additional pages in memory. You should write the code for the third part.

3.3 Output

For each of the given three sequences, should show the number of page faults using each of the four algorithms. Therefore, we expect 12 numbers, three per algorithm.

3.4 Grading

The grader will check your code on three sequences and see if your code gives the correct 9 results for the FIFO, LRU, and OPT algorithms, and whether your CUST algorithm beats LRU on the Locality of Reference access sequence.

The grading criteria is as follows:

- Summary: 10pts
- Compile: 5pts
- Code commented: 5pts
- Replacement algorithms' implementation:

```
OPT: 18pts
LRU: 18pts
CUST: 18pts
```

Testing on three sequences:

- Correct table generation for OPT, LRU, and FIFO: 18pts
- Beats LRU on the Locality of Reference access sequence: 8pts

The total points are 100pts