

CMSC412FinalProject

This project is the Final Project for CMSC 412.

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.

Driver Class

FinalProject.java

Algorithm Classes

Fifo.java

OPT.java

LRU.java

LFU.java

Utility Classes

outputTable.java

printClass.java

HelperClass.java

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.