Report Document – DemandPagingSimulator

Overview

This program is a demand paging virtual memory simulator implemented with Java. It simulates four different page replacement schemes:

* **FIFO** – next page to replace is the first one in among the current pages
* **OPT** – next page to replace is the one that will not be used for the longest amount of time
* **LRU** – next page to replace has been last used least recently among current pages
* **LFU** – next page to replace has been used the least in the past among current pages

The user, within a range of one to seven, can dictate the amount of physical frames in this simulator. The simulated single process that is running will always have a virtual memory of ten frames which means that the reference string values will range from zero to nine.

After the program starts, a menu prompts the user with eight different options. This menu will continue to prompt the user after executing the option chosen by the user until the user selects the *exit* option. In addition to *exit* and an option for each of the four paging algorithms, the other three options on the menu prompt are:

* *Read reference string* – reads user input for the reference string from the keyboard
* *Generate reference string* – randomly generates a reference string
  + Reference string length is given by the user
* *Display reference string* – simply outputs the current reference string

The reference string can only contain integers and white space. A reference string value can only be from zero to nine. Therefore, the reference string “12 3” is effectively the same as “1 2 3” as “12” in the first reference string will be evaluated as two separate numbers, not as twelve (since this is out of range of the virtual memory frames.) If a reference string input given by the user contains any characters other than white space, which is ignored, or integers then the user is notified and prompted to provide a new reference string. This is done as opposed to ignoring invalid characters in case the user made a mistake by entering a non-valid character.

There is no GUI for this program. All output and input are done via the console or command line.

**main – Starting the DemandPagingSimulator**

An integer, from one to seven, is expected as a command line argument by the program.

setFrameCount(int count)

This method sets the physical frame count. After the command line argument(s) are evaluated, if an argument has been provided, the argument is passed to this method as an integer. If no argument provided in command line, then this method is called with the maximum amount of physical frames allowed by default for convenience (especially if testing/using with an IDE.) This method calls validFrameCount(int count) to see if the count given is in the valid range. If not, then the default amount is used.

validFrameCount(int count)

This simply returns a Boolean on whether count is within the valid physical frame count range (0 – 7).

At this point, the frame count has been set and is displayed as confirmation to the user. Next, the referenceString and frames class variables are initialized as empty ArrayList and LinkedList, respectively. Finally, promptUser() is called.

**promptUser() – DemandPagingSimulator in Action**

promptUser()

This method displays the menu of options, captures the user’s selection and calls the processSelection method with this selection. This is all in a try/catch block to catch any invalid input given by the user. The try/catch also has a finally block which recursively calls this method to keep it in a loop until the user explicitly exits the program by choosing the exit option.

processSelection(int selection)