Operating Systems
POFM

Tyler McEwen, Dhaval Thanki, Abdulbasit Kasim, Joseph Olapade

29th November 2019

Introduction

This report consists of an in-depth review of our implementation of a Portable File Manipulator (POFM) that was implemented in the C programming language. Descriptions will include how the design was conceived, implementations and walkthroughs of the functions used within the Portable File Manipulator as well as limitations within its functionality. The report will also outline how we attempt to avoid allowing errors and limitations from creating a displeasing user experience with a friendly and informative array of "Helper" functions.

Upon reading this report, an individual may be able to understand the full functionality of our implementation of the aforementioned Portable file Manipulator. The individual will also be provided with a User Manual to give direct assistance to operate through their respective file system with ease when attempting to use the Portable file Manipulator.

Table of Contents

1. Design
    a. Conception
    b. Implementation
    c. Refinement
    d. Additional Information
2. User Manual
    a. Main Menu
        i. Create File
        ii. Delete File
        iii. Rename (created) File
        iv. Copy File
        v. Move File
        vi. Various Text Functionality
            1. Append Text
            2. Insert Text
            3. Remove Text
            4. Show File contents
3. Appendix
    a. Main Menu
        i. Create File
        ii. Delete File
        iii. Rename (created) File
        iv. Copy File
        v. Move File
        vi. Various Text Functionality
            1. Append Text
            2. Insert Text
            3. Remove Text
            4. Show File contents

1. Design

a. Conception

Upon initial analysis, we decided to use the CLI, the command-line interface, in order to maintain a low risk of irregular program faults, in order to implement a GUI, the graphical user interface, we need to handle numerous possible errors and given the nature and simplicity of this project we decided to use a minimalistic method and went ahead with a CLI approach.

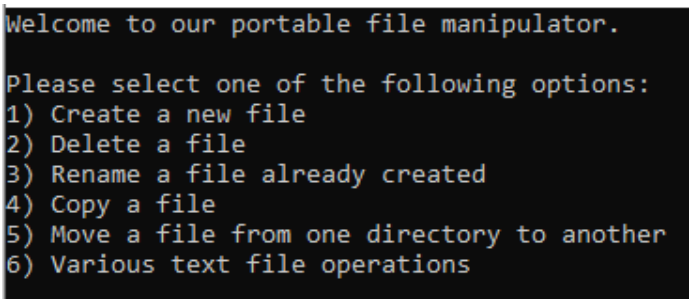
b. Implementation

A simple switch-case statement made it possible to create a Menu-like view of the commands we decided would be made accessible to the user. This menu allows the user to select from the many file operations by inputting the number associated with the operation (ie. 1 for the creation of a file).

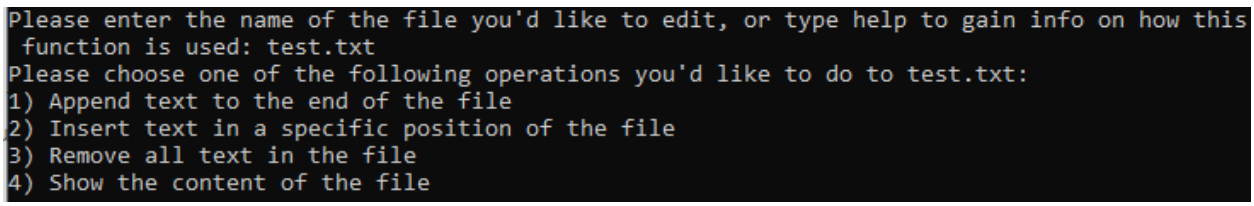
In addition, we chose to create an entirely new menu if the user wished to execute various text file operations since by doing so eliminates the need to present the user with a complicated menu. Therefore, if the user selects 6 they are first asked to enter the name of the file they would like to edit, then they are presented with the following new menu detailing only the operations that can be done on text files.

c. Refinement

Thus, upon the user executing the program they will first be presented with a menu that looks as follows:



If the user selects **"6)"** then they will be presented with the menu shown below;



d. Additional Information

In the example above, the user wishes to execute one or more of the various text file operations on the file *test.txt*. They can now do so simply by entering the number corresponding to the desired operation, as in the previous menu.

You may have noticed that the user is given the option to type the keyword **"help"** if they would like to gain more information on how the text operations menu works. Typing **"help"** not only gives a thorough description of how the function works, but it also gives the user the option to return back to the main menu if, after reading the description, they do not wish to execute any of the available options on their text file.

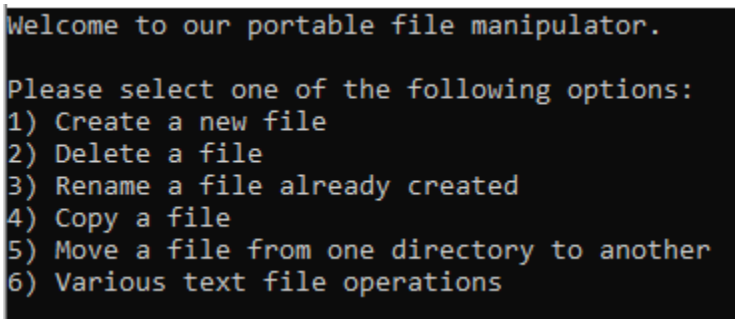
As shown in the second screenshot menu shown on the previous page, the user is given the option to type **"help"** if they would like to learn more about the specified function. If the user decides to do this, they will be presented with a thorough description of the function as well as the option to return to the main menu. All of our functions have this feature and our reasoning behind including the option to return to the main menu after learning more about the function is because if after learning exactly what the function does they may no longer desire to execute the function on their file.

By this method of implementation, the user does not need to exit the program and start from the beginning which we believe to be a great feature when taking time management into consideration.

2. User Manual

The following document will outline an in-depth explanation of the functions available for the user within the portable file manipulator. The user may gain insight on how to operate the program, use it to its full functionality and educate themselves on the inner workings of our portable file manipulator. To make things simple the demonstration will begin from an initial program launch, at the Main menu switch-case statement and go into an in-depth analysis of each function while we execute them on various files.

A. Main Menu



Upon execution of the program, you will be presented with this menu. This menu serves as the home base of the entire operation, it is here that you will both execute various file operations and be returned to if you decide, after reading the help functions, that you don't wish to continue with the function.

I. Create File

Let's start by creating a file. Follow the steps below to create a file:

1. Press **1** at the main menu
2. Press **0** if you'd like to learn more about the function or press **1** to continue. Here we pressed **0** to learn more
3. If 0 was pressed, you can now press **0** to return to the main menu or press 1 to continue with the creation of a file. Here we pressed 1
4. Enter the name, path, and extension of the file you'd like to create. Here we are creating test.txt in our current directory
5. You will now be notified if the file has been created successfully
6. You can now press **"y"** to return to the main menu or **"n"** to exit the program. We chose to
7. Exit

```
Please select one of the following options:
1) Create a new file
2) Delete a file
3) Rename a file already created
4) Copy a file
5) Move a files contents from one location to another
6) Various text file operations
1
You have chosen to create a file specified by you, the user. If you'd like to learn more about this function please press 0
, else press 1 to continue: 0
This function begins by asking you for the name and extension of the file you'd like to create and then creates it. If this
 is not the function you meant to select, kindly press 0 to return to the main menu, else press 1 to continue: 1
Please the name and extension (.txt, .bat, etc) you'd like to create: test.txt
File 'test.txt' successfully created.
Would you like to execute more operations? (y/n): n
```

*code for the createFile() function can be found in **Appendix 3. a) i.**

II. Delete File

Now let's delete the file renamed to testToDelete.txt in the last step. The steps are shown below:

1. Select 2 from the main menu
2. Press 0 to learn more about the function or 1 to continue. Here we chose to learn more
3. If you chose to learn more, press 0 to return to the main menu or 1 to continue operations. Here we chose to continue.
4. Enter the name and path of the file you'd like to delete. Here we chose to delete testToDelete.txt located on the Desktop.
5. If successful, you will be notified. If there are any problems then you will also be notified.
6. Press "y" to return to the main menu or "n" to exit the program. We chose to exit.

```
Please select one of the following options:
1) Create a new file
2) Delete a file
3) Rename a file already created
4) Copy a file
5) Move a files contents from one location to another
6) Various text file operations
2
You have chosen to delete a file that will soon be specified by you, the user. If you'd like to learn more about
this function please press 0, else press 1 to continue: 0
This function begins by asking you to specify the file you'd like to delete. After recieving your input, it will
proceed to delete it. If this is not the function you meant to select, kindly press 0 to return to the main menu,
 else press 1 to continue: 1

Please enter the name and extension of the file you'd like to delete: C:\Users\Tyler\Desktop\testToDelete.txt
The file specified 'C:\Users\Tyler\Desktop\testToDelete.' was succesfully deleted.
Would you like to execute more operations? (y/n): n

Process returned 0 (0x0)   execution time : 21.812 s
Press any key to continue.
```

*code for deleteFile() can be found in Appendix 3. a) ii.

III. Rename File

Now that we have two versions of the same file in different places on our PC let's rename one of them. Here we are going to rename the file located on the Desktop from testCopy.txt to testToDelete.txt as we are going to delete it in the next section. Follow the steps below to do the same:

1. Press 3 at the main menu
2. Press 0 to learn more about the function or 1 to continue with operations. Here we chose to learn more.
3. If you chose to learn more, press 0 to return to the main menu or 1 to continue with operations. Here we chose to continue
4. Enter the path and name of the file you'd like to rename. Here we are renaming the testCopy.txt file located on the Desktop
5. Enter the path and name of the file you'd like to rename the file entered in the previous step too. Here we are keeping the same path but renaming the file to testToDelete.txt
6. If the file is successfully renamed or if there are any problems, the function will let you know. Here the function notified us of the successful completion
7. Press "y" to return to the main menu or "n" to exit the program. Here we chose to exit the program

```
Please select one of the following options:
1) Create a new file
2) Delete a file
3) Rename a file already created
4) Copy a file
5) Move a files contents from one location to another
6) Various text file operations
3
You have chosen to rename a preexisting file. If you would like to learn more about this function please press 0, else press
 1 to continue: 0
This function will first ask you to enter the name of an already existent file. If the file exists it will ask you to input
what you'd like ot rename it to. If it doesn't exist however, it will prompt you to enter the name of a file that does.
If you don't wish to rename a file please press 0 to return to the main menu, else press 1 to continue: 1
Please enter name of the file that will be renamed: C:\Users\Tyler\Desktop\testCopy.txt
Please enter what you'd like to rename the file to: C:\Users\Tyler\Desktop\testToDelete.txt
File C:\Users\Tyler\Desktop\testCopy.txt has successfully been renamed to C:\Users\Tyler\Desktop\testToDelete.txt
Would you like to execute more operations? (y/n): n

Process returned 0 (0x0)   execution time : 34.691 s
Press any key to continue.
```

*code for renameFile() can be found in Appendix 3. a) iii.

IV. Copy File

Let us now create a copy of "test.txt" containing the text we just added in the last section. To copy a file follow the steps below:

1. Begin by pressing 4 at the main menu
2. Press 0 to learn more about the function or 1 to continue with operations. We chose 0 in the screenshot below.
3. If 0 was chosen, you now have the option to press 0 to return to the main menu or 1 to continue with operations. We chose 1.
4. Now enter the name of the file you'd like to copy. Here we want to copy test.txt.
5. Now enter the name of the file you'd like to copy the data to.
6. If the file exists, it will overwrite the content of the said file with the contents of the file input in step 4. On the other hand, if the file doesn't exist, as is the case in the screenshot below, press 1 to create it or 0 to return to the main menu.
7. If you opted to create the file as we did, the function will notify you as to whether the data has been successfully copied.
8. Press "y" to return to the main menu or "n" to exit the program.

```
1) Create a new file
2) Delete a file
3) Rename a file already created
4) Copy a file
5) Move a files contents from one location to another
6) Various text file operations
4
You have chosen to copy the contents of an existing file into an either already existent or an en
tirely new file. If you'd like to learn more about this function please press 0, else press 1 to
continue: 0
This file first will prompt you to input the name of the file you'd like to copy the contents of,
 then, upon checking for its existence, it will ask you to enter the name of the new file that yo
u'd like copy the contents into. If the file doesn't already exist, the program will ask if you'd
 like to create it. It will then copy the contents of the old file into the new one and notify yo
u upon completion. If this is not the function you'd like to execute, please press 0 to return to
 the main menu, else press 1 to continue: 1
Please enter the name of the file you'd like to copy: test.txt
Enter the name of the file you'd like to copy the data to: testCopy.txt
File does not exist. Press 1 if you'd like to create it or 0 to return to the main menu: 1
File successfully copied.
Would you like to execute more operations? (y/n): n

Process returned 0 (0x0)   execution time : 21.365 s
Press any key to continue.
```

*Code for copyFile() can be found in Appendix 3. a) iv.

V. Move File

Now that we have two copies of the same file let's move one to a different directory. In this example, it will be our desktop. Steps to do this can be found below:

1. Select 5 at the main menu
2. Press 0 to learn more about the function or press 1 to continue with operations. Here we chose to learn more.
3. Press 0 to return to the main menu or press 1 to continue. Here we chose to continue.
4. Enter the name of the file you'd like to move to. Here we are moving testCopy.txt.
5. Enter the path and name of the location you would like to move the file to. Here we are moving it to the Desktop and keeping the name the same.
6. If the location you chose to move it to is accessible and able to be read to, you will receive a notification saying the file has been successfully moved, as is the case below.
7. Press "y" to go back to the main menu or "n" to exit the program. Here we chose to exit.

```
Please select one of the following options:
1) Create a new file
2) Delete a file
3) Rename a file already created
4) Copy a file
5) Move a files contents from one location to another
6) Various text file operations
5
You have chosen to move a file to a new location. If you'd like to learn more about this function please press 0, else press
1 to continue: 0
This function begins by asking you the name and extension of the file you'd like to move. After recieving your input, it will
 then ask you to input the name of the file where you'd like to move the data. It will then proceed to do so. If this is not
the function you meant to select, kindly press 0 to return to the main menu, else press 1 to continue: 1
Please enter the name of the file you'd like to move: testCopy.txt
Enter the location where you'd like to move the file: C:\Users\Tyler\Desktop\testCopy.txt
File successfully moved.
Would you like to execute more operations? (y/n): n

Process returned 0 (0x0)   execution time : 52.046 s
Press any key to continue.
```

*code for moveFile() can be found in Appendix 3. a) v.

VI. Various Text Functions

Outlined below are the additional functions used by the user to populate created files with text data.

1. Append Text

Let us now add some text to our newly created file. The following steps are required to append text to a file:

1. Select 6 at the main menu to get to the text operations menu
2. Enter the name of the file you'd like to edit (append text to) at the prompt. Here we are using test.txt
3. Press 0 to learn more about the append text function, or press 1 to move on. Here we pressed 0 to learn more
4. If 0 was pressed, read about the function and press 0 if you'd like to return to the main menu or press 1 to continue
5. Here you will be notified if the file was opened successfully or not
6. Enter the text you'd like to append to the file. Here we chose to append "Hello, and welcome to our portable file manipulator."
7. You are now able to choose whether you'd like to return to the main menu or exit the program, by pressing "y" or "n" respectively.

```
Please select one of the following options:
1) Create a new file
2) Delete a file
3) Rename a file already created
4) Copy a file
5) Move a files contents from one location to another
6) Various text file operations
6
Please enter the name of the file you'd like to edit, or type help to gain info on how this function is used: test.txt
Please choose one of the following operations you'd like to do to test.txt:
1) Append text to the end of the file
2) Insert text in a specific position of the file
3) Remove all text in the file
4) Show the content of the file
1
You have chosen to append text to the end the specified text file. To learn more about this function please press 0, else
 press 1 to continue: 0
This function asks you, the user, to enter the text you'd like to append to the end of the file and then does it.If this
is not the function you meant to select, kindly press 0 to return to the main menu, else press 1 to continue: 1
File successfully opened. Please enter the text you'd like to append to the file:
Hello, and welcome to our portable file manipulator.
Would you like to execute more operations? (y/n): n

Process returned 0 (0x0)   execution time : 34.723 s
Press any key to continue.
```

*Code for the appendText() function can be found in Appendix 3. a) vi. 1.

2. Insert Text

Let's now insert some text in our original test.txt file at a position specified by us. To do so follow the steps below:

1. Press 6 at the main menu to gain access to the text file operations menu
2. Enter the name of the file you'd like to edit if the text file exists you will be presented with the menu. Here we entered "test.txt".
3. Now we are in the text operations menu. Press 2 to insert text at a specified position.
4. Press 0 to learn more about the function or press 1 to continue with operations. We chose to learn more.
5. If you chose to learn more too, you can either press 0 to return to the main menu or 1 to continue. We chose to continue.
6. Specify the character index that you would like to start writing. We chose 4.
7. Enter the text you would like to insert at that position. We chose to insert [your name here].
8. The function will now present you with what the text file contains after writing your text at the specified position.
9. Press "y" to return to the main menu or "n" to exit. We chose to exit.

```
Please select one of the following options:
1) Create a new file
2) Delete a file
3) Rename a file already created
4) Copy a file
5) Move a files contents from one location to another
6) Various text file operations
6
Please enter the name of the file you'd like to edit, or type help to gain info on how this function is used: test.txt
Please choose one of the following operations you'd like to do to test.txt:
1) Append text to the end of the file
2) Insert text in a specific position of the file
3) Remove all text in the file
4) Show the content of the file
2
You have chosen to insert text enter by you, the user, into a specified location of the chosen file. If this you'd like
to learn more about this function press 0, otherwise press 1 to continue: 0
This function begins by asking you the location in the file you'd like to begin writing. After that it will ask you what
 text you'd like to include at that position.If this is not the function you meant to select, kindly press 0 to return t
o the main menu, else press 1 to continue: 1

Please specify the character index where you'd like to start writing: 4
Please enter the string you'd like to insert at position 4:  [your name here]
Contents of file after insertion: Hello [your name here] , and welcome to our pofm.
Would you like to execute more operations? (y/n): n

Process returned 0 (0x0)   execution time : 43.449 s
Press any key to continue.
```

*code for insertText() can be found in Appendix 3. a) vi. 2.

3. Remove Text

Here we showcase our final function contained in our portable file manipulator. This function removes all text from a text file so we are going to part ways with the text contained within "test.txt". Follow the steps below to do the same:

1. Select 6 from the main menu to make your way over to the text operations menu
2. Enter the name of the file you'd like to make changes to. We entered "test.txt"
3. Select 3 from the text operations menu
4. Press 0 to learn more about the function or 1 to continue. We chose to learn more.
5. If you chose to learn more, press 0 to return to the main menu or 1 to continue. We chose to continue.
6. You will be notified if the text in the file was successfully deleted, as is the case below.
7. Press "y" to return to the main menu or "n" to exit. We chose to exit.

```
Please select one of the following options:
1) Create a new file
2) Delete a file
3) Rename a file already created
4) Copy a file
5) Move a files contents from one location to another
6) Various text file operations
6
Please enter the name of the file you'd like to edit, or type help to gain info on how this function is used: test.txt
Please choose one of the following operations you'd like to do to test.txt:
1) Append text to the end of the file
2) Insert text in a specific position of the file
3) Remove all text in the file
4) Show the content of the file
3
You have chosen to remove all text from the specified text file. If you'd like to learn more about this function please
press 0. If you'd like to continue then simply press 1.0
This function works by first opening the file for writing purposes and then simply prints empty quotations to the file t
hus overwriting the previous content with nothing. If you do not want to delete all the text in the specified file then
kindly press 0 to return to the main menu, else press 1 to continue: 1
Contents of file successfully deleted.
Would you like to execute more operations? (y/n): n

Process returned 0 (0x0)   execution time : 22.295 s
Press any key to continue.
```

*code for removeText() can be found in Appendix 3. a) vi. 3.

4. Show File contents

Now we are going to display the text contained within test.txt prior to removing all the text in the last section using our removeText() function. To display the text follow the steps below (NOTE: this function does have the feature to stop and ask if you'd like to continue reading after encountering a new line however the text in our file isn't long enough to trigger that feature. You can find the feature located in its code in Appendix A):

1. Press 6 at the main menu to make your way to the text file operations menu
2. Enter the name of the text file you'd like to modify. We entered "test.txt"
3. Select 4 from the text operations menu
4. Press 0 to learn more about the function or 1 to continue. We chose to learn more.
5. If you chose to learn more you can either press 0 to return to the main menu or 1 to continue. We chose to continue
6. If your file was able to be read from, the text contained within it will now be displayed.
7. Press "y" to return to the main menu or "n" to exit. We chose to exit.

```
Please select one of the following options:
1) Create a new file
2) Delete a file
3) Rename a file already created
4) Copy a file
5) Move a files contents from one location to another
6) Various text file operations
6
Please enter the name of the file you'd like to edit, or type help to gain info on how this function is used: test.txt

Please choose one of the following operations you'd like to do to test.txt:
1) Append text to the end of the file
2) Insert text in a specific position of the file
3) Remove all text in the file
4) Show the content of the file
4
You have chosen to display the text of the selected file. To learn more about this function press 0, or, if you'd like
 to continue, please press 1: 0
This function first opens the file for reading purposes then reads up until a new line is encountered. At this point y
ou have the option to either continue or stop reading from the file by typing y or n, respectively. If this is not the
 option you meant to select, press 0 to go back to the main menu or press 1 if you'd like to proceed.1
Hello [your name here] , and welcome to our pofm.
Would you like to execute more operations? (y/n): n

Process returned 0 (0x0)   execution time : 13.416 s
Press any key to continue.
```

*code for displayText() can be found in Appendix 3. a) vi. 4.

3. Appendix

The Appendix section contains raw code that supports the portable file manipulator, the code is provided to demonstrate functionality and portability.

```
612  //main function
613  int main()
614  {
615      printf("Welcome to our portable file manipulator.\n\n");
616      mainMenu(); //display main menu
617      return 0;
618  }
```

A. Main Menu

```
571  //main menu, provides the user with a menu with which to execute various
572  //file operations by entering the corresponding number
573  void mainMenu() {
574      int choice;
575
576      printf("Please select one of the following options:\n");
577      printf("1) Create a new file\n");
578      printf("2) Delete a file\n");
579      printf("3) Rename a file already created\n");
580      printf("4) Copy a file\n");
581      printf("5) Move a files contents from one location to another\n");
582      printf("6) Various text file operations\n");
583
584      scanf("%d", &choice); //read user input
585
586      while(choice > 6 || choice < 1) {
587          printf("Invalid input. Please enter a number between 1 and 6: ");
588          scanf("%d", &choice);
589      }
590      switch(choice) {
591          case 1:
592              createFile(); //create a file
593              break;
594          case 2:
595              deleteFile(); //delete a file
596              break;
597          case 3:
598              renameFile(); //rename a file
599              break;
600          case 4:
601              copyFile(); //copy a file
602              break;
603          case 5:
604              moveFile(); //move a file
605              break;
606          case 6:
607              textOps(); //brings user to a menu of various text operations
608              break;
609      }
610  }
```

I. Create File

```
19  //function used to create a file,
20  void createFile()   {
21      // Initialize necessary vars
22      int choice;
23      char createdFile[50];
24      FILE *newFile;
25      char c;
26
27      printf("You have chosen to create a file specified by you, the user. If you'd like to learn more ab
28      scanf("%d", &choice); //read user input
29      if(choice == 0) { //that is, the user would like to learn more about the functions operations
30          printf("This function begins by asking you for the name and extension of the file you'd like to
31          printf(" If this is not the function you meant to select, kindly press 0 to return to the main
32          scanf("%d", &choice); //read user input
33          while(choice > 1 || choice < 0) {
34          if(choice == 0) //that is, the user doesn not want to create a file and wants to return to the
35              mainMenu();
36          }
37      }
38      printf("Please the name and extension (.txt, .bat, etc) you'd like to create: ");
39      scanf("%s", createdFile); //read user input
40
41      newFile = fopen(createdFile, "ab+"); //create new file
42
43      if (newFile) { //that is, if the file has been successfully created
44          printf("File '%s' successfully created.\n", createdFile);
45      }
46      else { //file unable to be created
47          printf("Error: File '%s' was not created.", createdFile);
48          printf("\nAccess to this file may be restricted.\n");
49      }
50      fclose(newFile); //close file
51
52      getchar(); //eat up new line
53      printf("Would you like to execute more operations? (y/n): ");
54      scanf("%c", &c); //get user input
55      while(c != 'y' && c != 'n') { //while invalid input
56          printf("Invalid input. Please enter y or n.");
57          scanf("%c", &c);
58      }
59      if(c == 'y') //bring user back to main menu to execute more operations
60          mainMenu();
61      else
62          return;
63  }
```

II. Delete File

```
65  //function used to delete an already existing file
66  void deleteFile()   {
67      // Initialize necessary vars
68      FILE* fPtr;
69      int choice;
70      char targetFile[32];
71      int delSuccess;
72      char c;
73
74      printf("You have chosen to delete a file that will soon be specified by you, the user. If you'd lik
75      scanf("%d", &choice); //read user input
76      if(choice == 0) { //that is, user would like to learn more about the functions operations
77          printf("This function begins by asking you to specify the file you'd like to delete. After reci
78          printf(" If this is not the function you meant to select, kindly press 0 to return to the main
79          scanf("%d", &choice);//read user input
80          if(choice == 0) //direct user back to main menu
81              mainMenu();
82      }
83          printf("\nPlease enter the name and extension of the file you'd like to delete: ");
84              scanf("%s", targetFile); //read the name of the file the user would like to delete
85
86      if(!(fPtr = fopen(targetFile, "r"))) { //file does not exist or user does not have permission to op
87          printf("Error. %s does not exist.\n", targetFile);
88          return;
89          }
90      fclose(fPtr);
91
92      delSuccess = remove(targetFile); //attempt to remove file. remove() will return 0 if successful
93
94      if(delSuccess == 0) { //file successfully deleted
95          printf("The file specified '%s' was succesfully deleted.", targetFile);
96      }
97      else { //file has not been deleted
98          printf("Error. %s was not deleted. Access may be restricted\n", targetFile);
99      }
100
101     getchar(); //eat up new line
102     printf("\nWould you like to execute more operations? (y/n): ");
103     scanf("%c", &c); //get user input
104     while(c != 'y' && c != 'n') { //while invalid input
105         printf("Invalid input. Please enter y or n.");
106         scanf("%c", &c);
107     }
108     if(c == 'y') //bring user back to main menu to execute more operations
109         mainMenu();
110     else
111         return;
112 }
```

III. Rename (created) File

```
183  //function to rename an already existent file
184  void renameFile() {
185
186      FILE* fPtr;
187      int choice;
188      char oldFileName[50], newFileName[50], c;
189
190      printf("You have chosen to rename a preexisting file. If you would like to learn more about this fu
191      scanf("%d", &choice); //read user input
192      if(choice == 0) { //tell the user more about the functions operations
193          printf("This function will first ask you to enter the name of an already existent file. If the
194          printf("it will prompt you to enter the name of a file that does.\n");
195          printf("If you don't wish to rename a file please press 0 to return to the main menu, else pres
196          scanf("%d", &choice); //read user input
197          if(choice == 0) //redirect user to main menu
198              mainMenu();
199      }
200      printf("Please enter name of the file that will be renamed: ");
201      scanf("%s", oldFileName); //read name of file that the user would like renamed
202
203      fPtr = fopen(oldFileName, "r"); //attempt to open the file
204
205      while(!fPtr) //while file doesn't exist
206          {
207              printf("File does not exist. Please enter one that does: ");
208              scanf("%s", oldFileName); //read user input
209              fPtr = fopen(oldFileName, "r"); //attempt to open the file
210          }
211      fclose(fPtr); //close file
212
213      printf("Please enter what you'd like to rename the file to: ");
214      scanf("%s", newFileName); //read what the user would like the file to renamed to
215      fPtr = fopen(newFileName, "r"); //check to see if a file already exists under that name
216
217          while (fPtr) //while a file already exists under the given name
218          {
219              printf("File name already exists. Please enter something that doesn't: ");
220              scanf("%s", newFileName); //read user input
221              fPtr = fopen(newFileName, "r"); //check to see if a file already exists under the desired n
222          }
223      rename(oldFileName, newFileName); //rename the old filename to the new filename
224      printf("File %s has successfully been renamed to %s", oldFileName, newFileName); //notify user that
225
226      getchar(); //eat up new line
227      printf("Would you like to execute more operations? (y/n): ");
228      scanf("%c", &c); //get user input
229      while(c != 'y' && c != 'n') { //while invalid input
230          printf("Invalid input. Please enter y or n.");
231          scanf("%c", &c);
232      }
233      if(c == 'y') //bring user back to main menu to execute more operations
234          mainMenu();
235      else
236          return;
237
238  }
```

IV. Copy File

```
240  //function to copy the contents of a file into an already existent or entirely new file
241  void copyFile()
242  {
243      int choice;
244      FILE *oldFile, *newFile;
245      char ch, oldFileName[25], newFileName[25], c;
246
247      printf("You have chosen to copy the contents of an existing file into an either already existent or
248      scanf("%d", &choice); //read user input
249      if(choice == 0) { //tell the user more about the function
250          printf("This file first will prompt you to input the name of the file you'd like to copy the co
251          printf(" If this is not the function you'd like to execute, please press 0 to return to the mai
252          scanf("%d", &choice); //read user input
253          if(choice == 0) //redirect user to main menu
254              mainMenu();
255      }
256      printf("Please enter the name of the file you'd like to copy: ");
257      scanf("%s",oldFileName); //read the name of the file the user would like to copy the contents of
258
259      oldFile = fopen(oldFileName, "r"); //attempt to open it
260      while (!oldFile) { //while the file cannot be opened
261          printf("File does not exist. Please enter the name of a file that exists: ");
262          scanf("%s",oldFileName); //read user input
263          oldFile = fopen(oldFileName, "r"); //attempt to open file
264      }
265      if( oldFile == NULL ) { //if file cannot be read from
266          printf("Error. You do not have permission to read from this file.");
267          return;
268      }
269
270      printf("Enter the name of the file you'd like to copy the data to: ");
271      scanf("%s",newFileName); //get name of file the user would like to copy the data to
272
273      newFile = fopen(newFileName, "r"); //attempt to open the file
274
275      while(!newFile) { //while the file cannot be opened
276          printf("File does not exist. Press 1 if you'd like to create it or 0 to return to the main menu
277          scanf("%d", &choice);//get user input
278          if(choice == 0) //return user to main menu
279              mainMenu();
280
281          newFile = fopen(newFileName, "w"); //create file
282          }
283
284      fclose(newFile); //close file
285
286      newFile = fopen(newFileName, "wb"); //open file for writing purposes
287
288      if( newFile == NULL ) { //file cannot be written to
289        fclose(oldFile);
290        printf("Error. You do not have permission to write to %s", newFileName);
291        return;
292      }
293
294      while ((ch = fgetc(oldFile)) != EOF) { // copy source to new file char by char till the end of file
295          fputc(ch, newFile);
296      }
297
298      printf("File successfully copied.\n");
299
300      fclose(oldFile); //close file
301      fclose(newFile); //close file
302
303      getchar(); //eat up new line
304      printf("Would you like to execute more operations? (y/n): ");
305      scanf("%c", &c); //get user input
306      while(c != 'y' && c != 'n') { //while invalid input
307          printf("Invalid input. Please enter y or n.");
308          scanf("%c", &c);
309      }
310      if(c == 'y') //bring user back to main menu to execute more operations
311          mainMenu();
312      else
313          return;
314  }
```

V. Move File

```
114  //function to move a preexisting file to a new location
115  void moveFile() {
116
117      FILE *fPtr, *destPtr;
118      int choice;
119      char ch, fileName[30], dirName[100], c;
120
121      printf("You have chosen to move a file to a new location. If you'd like to learn more about this fu
122      scanf("%d", &choice); //read user input
123      if(choice == 0) { //user would like to learn more
124          printf("This function begins by asking you the name and extension of the file you'd like to mov
125          printf(" If this is not the function you meant to select, kindly press 0 to return to the main
126          scanf("%d", &choice); //read user input
127          if(choice == 0) //direct user back to main menu
128              mainMenu();
129      }
130      printf("Please enter the name of the file you'd like to move: ");
131      scanf("%s",fileName); //read user input containing the file they'd like to move
132
133      fPtr = fopen(fileName, "r"); //open file
134
135      while(fPtr == NULL) { //while the file either does not exist or cannot be opened
136          printf("Error. File either does not exist or you do not have persmission to open it. Please ent
137          scanf("%s",fileName); //get user input
138          if(strcmp(fileName, "quit") == 0) //exit program
139              return;
140          fPtr = fopen(fileName, "r"); //try to open file for reading purposes
141
142      }
143
144      printf("Enter the location where you'd like to move the file: ");
145      scanf("%s",dirName); //read user input detailing where they'd like to move the file
146
147      destPtr = fopen(dirName, "wb"); // Open file to write content of source
148
149      while(destPtr == NULL) { //the file either does not exist or cannot be opened
150          printf("Error accessing location. Check it's existence and your permissions and try again, or t
151          scanf("%s",dirName); //get user input
152          if(strcmp(dirName, "quit") == 0) { //exit program
153              fclose(fPtr);
154              return;
155          }
156          destPtr = fopen(dirName, "wb"); //attempt to open file
157      }
158
159      while (( ch = fgetc(fPtr) ) != EOF) { // Write content of source to new file char by char till the
160          fputc(ch, destPtr);
161      }
162
163      printf("File successfully moved.\n");
164
165      fclose(fPtr);
166      fclose(destPtr);
167
168      remove(fileName); // Remove source file
169
170      getchar(); //eat up new line
171      printf("Would you like to execute more operations? (y/n): ");
172      scanf("%c", &c); //get user input
173      while(c != 'y' && c != 'n') { //while invalid input
174          printf("Invalid input. Please enter y or n.");
175          scanf("%c", &c);
176      }
177      if(c == 'y') //bring user back to main menu to execute more operations
178          mainMenu();
179      else
180          return;
181  }
```

VI. Various Text Functionality

```
515  //function to display the menu detailing the operations that can be performed on the text file entered
516  void textOps() {
517
518      char fileName[25];
519      int choice;
520      int goBack = 0;
521      FILE* fPtr;
522
523      printf("Please enter the name of the file you'd like to edit, or type help to gain info on how this
524      scanf("%s", fileName); //read file user would like to modify
525
526      if(strcmp(fileName, "help") == 0) { //tell user more about function
527          printf("This function serves as a menu that allows the user(you) to choose between various oper
528          scanf("%d", &choice); //read user input
529          if(choice == goBack) //return user to main menu
530              mainMenu();
531          else { //user wants to continue
532              printf("Please enter the name of the file you'd like to edit: ");
533              scanf("%s", fileName); //read file name entered by user
534          }
535      }
536      fPtr = fopen(fileName, "r");
537      while(!fPtr) {
538          printf("File does not exist. Please enter the name of one that does: ");
539          scanf("%s", fileName);
540          fPtr = fopen(fileName, "r");
541      }
542      //display menu
543      printf("Please choose one of the following operations you'd like to do to %s: \n", fileName);
544      printf("1) Append text to the end of the file\n");
545      printf("2) Insert text in a specific position of the file\n");
546      printf("3) Remove all text in the file\n");
547      printf("4) Show the content of the file\n");
548
549      scanf("%d", &choice); //read user input
550
551      while(choice > 4 || choice < 1) {
552          printf("Invalid input. Please enter a number between 1 and 4: ");
553          scanf("%d", &choice); //read user input
554      }
555      switch(choice) {
556          case 1:
557              appendText(fileName);
558              break;
559          case 2:
560              insertText(fileName);
561              break;
562          case 3:
563              removeText(fileName);
564              break;
565          case 4:
566              displayText(fileName);
567              break;
568      }
569  }
```

1. Append Text

```
316 //function to append text onto an already existent or entirely new file
317 void appendText(char file[]) {
318
319     FILE* fPtr = fopen(file, "a"); //open file for appending
320     char text[50], c;
321     int choice;
322
323     printf("You have chosen to append text to the end the specified text file. To learn more about this
324     scanf("%d", &choice); //get user input
325     if(choice == 0) { //tell the user more about the function
326         printf("This function asks you, the user, to enter the text you'd like to append to the end of
327         printf("If this is not the function you meant to select, kindly press 0 to return to the main m
328         scanf("%d", &choice); //read user input
329         if(choice == 0) //redirect user back to main menu
330             mainMenu();
331     }
332     if(fPtr == NULL) { //if file cannot be opened
333         printf("Error opening file. Please check the spelling/existence of the file and try again.");
334     }
335     else { //file successfully opened
336         printf("File successfully opened. Please enter the text you'd like to append to the file: \n");
337
338         getchar(); //eat new line thus allowing to read user input
339         gets(text); //read the text that the user would like to append to the end of the file
340
341         fprintf(fPtr, "%s", text); //print the text to the file
342     }
343
344     fclose(fPtr); //close file
345
346     printf("Would you like to execute more operations? (y/n): ");
347     scanf("%c", &c); //get user input
348     while(c != 'y' && c != 'n') { //while invalid input
349         printf("Invalid input. Please enter y or n.");
350         scanf("%c", &c);
351     }
352     if(c == 'y') //bring user back to main menu to execute more operations
353         mainMenu();
354     else
355         return;
356 }
```

2. Insert Text

```
358 //function to insert text at a location in the file specified by the user
359 void insertText(char file[]) {
360     FILE* fPtr = fopen(file, "r+"); //open the file for reading and writing
361
362     int choice;
363     char text[50];
364     char temp[50];
365     char c;
366     memset(text, 0, sizeof(text)); //initialize array to 0's
367     memset(temp, 0, sizeof(temp)); //initialize array to 0's
368     int i = 0;
369     int pos;
370     printf("You have chosen to insert text enter by you, the user, into a specified location of the chc
371     scanf("%d", &choice); //read user input
372     if(choice == 0) { //tell user more about the function
373         printf("This function begins by asking you the location in the file you'd like to begin writing
374         printf("If this is not the function you meant to select, kindly press 0 to return to the main m
375         scanf("%d", &choice); //read user input
376         if(choice == 0) //bring user back to main menu
377             mainMenu();
378     }
379     while(fscanf(fPtr, "%c", &temp[i]) != EOF) { //load temp array with contents of the file
380         i++;
381     }
382
383     printf("\n");
384
385     printf("Please specify the character index where you'd like to start writing: ");
386     scanf("%d", &pos); //read user input
387
388     printf("Please enter the string you'd like to insert at position %d: ", pos);
389     getchar(); //eat new line thus allowing the user to input data
390     gets(text); //read the text the user would like to insert at the previously specified position
391     i=0;
392     int j = 0;
393     fseek(fPtr, 0, SEEK_SET); //set pointer at the beginning of the file
394     while(temp[i] != 0) { //read until end of text found in temp array
395
396         if(ftell(fPtr) == pos) { //if pointer is at the position where the user would like to add his t
397
398             fprintf(fPtr, "%c", temp[i]); //print last remaining character before location
399             while(text[j] != 0) { //while there is more text to add from the text the user would like t
400                 fprintf(fPtr, "%c", text[j]); //print text to file
401                 j++; //move ahead in array
402             }
403         }
404         else { //pointer is not at specified position
405             fprintf(fPtr, "%c", temp[i]);//continue printing text already found within the file
406         }
407         i++;
408     }
409
410     fseek(fPtr, 0, SEEK_SET); //set pointer to beginning of file
411     printf("Contents of file after insertion: ");
412     c = fgetc(fPtr); //print first char
413     while(c != EOF) { //while more text in file
414         printf("%c", c);
415         c = fgetc(fPtr);
416     }
417     fclose(fPtr); //close file
418
419     printf("\nWould you like to execute more operations? (y/n): ");
420     scanf("%c", &c); //get user input
421     while(c != 'y' && c != 'n') { //while invalid input
422         printf("Invalid input. Please enter y or n.");
423         scanf("%c", &c);
424     }
425     if(c == 'y') //bring user back to main menu to execute more operations
426         mainMenu();
427     else
428         return;
429 }
```

3. Remove Text

```
431 //function to remove all text from file
432 void removeText(char file[]) {
433     int choice;
434     char c;
435 
436     printf("You have chosen to remove all text from the specified text file. If you'd like to learn mor
437     scanf("%d", &choice); //read user input
438     if(choice == 0) { //tell user more about function
439         printf("This function works by first opening the file for writing purposes and then simply prin
440         if(choice == 0) //bring user back to main menu
441             mainMenu();
442     }
443     FILE* fPtr = fopen(file, "w"); //open file for writing purposes
444 
445     if(!fPtr) { //if file cannot be opened
446         printf("Error opening file. Either the file does not exist or you do not have permission to mod
447         return;
448     }
449     //else, if file can be opened
450     fprintf(fPtr, ""); //print blank to it thus emptying file
451     printf("Contents of file successfully deleted.\n");
452     fclose(fPtr); //close file
453 
454     getchar(); //eat up new line
455     printf("Would you like to execute more operations? (y/n): ");
456     scanf("%c", &c); //get user input
457     while(c != 'y' && c != 'n') { //while invalid input
458         printf("Invalid input. Please enter y or n.");
459         scanf("%c", &c);
460     }
461     if(c == 'y') //bring user back to main menu to execute more operations
462         mainMenu();
463     else
464         return;
465 }
```

4. Show File contents

```
469  //function to display the contents of a file, giving the user the option to stop at each new line encou
470  void displayText(char file[]) {
471      int choice;
472      char text[1000];
473      char c;
474      int i = 0;
475
476      printf("You have chosen to display the text of the selected file. To learn more about this function
477      scanf("%d", &choice); //read user input
478
479      if(choice == 0) { //tell user more about function
480          printf("This function first opens the file for reading purposes then reads up until a new line
481          scanf("%d", &choice); //read user input
482          if(choice == 0) //bring user back to main menu
483              mainMenu();
484      }
485
486      FILE* fPtr = fopen(file, "r"); //open file for reading
487
488      if(!fPtr) { //if file cannot be opened
489          printf("Error. File either doesn't exist or you do not have permission to read from it");
490      }
491
492      while(fscanf(fPtr, "%c", &text[i]) != EOF) { //while there is more text to read
493              if(text[i] == '\n') { //if new line is encountered
494                  printf("\nNew line encountered. Continue? (y/n)");
495                  getchar(); //eat new line to input can be entered
496                  scanf("%c", &c); //read user input
497
498                  if(c == 'n') //if user wants to stop reading
499                      exit(0); //exit
500              }
501          printf("%c", text[i]); //else, continue
502          i++;
503      }
504      printf("\n");
505      getchar();
506      printf("Would you like to execute more operations? (y/n): ");
507      scanf("%c", &c); //get user input
508      while(c != 'y' && c != 'n') { //while invalid input
509          printf("Invalid input. Please enter y or n.");
510          scanf("%c", &c);
511      }
512      if(c == 'y') //bring user back to main menu to execute more operations
513          mainMenu();
514      else
515          return;
516  }
```