Wanner HernandezR

CMSC 412 Week 8

Final Project

Professor Williams, Gregory

May 9, 2021

**Assignment:**

**Final Project CMSC – OPERATING SYSTEMS**

Design and implement a **Demand Paging** virtual memory simulator!  
It must be a text based application (NOT a GUI based one).  
You can use the **C/C++** or **Java** programming language.  
The following algorithms must be implemented: **FIFO**, **OPT, LRU** and **LFU**.  
The application must simulate the execution of each of these algorithms on a hypothetical computer having only **N** physical frames (numbered from **0 to N-1, N<8**), assuming that the single process that is running has a virtual memory of **ten** frames (numbered from **0 to 9**). The number N should be a number provided in the command line as an argument.

The algorithms will be simulated based on a reference string (a sequence of pages that are to be accessed) that will be either read from the keyboard or randomly generated.

***THE SIMULATION MUST FOLLOW THE ANIMATED EXAMPLES FROM THE ONLINE MODULE 3 AS CLOSE AS POSSIBLE IN ALL ASPECTS !!!***

The program should be menu-based and the menu will keep the user in a **loop** containing the following options:

**0 – Exit**

Will exit the program

**1 – Read reference string**

A reference string will be read from the keyboard and stored in a buffer. Each value of the reference string will be verified and validated (or rejected).  
Using option 1 again will result in overwriting the old reference string.  
**2 – Generate reference string**

A reference string will be randomly generated; the length of the reference string will be given by the user interactively. The string will be stored in a buffer.  
Using option 2 more than once will result in overwriting the old reference string.  
**3 – Display current reference string**

Will display the stored reference string; if there is no reference string stored yet, an error message will be displayed.  
**4 – Simulate FIFO**Will simulate the step by step execution of the FIFO algorithm using the stored reference string; if there is no reference string stored yet, an error message must be displayed.

The user will press a key after each step of the simulation to continue the simulation. The total number of faults will be displayed at the end of the simulation.  
**5 – Simulate OPT**Will simulate the step by step execution of the OPT algorithm using the stored reference string; if there is no reference string stored yet, an error message must be displayed.

The user will press a key after each step of the simulation to continue the simulation. The total number of faults will be displayed at the end of the simulation.  
**6 – Simulate LRU**Will simulate the step by step execution of the LRU algorithm using the stored reference string; if there is no reference string stored yet, an error message must be displayed.

The user will press a key after each step of the simulation to continue the simulation.

The total number of faults will be displayed at the end of the simulation.

**7 – Simulate LFU**

Will simulate the step by step execution of the LFU algorithm using the stored reference string; if there is no reference string stored yet, an error message must be displayed.  
The user will press a key after each step of the simulation to continue the simulation. The total number of faults will be displayed at the end of the simulation.

Selecting a different option will result in an error message but the user will NOT exit the loop!

**Description:**

It acts as a Demand Paging Simulator based off of Module 3 from the reading. It uses the exact wording from the simulations in the reading and will also print out the different algorithms in an easy to read format. The user will be prompted with several options, and when in a simulation will be prompted to continue or exit.

**File:**

* Driver Class:
* FinalProject.java
* Algorithm Classes:
* Fifo.java
* OPT.java
* LRU.java
* LFU.java
* Utility Classes
* outputTable.java
* printClass.java
* HelperClass.java

**Instructions/Guide:**

This project simulates a Demand Paging Virtual Memory machine in Java. It allows the user to input a number of variables representing pages into a reference string (a virtual representation of a sequence of pages), then swaps these pages into virtual representations of physical frames, using various algorithms that are similarly used in demand paging. The four algorithms used are First In First Out(FIFO), Optimum page replacement(OPT), Least Recently Used(LRU), and Least Frequently Used(LFU). There are 5 physical frames labelled 0-4 and 10 virtual memory frames labelled 0-9. The FIFO algorithm uses a Linked List to accurately simulate the algorithm. It proved to be the simplest data structure for adding pages to the top of the queue and removing from the bottom. An Array List was used for the OPT and LRU algorithms, as this was the simplest data structure for filling in null physical frames, then swapping out specified frames within the list for frames in the reference string. OPT and LRU use very similar algorithms, as LRU essentially does the same thing as OPT except looks backwards instead of forwards.

The program itself is based off a loop that allows the user to switch between 7 options: 0 Exits out of the program safely.

1 allows the user to create a reference string. First it asks for the length of the reference string. Then it loops through this length until all the user’s pages are entered. Selecting 1 again overwrites any old reference string.  
2 generates a random reference string. Like 1, it asks for a number of pages from the user. 2 loops through this number, generating a new random page for each iteration, and storing it in the reference string. Like 1, using 2 more than once erases the old reference string. Whatever pages exist in the reference string,

3 will display it for the user in a readable format.  
4-7 are the FIFO, OPT, LRU and LFU algorithms, respectively. They commit nothing to memory, so they can be used over and interchangeably on the same reference string without having to reenter the reference string.

**Tests:**

**Test 1**

Using the references string from Homework 6:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 7 | 5 | 4 | 0 | 1 | 4 | 7 | 3 | 6 | 5 | 0 | 4 | 7 | 3 | 2 | 1 |

**Test Read Input:**

**Graphical user interface, text, application, Word

Description automatically generated**

**Test FIFO #1:**

**Graphical user interface, text, application, table, Word

Description automatically generated**

**Test FIFO #2:**

**Graphical user interface, text, application, table

Description automatically generated**

**Test FIFO #3:**

**Graphical user interface, text, application

Description automatically generated**

**Test OPT #1:**

**Graphical user interface, text, application, Word

Description automatically generated**

**Test OPT #2**

**Graphical user interface, text, application, table

Description automatically generated**

**Test OPT #3**

**Graphical user interface, text, application

Description automatically generated**

**Test LRU #1**

Graphical user interface, text, application, table, Word

Description automatically generated

**Test LRU #2**

**Graphical user interface, application, table, Word

Description automatically generated**

**Test LRU #3**

**Graphical user interface, text, application, table

Description automatically generated**

**Test LFU #1**

**Graphical user interface, text, application, Word

Description automatically generated**

**Test LFU #2**

**Graphical user interface, text, application, Word

Description automatically generated**

**Test LFU #3**

**Graphical user interface, text, application

Description automatically generated**

**Test 2**

Using the reference string:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 9 | 1 | 8 | 2 | 7 | 3 | 6 | 4 | 5 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| N | = | 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Test Read Input:**

**Graphical user interface, text, application, email

Description automatically generated**

**Test FIFO #1**

**Graphical user interface, text, application, Word

Description automatically generated**

**Test FIFO #2**

**Graphical user interface, application, table

Description automatically generated**

**Test FIFO #3**

**Graphical user interface, text, application

Description automatically generated**

**Test OPT #1**

**Graphical user interface, text, application, Word

Description automatically generated**

**Test OPT #2**

**Graphical user interface, application, table

Description automatically generated**

**Test OPT #3**

**Graphical user interface, text, application

Description automatically generated**

**Test LRU #1**

**Graphical user interface, text, application, Word

Description automatically generated**

**Test LRU #2**

**Graphical user interface, application, table

Description automatically generated**

**Test LRU #3**

**Graphical user interface, text, application

Description automatically generated**

**Test LFU #1**

**Graphical user interface, text, application, Word

Description automatically generated**

**Test LRU #2**

**Graphical user interface, application, table

Description automatically generated**

**Test LRU #3**

**Graphical user interface, text, application

Description automatically generated**

**Compile Successes**

**Graphical user interface, text, application

Description automatically generated**

Summery:

Throughout this Final Project I was able to make my code work correctly, I was able to learn much more about Design and implement a Demand Paging virtual memory simulator. I was able to learn a lot throughout this project and learn much more about Operating Systems. I am thankful that I was able to take this class and learn much more about this class and I am looking forward on learning more and increasing my skills.