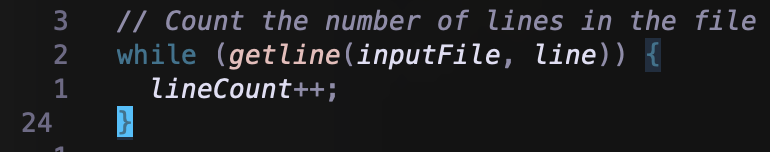
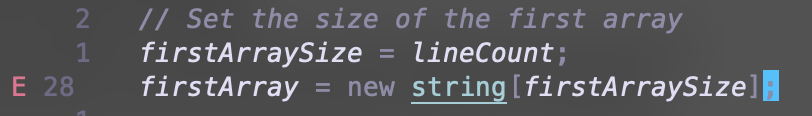
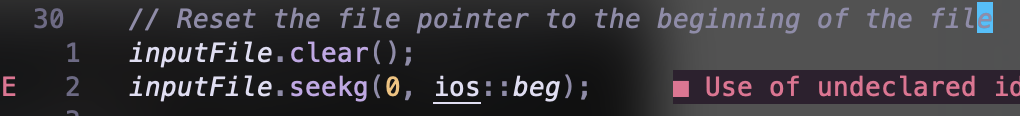
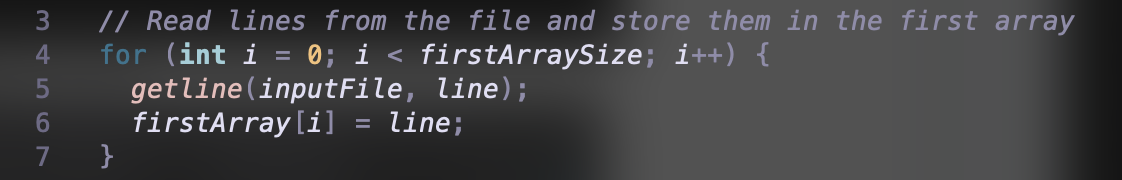
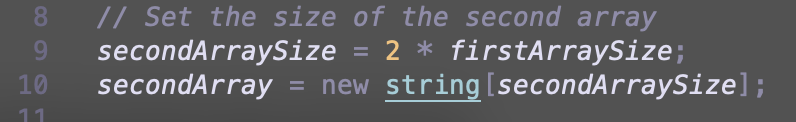
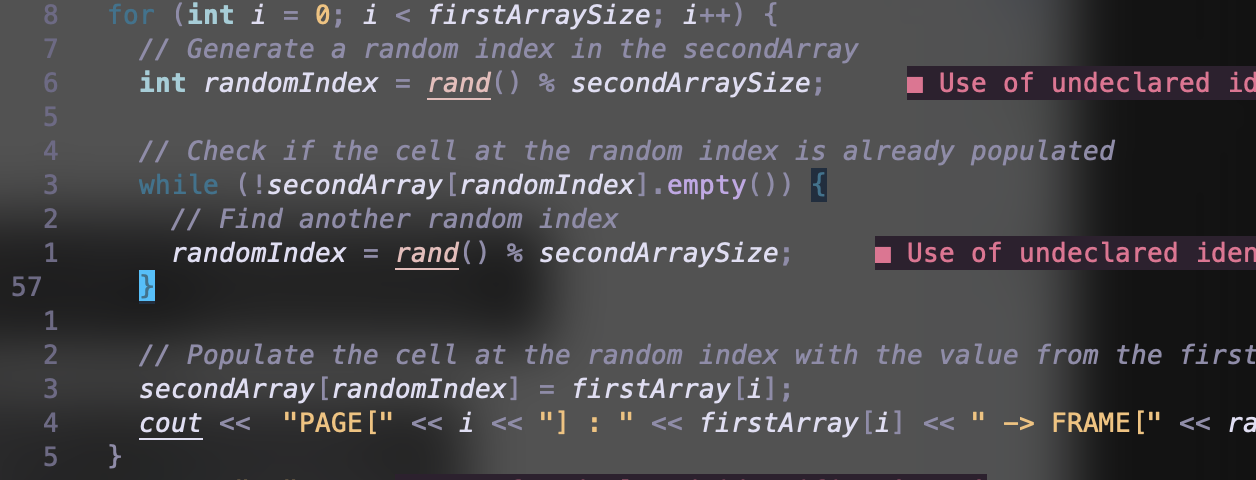
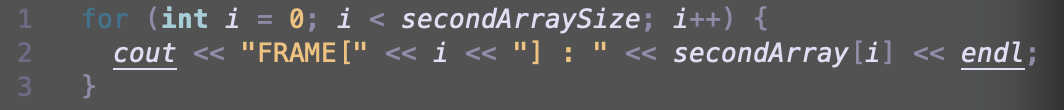
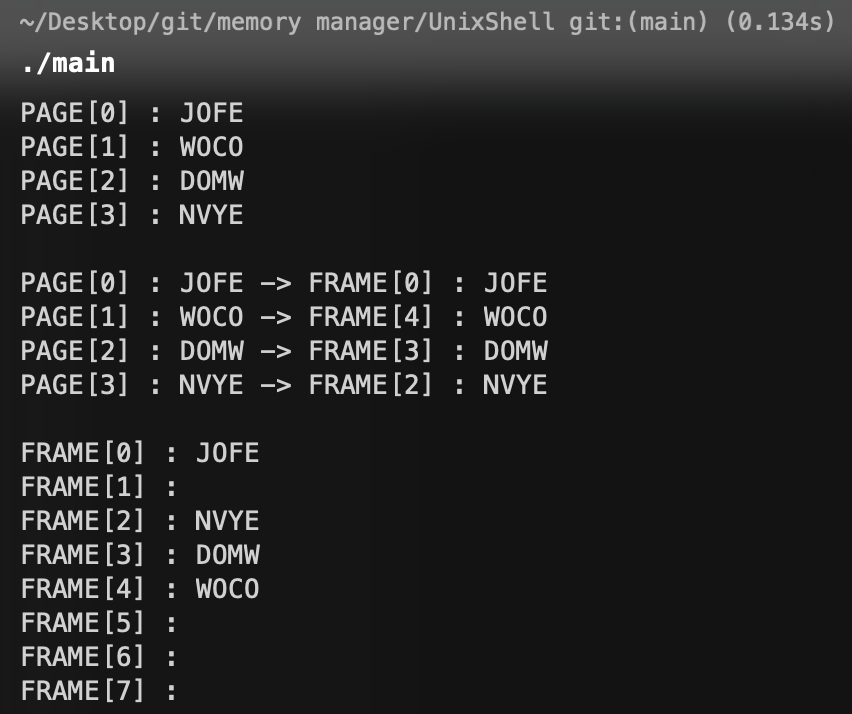
Project 4: Pager - A Virtual Memory Manager

CST-315 Operating Systems

Angel Velazquez, Nathan Dilla

March 10, 2024

In this project, we simulated a basic concept in operating systems called page replacement in virtual memory. Here is how the code operates:

* Header includes:
  + #include <iostream>: Standard input/output stream functionality.
  + #include <fstream>: File stream functionality.
  + #include <string>: String manipulation functions.
  + #include <cstdlib>: Standard C library functions for general purposes, used for rand() and srand().
  + #include <ctime>: C library for handling time, used for seeding the random number generator with the current time.
* Namespace:
  + using namespace std; The code uses the standard namespace, allowing the use of standard C++ functions without prepending them with std::.
* Main function:
  + ifstream inputFile("virtualMemory.txt"); Opens a file named "virtualMemory.txt" for input. This file presumably contains some data related to virtual memory.
  + string\* firstArray; and string\* secondArray; Declare two pointers to strings, which will be used as dynamic arrays to store data.
  + string line; Declares a string variable to hold each line read from the file.
  + int firstArraySize = 0; and int secondArraySize = 0; Initialize variables to store the sizes of the first and second arrays.
  + int lineCount = 0; Initialize a variable to count the number of lines in the file.
  + srand(time(0)); Seeds the random number generator with the current time.
  + Counting lines in the file:
  + Setting the size of the first array:
  + Resetting the file pointer to the beginning of the file:
  + Reading lines from the file and storing them in the first array:
  + Setting the size of the second array:
  + Printing the contents of the first array:
  + Iterating through the first array and simulating the page replacement:
  + Printing the contents of the second array, so frames after page replacement:
  + Output of program:
  + 

In summary, the program models a scenario where pages from virtual memory are loaded into physical memory frames, simulating the process of page replacement by randomly replacing pages in frames. The output helps visualize how pages are assigned to frames in a simplified virtual memory system.

GITHUB LINK: [HERE](https://github.com/nathdilla/UnixShell)