Department of Software Engineering

Lakehead University

ESOF3655: Operating Systems

Project 4:

Team members

Name: Haider ibrahim

Name: Adithya Krishna Neelangode Venkataraman

Name: Immanuel Akrofie-Bruce

Name: Jean Maiken Apuada

Name: Oluwadamilola Elias Tuwase

**Table of contents**

[**1.0 Introduction**](#_kuas6kyrqqyv) **3**

[1.1 System Overview](#_6bcwyqvdnl9c) 3

[1.2 Design system and implementation](#_uwxqg2er3r8c) 5

[**2.0 User Manual**](#_3ekw608lu990) **8**

[2.1 Create a file](#_h31lq0n597ji) 8

[2.2 Rename a File](#_cclg5qbkf3ab) 9

[2.3 Delete a File](#_gceydg72sko0) 10

[2.4 Copy a File](#_v0oflkmjmt07) 11

[2.5 Move a File](#_wfckv5z05nf8) 12

[2.6 Append to end of the File](#_667ezt6r4kf) 14

[2.7 Insert text in specified position](#_3u36ucthtbth) 15

[2.8 Remove text in the file](#_hic5h3k5vohe) 16

[2.9 Help Function](#_rrefi3g9hzri) 17

# **1.0 Introduction**

## **1.1 System Overview**

This a Portable File Manipulation (pofm) project. File manipulation is the capability of the operating system to grant permission to programs for operating on files, as well as providing an interface to the user to be able to create, delete, rename, copy, move a file from one directory to another, to be able to append text to the end of a file, to be able to insert text in a specific position, remove all text in a file and to be able to show the content of the file.

File manipulation has a lot of advantages but it does have disadvantages as well. The advantages are as follows:

**Advantages:**

* Portability :-

Since the program does not use any sort of system calls, it is not exclusively usable to just the Windows OS. The use of C++ language allows for usage in other platforms as well.

* Accessibility:-

Opening of file for reading and writing via the internal stream buffer which supports input and output operations.

**Disadvantages:**

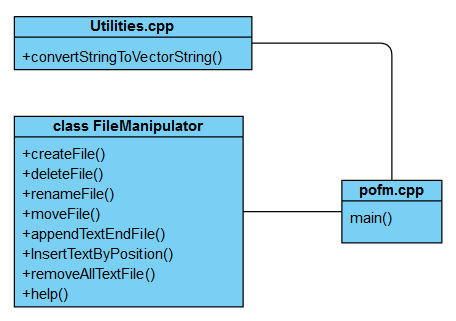
* Data Redundancy :-

Where data is inserted more than once, and there is no method of validating whether there are duplicates in the system.

* Data Mapping and Access:-

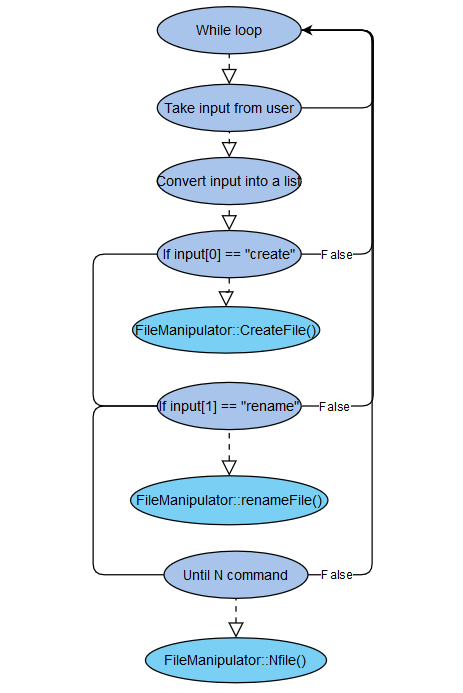
All related information may be grouped and stored in the system, no mapping between two files, that is two independant files are not linked. This is an inefficient and effective method to access data.

## **1.2 Design system and implementation**



System is designed in a very simple manner. It's made up of a FileManipulator class that contains all functions related to manipulating and changing files. Also, there is the Utilities cpp that contains functions to simplify various actions, but not related to file manipulation. In light of that, the pofm.cpp contains the actual usage of functions in utilities and FileManiupulator class to allow the user to manipulate the files. This implementation was chosen because it is simple, it allows for better organization of code in terms of keeping functions related to each other together or bundle together in the class. Moreover, it allows for code reusability. If we implemented everything in main() then, it would be hard to extract code that can be used in future programs. In addition, this implementation allowed for some sort of encapsulation, that is each member of the team didn’t have to worry about the other functions, just his assigned function.

pofm.cpp



In the main program, the input is taken as a string then converted into a list. This approach was chosen because it makes it easier for functions to work with lists that contain the command at index 0, and contain all the other arguments. After dividing the input into a list, the main takes the element at index 0 and compares it sequentially with every if statement until one of if statements is true, then it goes inside the loop to execute the corresponding function. This implementation was chosen, because it's simple.

# **2.0 User Manual**

## **2.1 Create a file**

**Name:**

createFile : create a new file

**Function Synopsis:**

createFile nameofFile

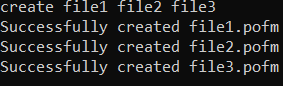
createFile nameofFile1 nameNameFile2….nameOfFileN

**Description:**

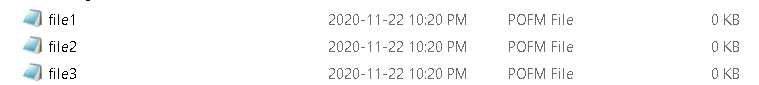
When this function is involved with any name, it will create a file in the current directory of the user. Also, it can be invoked with multiple files names and they will all be created in the same directory.

**Screenshot:**

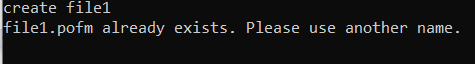
**Creating multiple files**

****

**The result**

****

**Error handling**



## **2.2 Rename a File**

**Name:**

renameFile: rename an existing file

**Function Synopsis:**

rename oldFileName

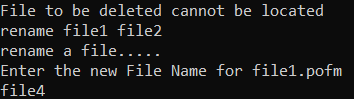
rename oldFileName1, oldFileName2, …,oldFileName(N)

**Function Description:**

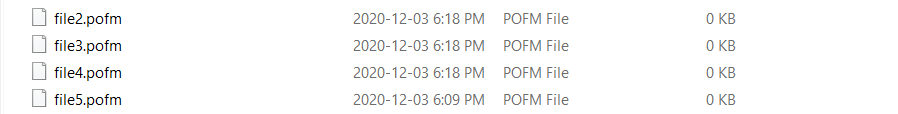
The function accepts the oldFileName and checks if the file exists. If the file exists, it requests for a newFileName. Hence, enabling it to rename multiple files.

**Screenshots**

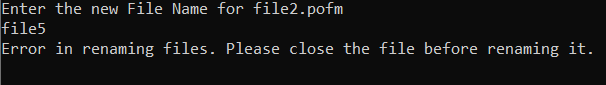
**Renaming multiple files**



**Result**



**Error Handling**



## **2.3 Delete a File**

**Name:**

deleteFile : Delete an existing file.

**Function Synopsis:**

delete fileName

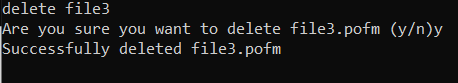
delete fileName1, fileName2, …, fileName(N)

**Function Description:**

The function accepts the fileName to be deleted from the user, checks if the file exists and requests the user for double confirmation before deleting the file.

**Screenshots**

Deleting a file



**Result**



**Error Handling**



## **2.4 Copy a File**

**Name:**

copyFile : Copy a file from an existing file.

**Function Synopsis:**

copy sourceNameandDirectory, locationNameandDirectory

**Function Description:**

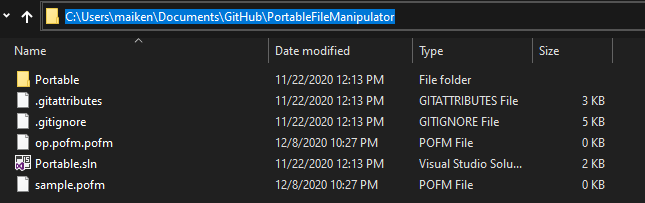
The function accepts two arguments: the fileName to be copied as well as the location where to put the copied file. Function checks if the file exists and gives appropriate warning if file does not exist.

**Screenshots**

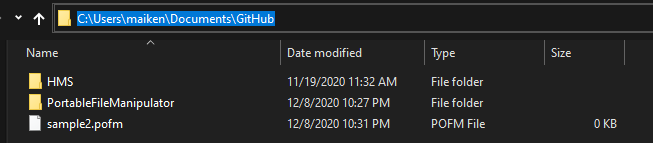




(1st string is the command, 2nd string is the source file, 3rd string is the target location)

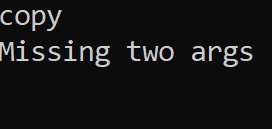


(source file: sample.pofm)

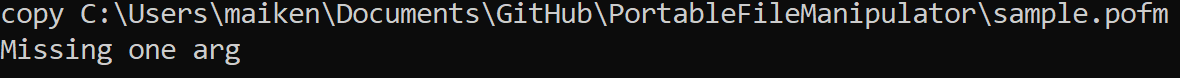


(copied file: sample2.pofm)

**Error Handling**



(missing source and target directory)



(missing target directory)

## **2.5 Move a File**

**Name:**

moveFile: Move a file from one directory to another.

**Function Synopsis:**

move sourceNameandDirectory, locationNameandDirectory

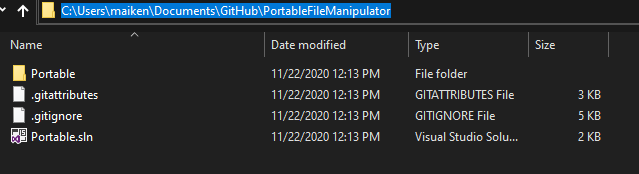
**Function Description:**

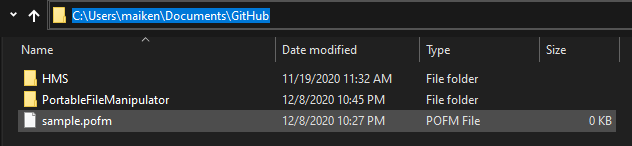
The function accepts two arguments: the fileName to be moved as well as the location where to put the file. Function checks if the file exists and gives appropriate warning if file does not exist.

**Screenshots:**



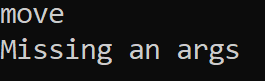




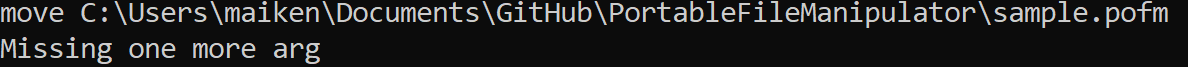


(moved file: sample.pofm)

**Error Handling:**



(missing source directory and target directory)



(missing target directory)

## **2.6 Append to end of the File**

**Name:**

append : It takes text from the user and appends to the end of the file.

**Function Synopsis:**

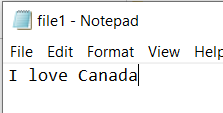
append fileName ‘text’

**Function Description:**

The function accepts the name of file as input and as well the text to be appended at the end of file. The text to be appended must between two single quotations marks otherwise the software will through an error message.

**Screenshots:**

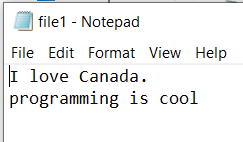
(Before)The inside of file1



Append to the file



(After)The inside of file1



**Error Handling:**



## **2.7 Insert text in specified position**

**Name:**

Insert : It takes text from the user and inserts based on the specified position.

**Function Synopsis:**

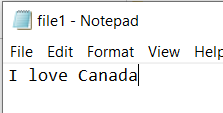
insert fileName postion ‘text’

**Function Description:**

The function accepts the name of file as input and specified position as well the text to insert.

**Screenshots:**

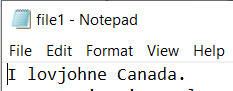
(Before)The inside of file1



Insert to the file



(After)The inside of file1

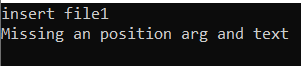


**Error Handling:**

Missing three arguments: filename, position, text



Missing two arguments: position and text



Missing the position



## **2.8 Remove text in the file**

**Name:**

removedata : It removes all the data in the file.

**Function Synopsis:**

removedata fileName

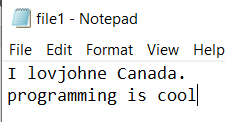
removedata fileName1, fileName2, …, fileName(N)

**Function Description:**

The function accepts either a single file or multiple files and it removes all the data inside the file.

**Screenshots:**

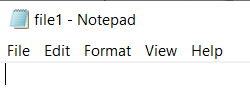
(Before)The inside of file1



Removedata from file



(After)The inside of file1



**Error Handling:**



## **2.9 Help Function**

**Name:**

removedata : It list out all the available functions

**Function Synopsis:**

Help

Help create

Help rename

Help delete

Help rename

Help removedata

Help append

Help insert

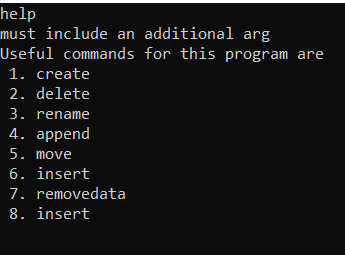
Help copy

**Function Description:**

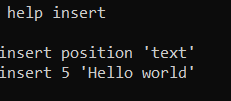
The function lists out all the available commands, gives a brief summary of what is functionality of the command and how to use it with some examples.

**Screenshots:**

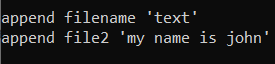
Help listing out all the available commands

****

Help with insert

****

Help with insert

****

**Error Handling:**

There is no error handling.