PROJECT REPORT

"File Management System"

SUBMITTED BY:

SHUBHAY GAUTAM 21104051

MOGISH HASHMI 21104038

SUBMITTED TO:

Dr. Ashish Singh Parihar

Table of Contents

BACKGROUND:

File management is one of the basic and important features of an operating system. Operating systems are used to manage files of computer systems. All the files with different extensions are managed by the operating system.

A file is a collection of specific information stored in the memory of a computer system. File management is defined as the process of manipulating files in a computer system, its management includes the process of creating, modifying and deleting the files.

Our files have several common characteristics built in. Each file is made up of data, but also metadata is embedded into the file to help the operating system (OS) manage how the file works and how it is stored. Metadata records file information such as the author, file creation date, modified date, and file size.

File management is a process of maintaining any kind of records in a proper manner like your work document or your money records. This is the process to divide things in different stages and in writing so that in future when needed it will be easy to get that particular record.

In the 20th century, vertical filing cabinets were introduced to store a different kind of files. Then the computer was used to store a different kind of file in the system with the help of the LAN/wan network. Then portable flash drives were introduced to store files and to transfer data from one system to another. Then cloud storage was introduced. This cloud storage made it easy to store files from anywhere and from any computer. This prevents the user from losing the data or from any data threat with their high-security methods.

INTRODUCTION:

Computer users spend time every day interacting with digital files and folders, including creating, downloading, naming, moving, saving, copying, reviewing, navigating, searching for, sharing, and deleting them. This activity, called *file management* (FM).

File management is an art of storing, naming, sorting and handling documents files in a systematic manner. So that in future it will be easy to retrieve data.

A ***file management system*** is a type of software that manages data files in a computer system. It has limited capabilities and is designed to manage individual or group files, such as special office documents and records.

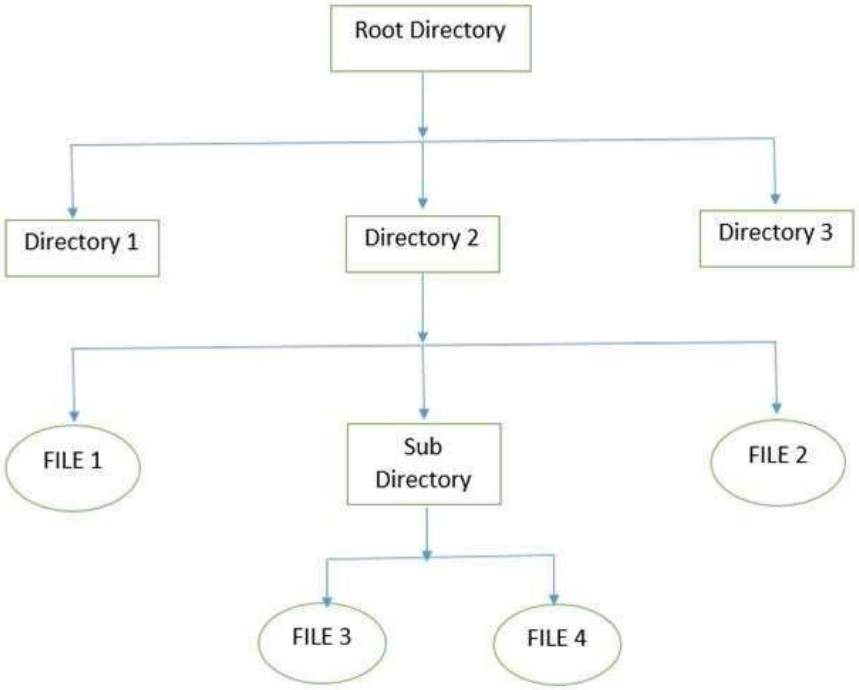
The following are some of the tasks performed by file management of operating system of any computer system:

1. It helps to create new files in the computer system and place them at specific locations.
2. It helps in easily and quickly locating these files in the computer system.
3. It helps to store the files in separate folders known as directories. These directories help users to search files quickly or to manage the files according to their types or uses.
4. It helps the user to modify the data of files or to modify the name of the file in the directories etc.

File management helps users to organize their valuable documents in a systematic manner for better and efficient use of it.

PROJECT PARADIGM

The file is actually the collection of associated information. This file-system is prearranged into a directory for efficient usage. Every directory has a number of files and other directories. The directory is defined as a bit which distinguishes the entries that explained file and subdirectories in the recent directory. By theoretically we may change the file into a directory by changing its bit. A file system is considered as an element of an operating system that manages the storage space and operation of files on media like disks.



The above figure shows the general hierarchy of the storage in an operating system. In this figure the root directory is present at the highest level in the hierarchical structure. It includes all the subdirectories in which the files are stored. Subdirectory is a directory present inside another directory in the file storage system. The directory base storage system ensures better organization of files in the memory of the computer system.

MECHANISM AND WORKING:

Menu Code:

```
#include <stdio.h> int

main(void) {

printf("================================================================================
\n");

        printf("    ------------------------File Management Project        -------------------
--------\n");

printf("================================================================================
\n");

        printf("Welcome, The Main Menu is given below:\n");


                printf("1- List all Files and Directories\n"); printf("2- Create

                New Files\n");

                printf("3- Delete Existing Files\n"); printf("4-

                Rename Files\n"); printf("5- Edit File Content\n");

                printf("6- Search Files\n");

                printf("7- Details of Particular File\n"); printf("8- View

                Content of File\n"); printf("9- Sort File Content\n");

                printf("10- List only Directories(Folders)\n"); printf("11- List Files of

                Particular Extension\n"); printf("12- Count Number of

                Directories\n"); printf("13- Count Number of Files\n");

                printf("14- Sort Files in a Directory\n"); printf("0-

                Exit\n");
```

```
                printf("\nWhat action you want to Perform?\nEnter 1-14\n");;
return 0;
}
```

Main Code:

```
#!/bin/bash i="0"
while [ $i -lt 100 ] do
gcc project.c -o proj
./proj read
opt1
if [ $opt1 == 1 ] then
        echo "List all files and Directories here.." echo "Showing all
        files and directories.                                  "
        sleep 3
        echo "Loading.." sleep 3
        echo "  ---------------------------------OutPut ------------------------
----------"
        ls
        echo " "


elif [ $opt1 == 2 ] then
        echo "Create New Files here.."
```

```
        echo "Which type of file you want to create !" echo "1- .c"
        echo "2- .sh"
echo "3- .txt"
echo "Enter your choice from 1-3" read
filechoice

        if [ $filechoice == 1 ] then
        echo "Enter File Name without .c Extension" read filename
        touch $filename.c
        echo "  ---------------------------------OutPut ---------------------------
-----------"
        echo "File Created Successfully" echo " "

        elif [ $filechoice == 2 ] then
        echo "Enter File Name without .sh Extension" read
        filename2
        touch $filename2.sh
        echo "  ---------------------------------OutPut ---------------------------
-----------"
        echo "File Created Successfully" echo " "

        elif [ $filechoice == 3 ] then
```

```
        echo "Enter File Name without .txt Extension" read filename3
        touch $filename3.txt
        echo "  ---------------------------------OutPut ----------------------------
-----------"
        echo "File Created Successfully" echo " "
        else
        echo "Inavlid Input..Try Again." echo " "
        fi

elif [ $opt1 == 3 ] then
        echo "Delete existing files here.. "
        echo "Enter name of File you want to Delete!"
        echo "Note: Please Enter full Name with Extension." read delfile
        echo "  ---------------------------------OutPut ----------------------------
-----------"


        if [ -f "$delfile" ]; then
        if [ -f "$delfile" ]; then
                rm $delfile
                echo "Successfully Deleted."
echo " "
```

```
        else
echo "File Does not Exist..Try again" echo " "
fi


elif [ $opt1 == 4 ] then

        echo "  ---------------------------------OutPut ---------------------------
-----------"

        echo "Rename files here.."

        echo "Enter Old Name of File with Extension.." read old

        echo "Checking for file..." sleep 3


        if [ -f "$old" ]; then

                echo "Ok File Exist."

                echo "Now Enter New Name for file with Extension" read new

                mv $old $new

                echo "Successfully Rename."

                echo "Now Your File Exist with $new Name"

        else

                echo "$old does not exist..Try again with correct filename."

        fi


echo " "
```

```
elif [ $opt1 == 5 ] then
        echo "Edit file content here.."
        echo "Enter File Name with Extension : " read edit
        echo "  ----------------------------------OutPut ----------------------------
-----------"
        echo "Checking for file.." sleep 3

        if [ -f "$edit" ]; then
                echo "Opening file.." sleep 3
                nano $edit
                echo " "
        else
                echo "$edit File does not exist..Try again."
        fi
elif [ $opt1 == 6 ] then
echo "Search files here.."
echo "Enter File Name with Extension to search" read f

        echo "  ----------------------------------OutPut ----------------------------
-----------"
        if [ -f "$f" ]; then
```

```
                echo "Searching for $f File" echo "File
                Found."
                find /home -name $f echo " "
        else
                echo "File Does not Exist..Try again." echo " "
        fi


elif [ $opt1 == 7 ] then
then
        echo "Detail of file here.."
        echo "Enter File Name with Extension to see Detail : " read detail
        echo "  ---------------------------------OutPut ----------------------------
-----------"
        echo "Checking for file.." sleep 4

        if [ -f "$detail" ]; then
                echo "Loading Properties.." stat $detail

        else
                echo "$detail File does not exist..Try again"
```

```
        fi
echo " "



elif [ $opt1 == 8 ] then
        echo "View content of file here.." echo "Enter File
        Name : "
        read readfile
        echo "  ----------------------------------OutPut ----------------------------
-----------"
        if [ -f "$readfile" ]; then
echo "Showing file content.."
                sleep 3
                cat $readfile
else
                echo "$readfile does not exist"
fi
echo " "


elif [ $opt1 == 9 ] then
        echo "Sort files content here.."
        echo "Enter File Name with Extension to sort :" read sortfile
        echo "  ----------------------------------OutPut ----------------------------
-----------"
        if [ -f "$sortfile" ]; then
```

```
                echo "Sorting File Content.." sleep 3
                sort $sortfile


        else
                echo "$sortfile File does not exist..Try again."
        fi
echo " "



elif [ $opt1 == 10 ] then
        echo "  ---------------------------------OutPut ---------------------------
----------"
        echo "List of all Directories here.." echo "showing
        all Directories..." echo "Loading.."
        sleep 3 ls
        -d */ echo
        " "


elif [ $opt1 == 11 ] then


        echo "List of Files with Particular extensions here.." echo "Which type of file
        list you want to see?"
        echo "1- .c"
        echo "2- .sh"
```

```
        echo "3- .txt"

        echo "Enter your choice from 1-3" read extopt


        echo "  ---------------------------------OutPut ------------------------
-----------"
if [ $extopt == 1 ] then

        echo "List of .c Files shown below." echo
"Loading.."

        sleep 3
ls *.c
elif [ $extopt == 2 ] then

        echo "List of .sh Files shown below." echo

        "Loading.."

        sleep 3 ls

        *.sh


        elif [ $extopt == 3 ] then

        echo "List of .txt Files shown below." echo

        "Loading.."

        sleep 3 ls

        *.txt


        else

        echo "Invalid Input..Try again.."
fi

echo " "
```

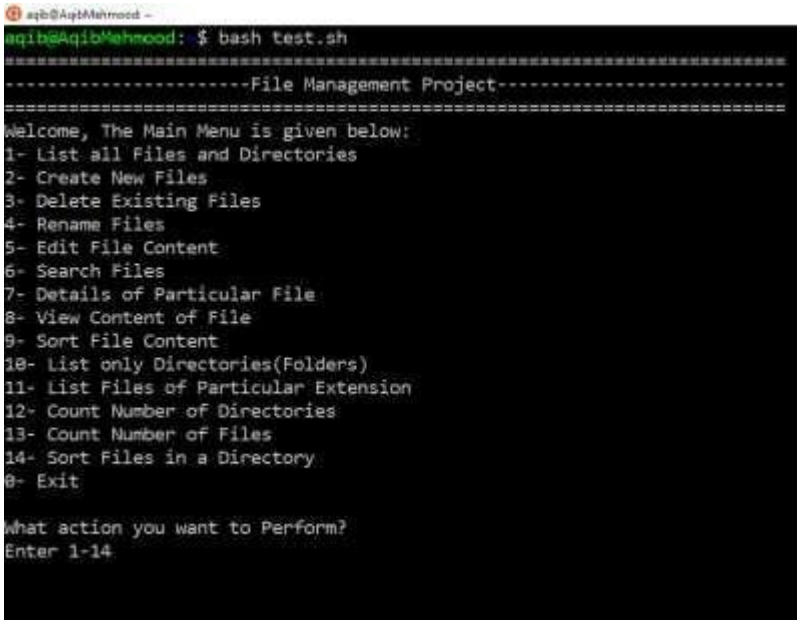
```
elif [ $opt1 == 12 ] then
        echo "  ----------------------------------OutPut ------------------------
-----------"
        echo "Total number of Directories here.." echo "Loading all
        directories.."
        sleep 3
        echo "Counting.." sleep
        3
        echo "Number of Directories are : " echo */ | wc
        -w
echo " "

elif [ $opt1 == 13 ] then
        echo "  ----------------------------------OutPut ------------------------
-----------"
        echo "Total Numbers of Files in Current Directory here.." echo "Loading all
        files.."
        sleep 3
        echo "Number of Files are : "
        ls -l | grep -v 'total' | grep -v '^d' | wc -l
echo " "
elif [ $opt1 == 14 ] then
echo "  ----------------------------------OutPut ------------------------------
---"
```

```
        echo "Sort Files here.."
echo "Your Request of Sorting file is Generated." echo "Sorting.."
sleep 3

        ls | sort echo " "


elif [ $opt1 == 0 ] then

        echo "Good Bye.."

        echo "Successfully Exit" break

else

echo "Invalid Input..Try again.                    "

fi i=$[$i+1]

done
```

Output Screenshots:

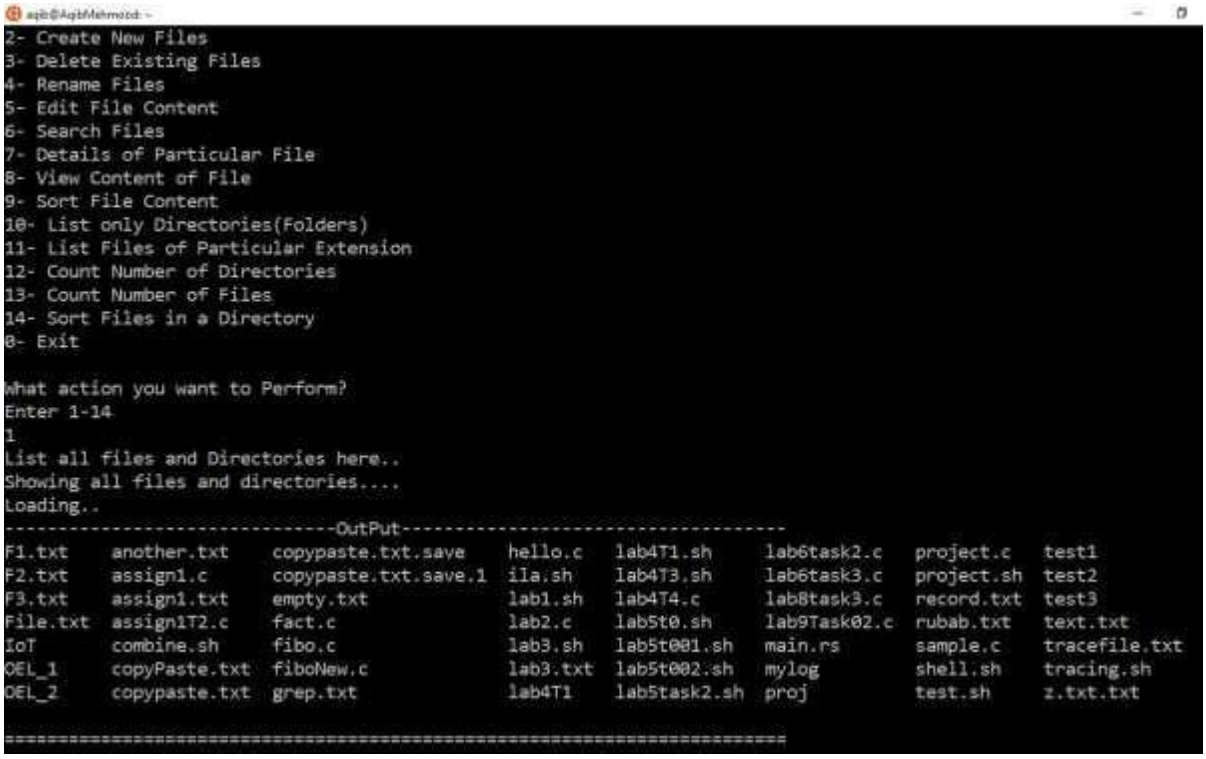
Main Menu:

Main menu of Project that displays all the available options to the users. The users need to choose one out of 14 and the particular command will be executed according to the user input.



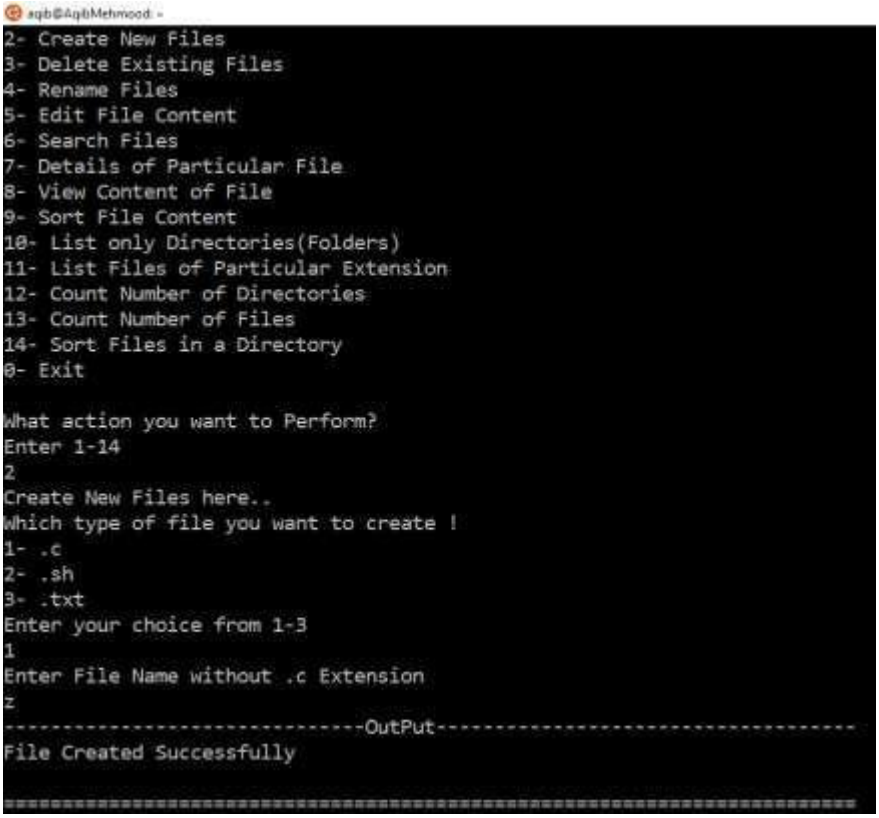
Choice 01 Output:

If user enter 1 then the List of all Files and Directories will be displayed.



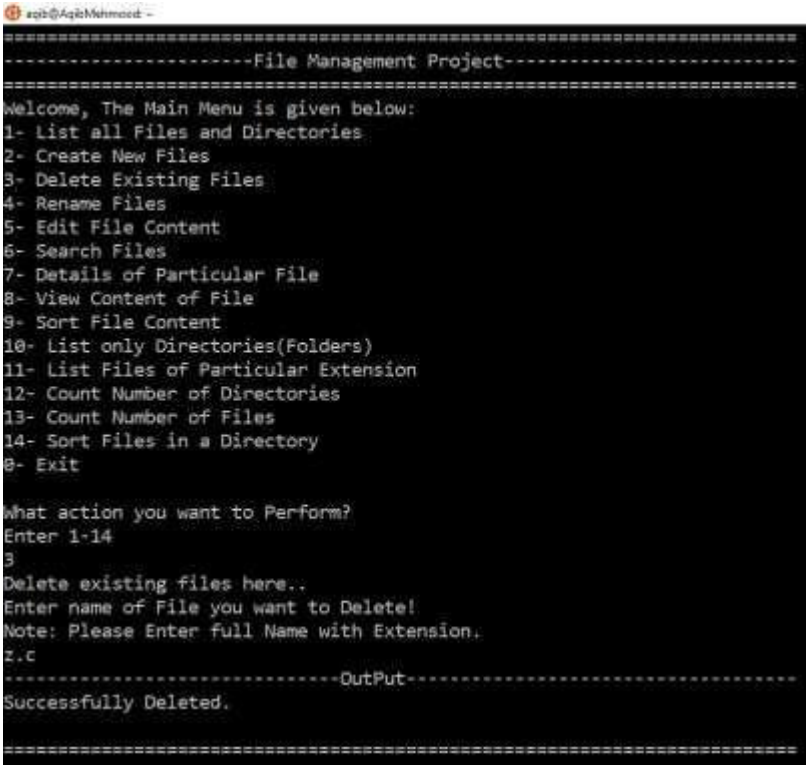
Choice 02 Output:

If the user wants to create a new file then he needs to enter 2.



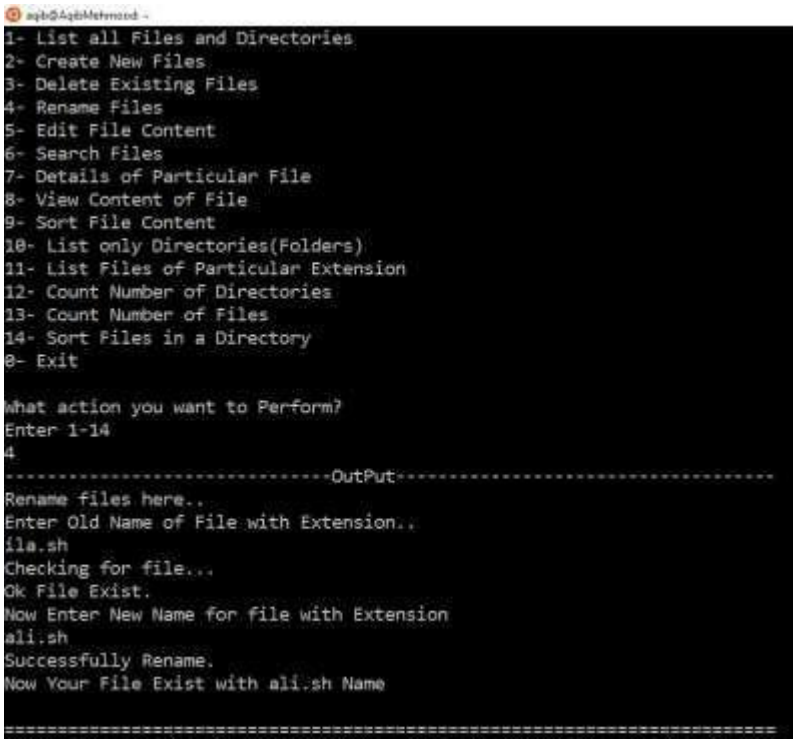
Choice 03 Output:

If a user wants to delete an existing file then he needs to enter 3.



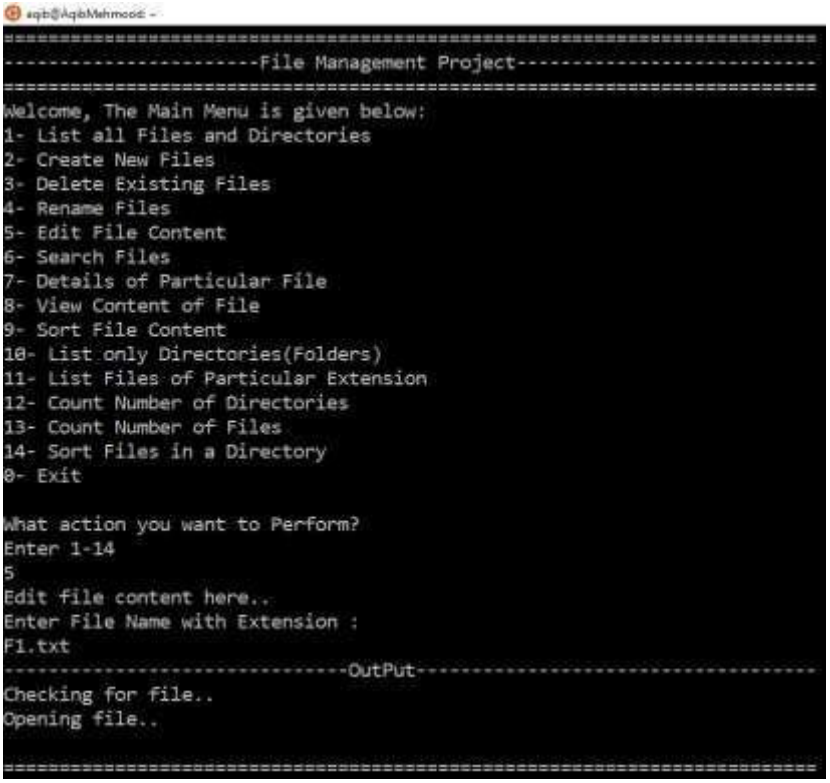
Choice 04 Output:

If user wants to rename an existing file then he needs to enter 4.



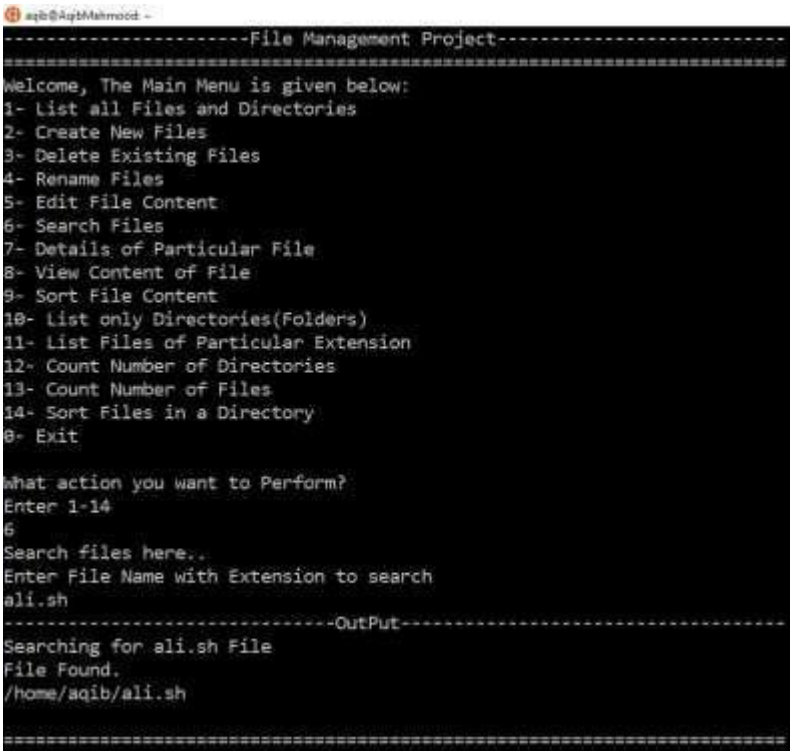
Choice 05 Output:

If a user wants to edit file content then he needs to enter 5.



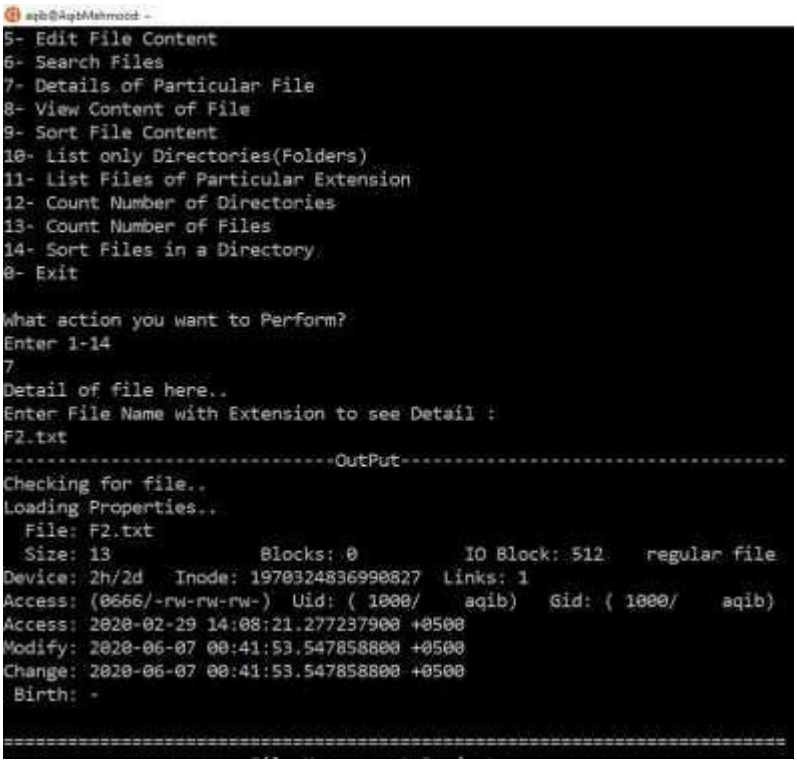
Choice 06 Output:

If user wants to search for a file then he needs to enter 6.



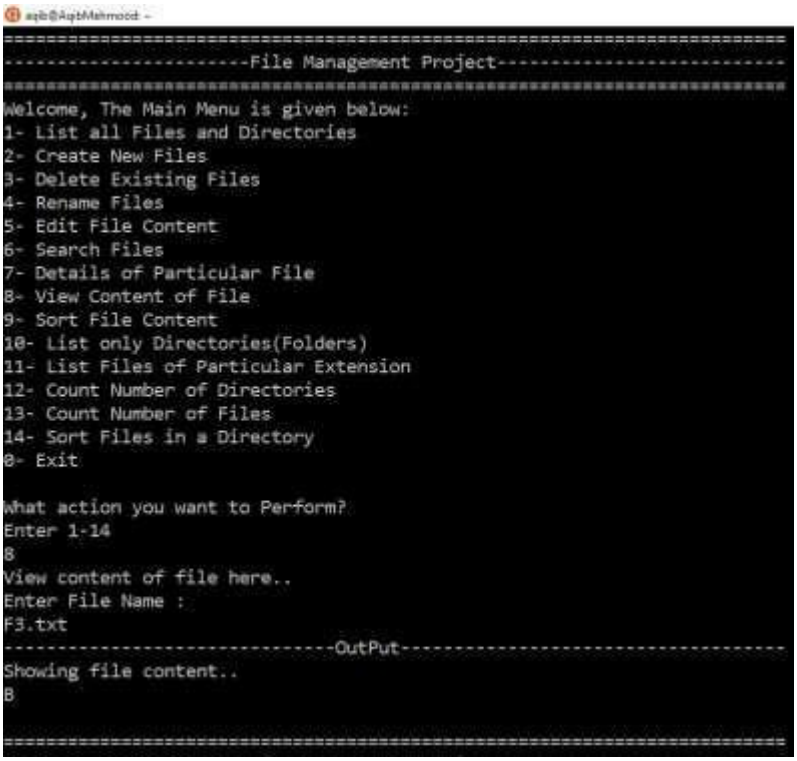
Choice 07 Output:

If the user wants to see the details of the file then he needs to enter 7.



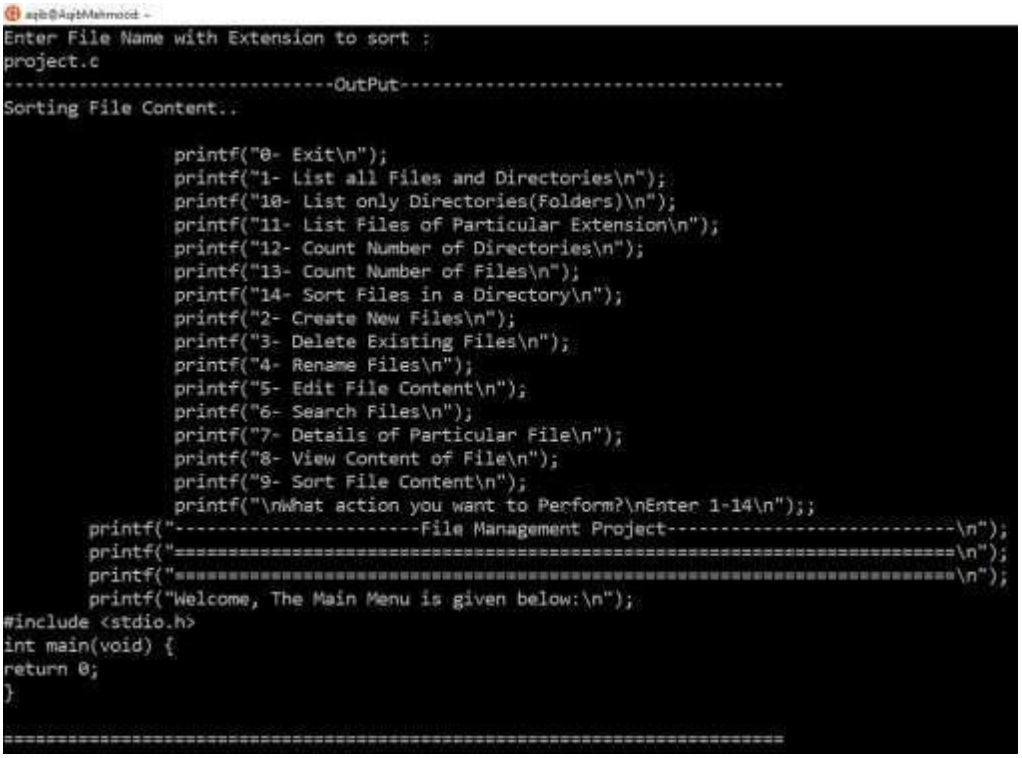
Choice 08 Output:

If the user wants to view the content of the file then he needs to enter 8.



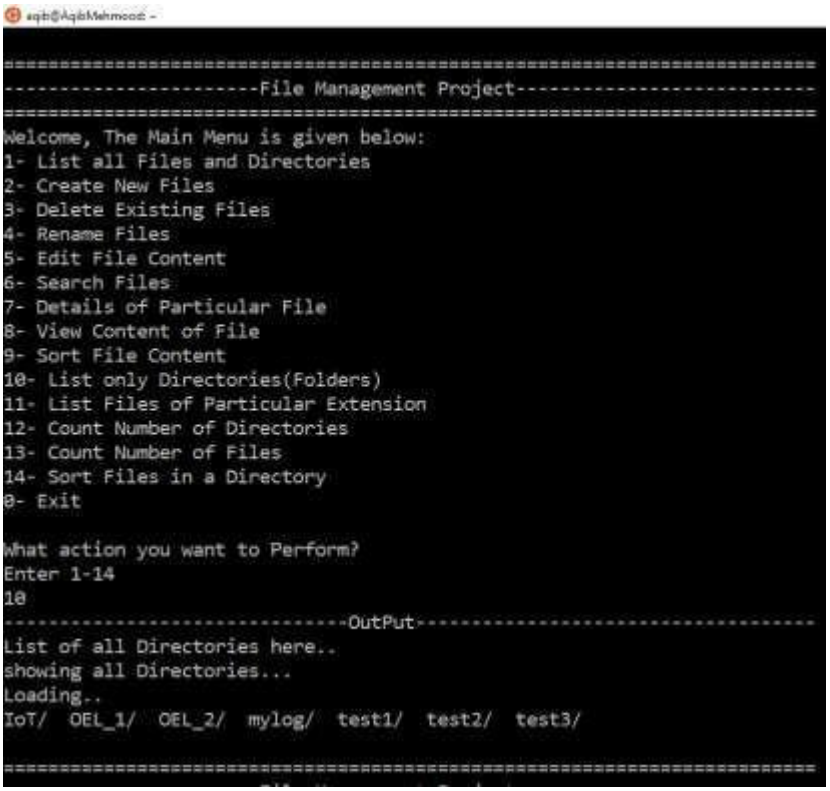
Choice 09 Output:

If the user wants to sort the file content then he needs to enter 9.



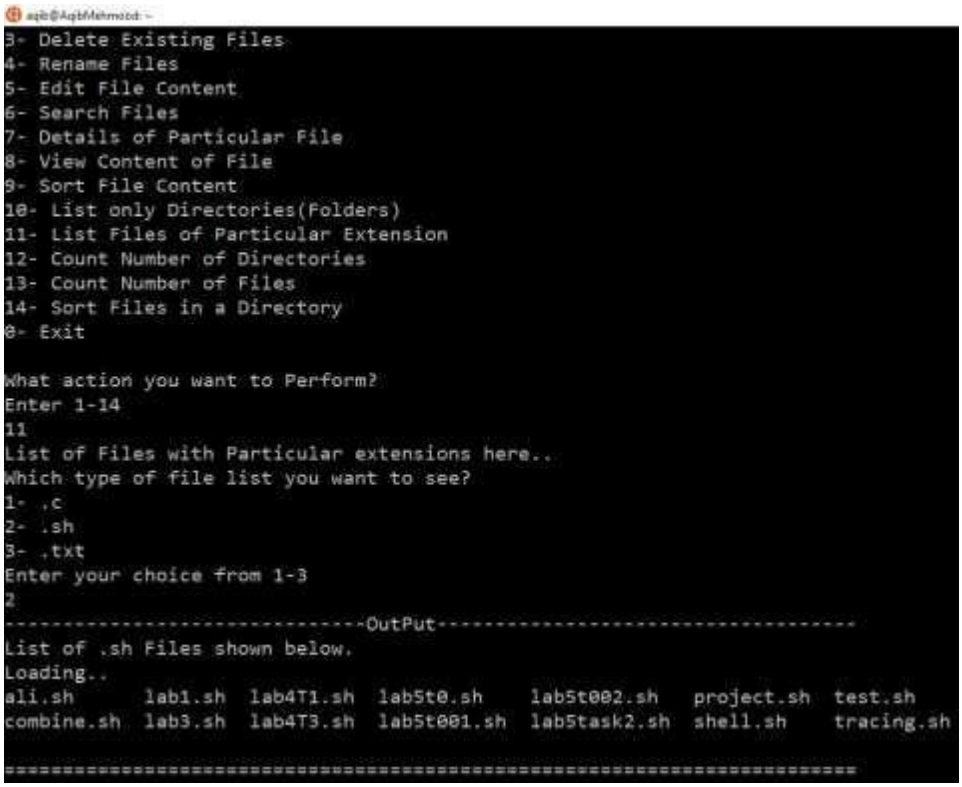
Choice 10 Output:

If user wants to list all directories then he needs to enter 10.



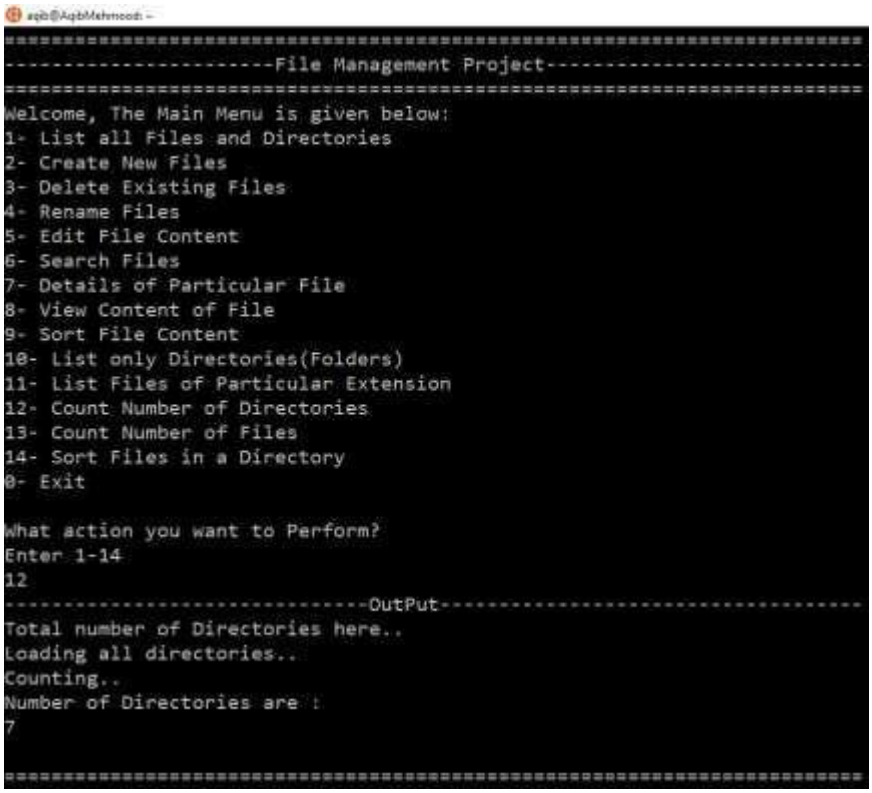
Choice 11 Output:

If a user wants to list all files with the same extension then he needs to enter 11.



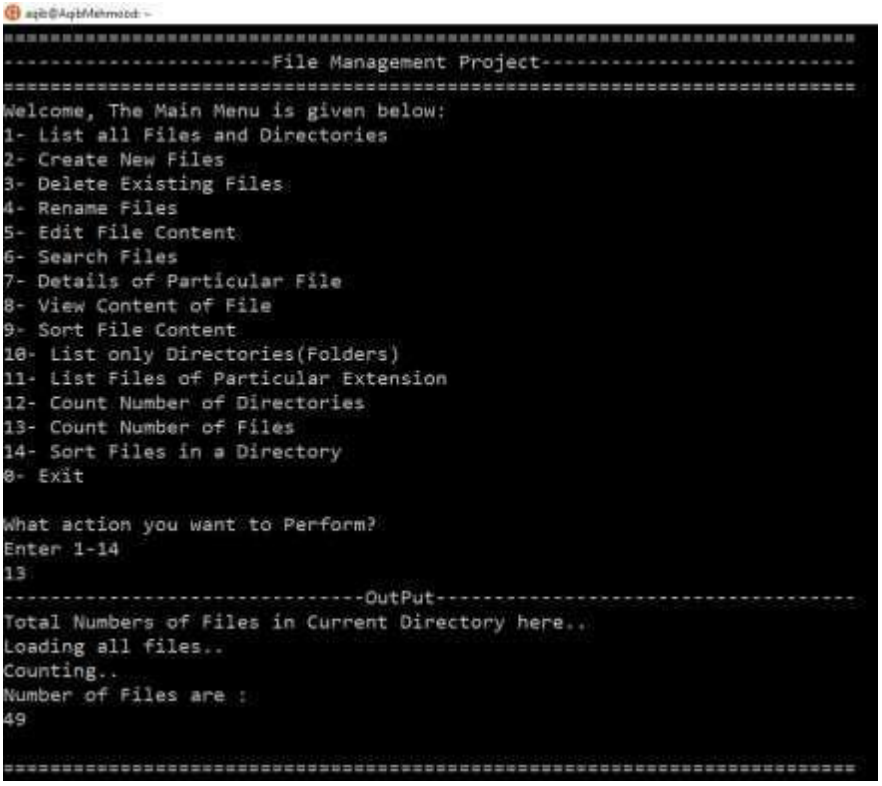
Choice 12 Output:

If the user wants a number of directories then he needs to enter 12.



Choice 13 Output:

If user wants to count number of files then he needs to enter 13.

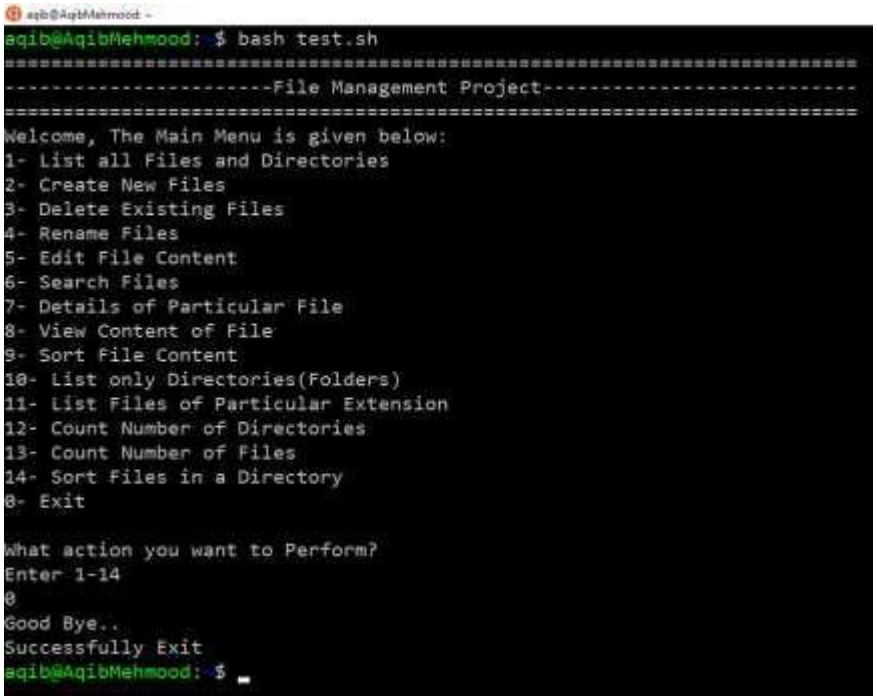


Choice 14 Output:

If the user wants to sort all files in a directory then he needs to enter 14.

Exit option:

If user wants to exit from Management system then he needs to enter 0.



FUNCTIONALITIES:

The following are some of the functionalities or tasks performed by file management system:

1. List all Files and Directories.
2. Create New Files.
3. Delete Existing Files.
4. Rename an Existing Files.
5. Edit Files Content.
6. Search for Files.
7. Details of Particular File.
8. View Content of File.
9. Sort Files Content.
10. List only Directories.
11. List Files of particular Extension.
12. Count Number of Directories.
13. Sort all Files in a Directories.

The details of all above functionalities are already explained under MECHANISM AND WORKING heading in the form of code of each function.

REMAINING CODE MODULES, API'S AND PLATFORMS:

No other remaining side work apart from the displayed work above is used in this project. All the functionalities and code of each function is explained above. In this project we use Ubuntu subsystem terminal with C language and bash scripting. So no other platform, API or plugins used in this project.

FUTURE WORK:

This is the most basic version of the file management system. So in future we can improve the current version's functionalities and can add more new functionalities to the system. In the current version of the files management system there are 13 different options for a user to manage files and directories. In the future we can add more choices for users by understanding the advanced concept about file management in the Linux operating system. So this will definitely help users to manage files in a more easy and comfortable manner.

CONCLUSION:

The project contains some basic functionalities regarding file management like creating new files, delete existing files, rename files, edit files, read or write files and so on. All the functionalities are working on the basis of user's input from keyboard. There are different basic functions that users can perform on files. These functions are written in C language and bash scripting. All these functionalities are discussed above in the form of code as well as in simple natural language. So everyone having the basic knowledge of computer can use this file management system to perform different functions on files.

REFERENCES:

- Main Idea from includehelp.com Submitted by Amit Shukla, on August 14, 2017 < https://www.includehelp.com/operating-systems/file-management-in-operating-system.aspx >
- How to rename a file answer by Mazhar MIK on askubuntu.com

  <https://askubuntu.com/questions/280768/how-to-rename-a-file-in-terminal#:~:text=A%20simple%20way%20to%20rename,from%20one%20name%20to%20another.&text=where%20%E2%80%9Cfile1.>
- Use of stat command answer by <https://linuxhint.com/linux_stat_command/>

~~~~~~**/THE END/**~~~~~~
~~~~~~