CS 420
HOMEWORK ASSIGNMENT H5

DUE DATE: Wednesday, November 7

Problem 1 is worth 60 points and the other four problems are worth 10 points each.

1. Write a program that implements both the FIFO and LRU page-replacement algorithms presented chapter 9 of the textbook.

   First, generate a random page- reference string (can use `PageGenerator`, see below) where page numbers range from 0 to 9. Apply the random page-reference string to each of the two algorithms, and record the number of page faults incurred by each algorithm. Implement the replacement algorithms so that the number of page frames can be determined by the user. Assume that demand paging is used.

   Your algorithms will be based on the abstract class shown below (source code available on Canvas with this assignment).

```
public abstract class ReplacementAlgorithm
{
   // the number of page faults
   protected int pageFaultCount;
   // the number of physical page frame
   protected int pageFrameCount;
   // pageFrameCount - the number of physical page frames
   public ReplacementAlgorithm(int pageFrameCount) {
        if (pageFrameCount < 0)
             throw new IllegalArgumentException();
        this.pageFrameCount = pageFrameCount;
        pageFaultCount = 0;
   }
   // return - the number of page faults that occurred.
   public int getPageFaultCount() {
        return pageFaultCount;
   }
   // int pageNumber - the page number to be inserted
   public abstract void insert(int pageNumber);
}
```

   Design and implement two classes—`LRU` and `FIFO`—that extend `ReplacementAlgorithm`. Each of these classes will implement the `insert()` method, one class using the LRU page-replacement algorithm and the other using the FIFO algorithm.

The source code for the following classes can be downloaded from the Canvas assignment:

a. `PageGenerator` - a class that generates page-reference strings with page numbers ranging from 0 to 9. The size of the reference string is passed to the `PageGenerator` constructor. Once a `PageGenerator` object is constructed, the `getReferenceString()` method returns the reference string as an array of integers.

b. `Test` - used to test your FIFO and LRU implementations of the `ReplacementAlgorithm` abstract class. Test is invoked as

```
java Test <reference string size> <number page frames>
```

For example, to run Test using a reference string of 30 values and 5 physical memory frames, you would use:

```
java Test 30 5
```

Once you have implemented the FIFO and LRU algorithms, experiment with a different number of page frames for a given reference string and record the number of page faults. Does one algorithm perform better than the other? For a given reference-string size, what is the optimal number of page frames?

Create an MS-DOS batch file named `replace.bat` that will execute the program. Store all files necessary for the execution of your application in a folder named `Prob1` that you submit electronically for the assignment, as instructed below. Be sure the output of the program clearly indicates what the program is doing.

2. Consider a pure paging system that uses 32-bit addresses (each of which specifies one byte of memory), contains 128MB of main memory and has a page size of 8KB.

   a. How many page frames does the system contain?
   b. How many bits does the system use to maintain the displacement, d?
   c. How many bits does the system use to maintain the page number, p?

3. Consider a pure paging system that uses three levels of page tables and 64-bit addresses. Each virtual address is the ordered set v = (p, m, t, d), where the ordered triple (p, m, t) is the page number and d is the displacement into the page. Each page table entry is 64 bits (8 bytes). The number of bits that store p is n_p, the number of bits that store m is n_m and the number of bits to store t is n_t.

   a. Assume $n_p = n_m = n_t = 18$.
      i. How large is the table at each level of the multi- level page table?
      ii. What is the page size, in bytes?

b. Assume $n_p = n_m = n_t = 14$.
   i. How large the table at each level of the multi- level page table?
   ii. What is the page size, in bytes?

c. Discuss the trade-offs of large and small table sizes.

4. Discuss how fragmentation manifests itself in each of the following types of virtual memory systems.

   a. segmentation
   b. paging
   c. combined segmentation/paging

5. Consider the following experiment and <u>explain the observations</u>.

   A process is run by itself on a paged machine. It begins execution with its first procedure page. As it runs, the pages it needs are demand-paged into available page frames.

   The number of available page frames far exceeds the number of pages in the process. But there is a dial external to the computer that allows a person to set the maximum number of page frames the process may use.

   Initially, the dial is set at two frames, and the program is run to completion. The dial is then set at three frames, and again the program is run to completion. This continues until the dial is eventually set to the number of available page frames in main memory, and the process is run for the last time. For each run, the run time of the process is recorded.

   Observations:

   As the dial is changed from two to three to four, the run times improve dramatically. From four to five to six, the run times still improve each time, but less dramatically. With the dial settings of seven and higher, the run time remains essentially constant.

Either type your solutions or print legibly. Solutions that cannot be easily deciphered are incorrect!

General Instructions:

- First problem counts for 40% of the score.

- Homework submissions must be prepared using computer document preparation applications such a word processor or similar editor. Handwritten solutions are not acceptable – neatness, readability and grammar count!

- Homework submissions will be clearly marked with the student’s name, date and assignment identification at the top of the first page.

- All homework is to be completed by each student individually and represent that student's original, unassisted work. Any material copied in any way from other sources must be clearly identified and attributed.

- The non-programming problem solutions and the source code for programs are printed on paper and submitted at the start of class on the due date.

Programming Instructions:

- Put a block of comments at the beginning of every physical file containing program source code that includes your name, the course name and number, information identifying what functions the program is designed to perform, and instructions how to execute the program. (required)

- For the programming problems, place the Java source files, class files, batch files and all other files necessary to execute each program you write into a separate Windows folder that is named `Prob1`, `Prob2`, etc.

- Place a batch file in each problem folder that will execute the program in that folder.

  - For example, The `prob1.bat` file is a text file with the following format (where `TheProgam` is the name of the class file containing the `main()` method):

    ```
    java TheProgam
    pause
    ```

  - Double-clicking the `prob1.bat` file should cause your program to execute.
  - Don't forget to put the `pause` command in the last line

- Place the programming problem folders into a zip file. Create the zip file so that the folder structure (path) is also recorded by selecting the "save full path info" option. Use your email ID as the zip file name. Example of file structure for submission:

  - Zip file named `brixiusn.zip` contains folders named `Prob1`, `Prob2 and Prob3`. Each folder has source code, class files and any required data files or batch files. The zip file records the path information for each file.

- Submit the zip file in the course Canvas site using the assignment submission capability in the same location you accessed this assignment information. You can also add comments when you submit a file for the assignment.

- Each assignment must be submitted on Canvas by the start of class on the day the assignment is due.

- Submit only one zip file with your entire assignment.

- If you have already submitted a homework assignment and then decide you must resubmit the assignment before it is due, you can submit another zip file to replace previous submissions. You can also use comments with the submission to further explain your submission to the grader. You may resubmit as many times as you find necessary before the assignment due date.

- DO NOT FORGET TO ALSO SUBMIT ANY NON-PROGRAMMING PROBLEM SOLUTIONS AND SOURCE CODE ON PAPER AT THE START OF CLASS ON THE DUE DATE.