THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

OPERATING SYSTEMS: UCS303
Thapar Institute of Engineering and Technology, Patiala
(Summer Semester 2024)

PROJECT REPORT ON:

DOCUMENT ADMINISTRATION- FILES & DIRECTORIES

SUBMITTED BY:
Ravneet Kaur - 102203202
Ruhani Grover - 102203864
Saksham Mutneja - 102103359

SUBMITTED TO:
Dr. Shashank S Singh

TABLE OF CONTENTS

ABSTARCT

The Document Administration System is designed with an intuitive user interface that accommodates both novice and experienced users. Its flexibility and scalability make it suitable for diverse environments, from personal use to enterprise-level applications.

In summary, this enhances the efficiency of digital file handling by providing a structured approach to file and directory management. Its comprehensive features streamline daily tasks, improve data organization, and ensure secure access, ultimately supporting better data management practices and operational efficiency.

This project on Document Administration- Files & Directories was executed under the guidance of and valuable inputs from Dr. Shashank S Singh.

INTRODUCTION

Document Handling is a cornerstone of modern operating systems, essential for organizing and manipulating data stored on computer systems. It encompasses tasks such as creating, modifying, and deleting files, ensuring that information is stored efficiently and accessed easily. Each file contains both data and metadata, including details like authorship, creation and modification dates, and size, facilitating effective management by the operating system (OS).

Throughout history, the evolution of file management has paralleled advancements in technology. From traditional vertical filing cabinets to the digital age, where LAN/WAN networks enabled file storage and sharing across computers, innovations like portable flash drives and cloud storage revolutionized accessibility and security. Cloud storage, for instance, provides users the flexibility to store and retrieve files from anywhere, safeguarding against data loss with robust security measures.

In practical terms, file management involves a range of everyday activities for computer users. These include creating new files at specified locations, locating files swiftly through directories, and organizing files based on their types or uses. It also facilitates tasks such as modifying file data or renaming files within directories. Such systematic organization ensures that valuable documents are readily accessible and effectively utilized, enhancing productivity and workflow efficiency.

A document management system, typically integrated within the OS, serves as specialized software designed to oversee and manipulate data files. It supports the creation, organization, and retrieval of individual or group files, such as office documents and records. By streamlining these operations, file management systems contribute significantly to optimizing user interaction with digital files, promoting seamless workflow management and data integrity.

MOTIVATION

The primary motivation behind this project is to develop an efficient and user-friendly Document Administration System that addresses the challenges faced by users in organizing and accessing their files. Elaborating on the motivation behind developing a Document Administration System, we delve deeper into the reasons why such a system is needed and the benefits it offers to the users.

1. **Organizational Challenges:**
   Many individuals and organizations face significant challenges when it comes to managing an ever-increasing volume of digital files. Traditional file systems often rely on hierarchical folder structures that can become convoluted and difficult to navigate as the number of files grows. This leads to wasted time and effort spent searching for specific files, resulting in decreased productivity.
2. **File Disorganization:**
   Without a proper file management system, files tend to become disorganized, making it challenging to find and retrieve specific documents when needed. This lack of organization can result in duplicate files, outdated versions, and inconsistent naming conventions, leading to confusion and errors.
3. **Time Efficiency:**
   Efficient document management is crucial for maximizing productivity. The ability to quickly locate files, access relevant information, and collaborate seamlessly with team members saves valuable time and allows individuals and organizations to focus on more critical tasks.

4. **Enhanced User Experience:**
   Traditional file systems often have complex and unintuitive interfaces, causing frustration and confusion among users. By developing a user-friendly FMS with an intuitive design and features that cater to users' needs, we aim to provide a seamless and pleasant user experience, promoting efficiency and satisfaction.
5. **Advanced File Categorization and Tagging:**
   A modern FMS can incorporate advanced file categorization techniques, such as metadata tagging. This allows users to organize files based on specific attributes like file type, date or keywords.

OBJECTIVE

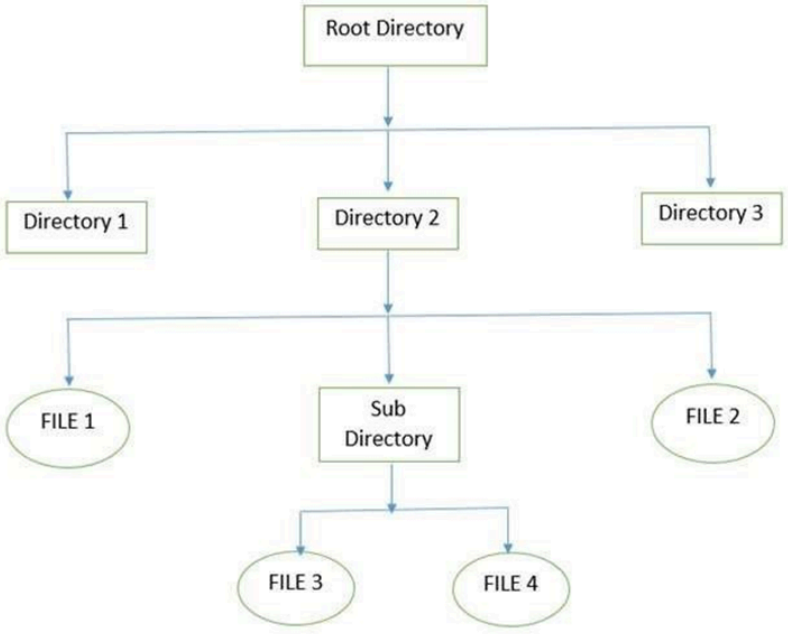
This project's scope encompasses the design, development, and implementation of a robust file management system.
The key objectives include:

1. **Developing a user-friendly interface:**
   Create an intuitive and visually appealing interface that allows users to easily navigate, organize, and search their files.
2. **File categorization and tagging:**
   Implement advanced file categorization techniques, such as metadata tagging, to enable users to organize files based on specific attributes, such as file type, date, or keywords.
3. **Efficient search functionality:**
   Develop a powerful search feature that enables users to quickly locate files based on various search criteria, including file names, content, tags and metadata.
4. **Version control and backup:**
   Implement version control mechanisms to enable users to track and manage multiple versions of files. Additionally, it provides automatic backup capabilities to prevent data loss and ensure file integrity.
5. **Security and access control:**
   Incorporate robust security measures, including user authentication, access control, and encryption, to protect sensitive files from unauthorized access and ensure data privacy.

METHODOLOGY

The file is actually the collection of associated information. This file system is prearranged into the directory for efficient usage. Every directory has a number of files and other directories. The directory is defined as a bit that distinguishes the entries that explain the files 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 an element of an operating system that manages the storage space and operation of files on media like disks. The figure below shows the general hierarchy of 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-based storage system ensures better file organization in the memory of the computer system.



CODING IMPLEMENTATION

MAIN FUNCTION:

```
#include <stdio.h>

int main(void)
{
   printf("-*-File Management Project-*-\n");
   printf("Welcome, 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 function to Perform? Enter 1-14:\n");
   return 0;
}
```

FUNCTION 1:

```
#!/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
```

FUNCTION 2:

```
elif [ $opt1 == 2 ]
then
echo "Create New Files here. Which type of file?"
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"
```

```
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”
else
echo ”Inavlid Input. Try Again.”
fi
```

FUNCTION 3:

```
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
rm $delfile
echo ”Successfully Deleted.”
else
echo ”File Does not Exist..Try again”
echo ” ”
fi
```

FUNCTION 4:

```
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
else
fi
echo ”Ok File Exists.”
echo ”Now Enter New Name for file with Extension:”
read new
mv $old $new
echo ”Successfully Rename.”
echo ”Now Your File Exist with $new Name”
echo ”$old does not exist...Try again with correct
filename.”
```

FUNCTION 5:

```
elif [ $opt1 == 5 ]
then
echo ”Edit file content here..”
echo ”Enter File Name with Extension : ”
read edit
echo ”-----OutPut-----”
echo ”Checking for file...”
if [ -f ”$edit” ];
then
else
fi
echo ”Opening file...”
nano $edit
echo ”$edit File does not exist...Try again.”
```

FUNCTION 6:

```
elif [ $opt1 == 6 ]
then
echo ”Search files here..”
echo ”Enter File Name with Extension to search”
read f
echo ”-----OutPut-----”
if [ -f ”$f” ];
then
else
fi
echo ”Searching for $f File”
echo ”File Found.”
find /home -name $f
echo ” ”
echo ”File Does not Exist..Try again.”
echo ” ”
```

FUNCTION 7:

```
elif [ $opt1 == 7 ]
then
echo ”Details of file here..”
echo ”Enter File Name with Extension to see Detail : ”
read detail
echo “-----OutPut-----”
echo ”Checking for file...”
sleep 4
if [ -f ”$detail” ];
then
else
echo ”Loading Properties..”
stat $detail
echo ”$detail File does not exist..Try again”
fi echo ” ”
```

FUNCTION 8:

```
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..”
else
fi
echo ” ”
sleep 3
cat $readfile
echo ”$readfile does not exist”
```

FUNCTION 9:

```
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
else
fi echo ” ”
echo ”Sorting File Content..”
sleep 3
sort $sortfile
echo ”$sortfile File does not exist..Try again.”
```

FUNCTION 10:

```
elif [ $opt1 == 1Ø ]
then
echo ”-----OutPut-----”
echo ”List of all Directories here...”
echo ”showing all Directories...”
echo ”Loading...”
sleep 3
ls -d */
```

FUNCTION 11:

```
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 ” ”
```

FUNCTION 12:

```
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 ” ”
```

FUNCTION 13:

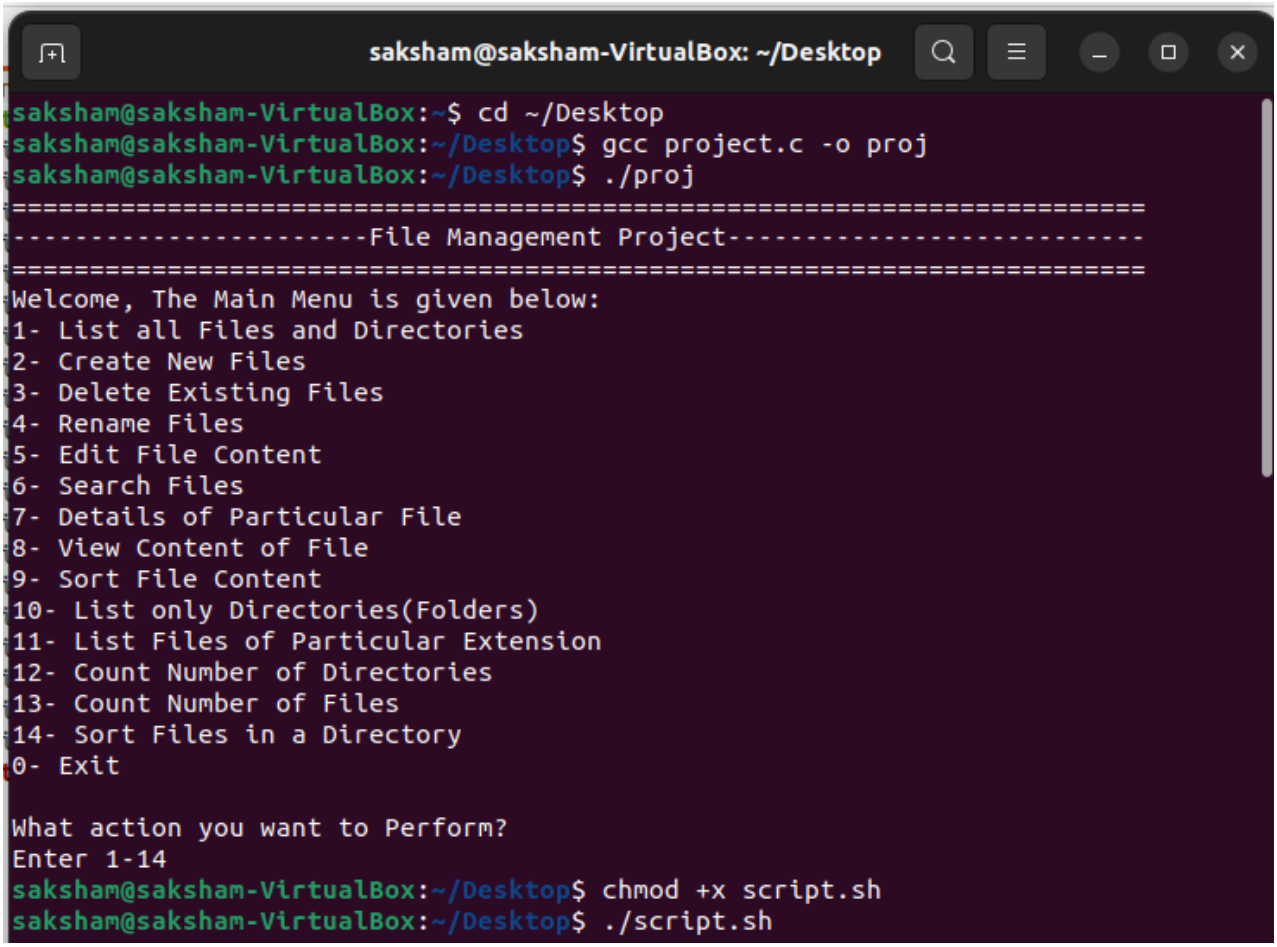
```
elif [ $opt1 == 13 ]
then
echo “-----OutPut-----”
echo ”Total Numbers of Files in Current Directory
here...”
echo ”Loading all files...”
sleep 3
echo ”Number of Files are : ”
ls -l | grep -v ’total’ | grep -v ’^d’ | wc -l
echo ” ”
```

FUNCTION 14:

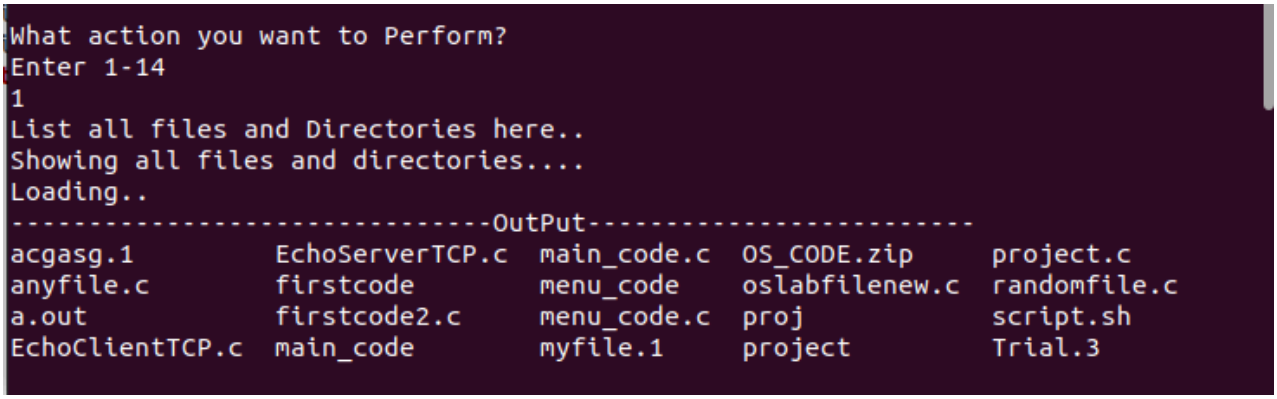
```
elif [ $opt1 == 14 ]
then
echo “-----OutPut-----”
echo ”Sort Files here...”
echo ”Your Request of Sorting file is Generated.”
echo ”Sorting...”
sleep 3
ls | sort
echo ” ”
elif [ $opt1 == Ø ]
then
echo ”Good Bye..”
echo ”Successfully Exit”
break
else
echo ”Invalid Input. Try again....”
fi
i=$[$i+1]
done
```

RESULT ANALYSIS

MAIN MENU FUNCTION:



FUNCTION 1: List all files and directories



FUNCTION 2: Create new files

```
What action you want to Perform?
Enter 1-14
2
Create New Files here..
Which type of file you want to create !
1- .c
2- .sh
3- .txt
Enter your choice from 1-3
1
Enter File Name without .c Extension
osprojectcode
--------------------------------OutPut-------------------------
File Created Successfully
```

FUNCTION 3: Delete existing files

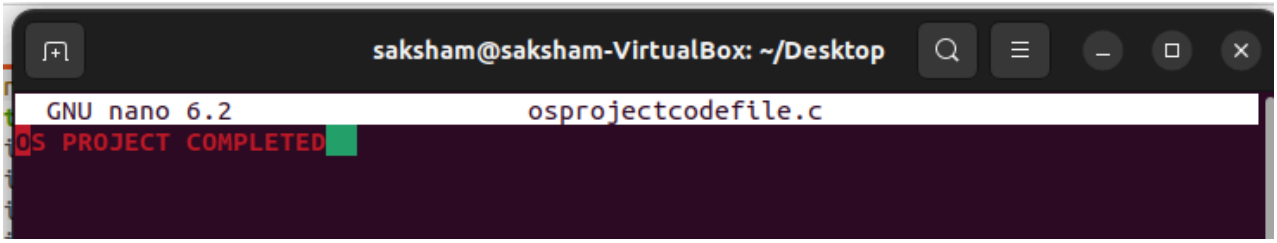
```
What action you want to Perform?
Enter 1-14
3
Delete existing files here..
Enter name of File you want to Delete!
Note: Please Enter full Name with Extension.
osprojectcode.c
--------------------------------OutPut-------------------------
Successfully Deleted.
```

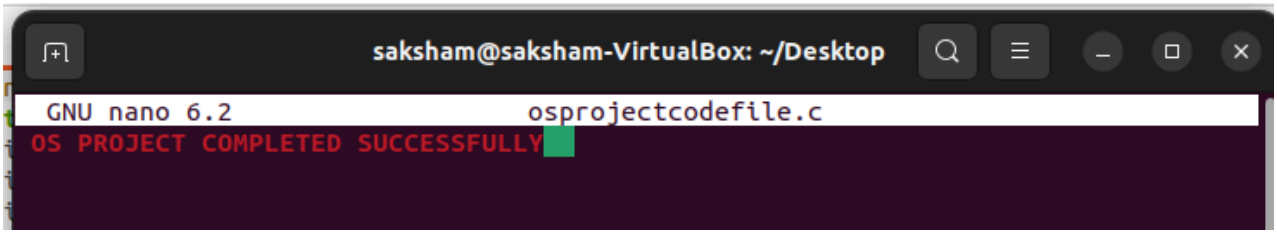
FUNCTION 4: Rename files

```
What action you want to Perform?
Enter 1-14
4
Rename files here..
Enter Old Name of File with Extension..
randomfile.c
Checking for file...
Ok File Exist.
Now Enter New Name for file with Extension
osprojectcodefile.c
Successfully Rename.
Now Your File Exist with osprojectcodefile.c Name
```

FUNCTION 5: Edit file content

```
What action you want to Perform?
Enter 1-14
5
Edit file content here..
Enter File Name with Extension :
osprojectcodefile.c
--------------------------------OutPut-------------------------
Checking for file..
Opening file..
```





FUNCTION 6: Search for files

```
What action you want to Perform?
Enter 1-14
6
Search files here..
Enter File Name with Extension to search
firstcode.c
--------------------------------OutPut-------------------------
File Does not Exist..Try again.
```

```
What action you want to Perform?
Enter 1-14
6
Search files here..
Enter File Name with Extension to search
project.c
--------------------------------OutPut-------------------------
Searching for project.c File
File Found.
/home/saksham/Desktop/project.c
```

FUNCTION 7: Fetch details of a file

```
What action you want to Perform?
Enter 1-14
7
Detail of file here..
Enter File Name with Extension to see Detail :
randomfile.c
------------------------------OutPut------------------------
Checking for file..
Loading Properties..
  File: randomfile.c
  Size: 23              Blocks: 8          IO Block: 4096   regular file
Device: 803h/2051d      Inode: 1241840     Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/ saksham)   Gid: ( 1000/ saksham)
Access: 2024-07-14 23:51:52.724079049 +0530
Modify: 2024-07-14 23:51:30.952041078 +0530
Change: 2024-07-14 23:51:30.952041078 +0530
 Birth: 2024-07-14 23:28:09.535932805 +0530
```

FUNCTION 8: View contents of a file

```
What action you want to Perform?
Enter 1-14
8
View content of file here..
Enter File Name :
randomfile.c
------------------------------OutPut------------------------
Showing file content..
OS PROJECT COMPLETED
```

FUNCTION 9: Sort file content

```
What action you want to Perform?
Enter 1-14
9
Sort files content here..
Enter File Name with Extension to sort :
project.c
------------------------------OutPut------------------------
Sorting File Content..
#include <stdio.h>
int main(void) {
printf("0- Exit\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("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("----------------------File Management Project--------------------------\n");
printf("=============================================================================\n");
printf("=============================================================================\n");
printf("\nWhat action you want to Perform?\nEnter 1-14\n");;
printf("Welcome, The Main Menu is given below:\n");
return 0; }
```

FUNCTION 10: List all directories

```
What action you want to Perform?
Enter 1-14
10
List of all Directories here..
showing all Directories...
Loading..
------------------------------OutPut-------------------------
operating/  system/
```

FUNCTION 11: List files with a particular extension

```
What action you want to Perform?
Enter 1-14
11
List of Files with Particular extensions here..
Which type of file list you want to see?
1- .c
2- .sh
3- .txt
Enter your choice from 1-3
1
------------------------------OutPut-------------------------
List of .c Files shown below.
Loading..
anyfile.c        EchoServerTCP.c  main_code.c  oslabfilenew.c  randomfile.c
EchoClientTCP.c  firstcode2.c     menu_code.c  project.c
```

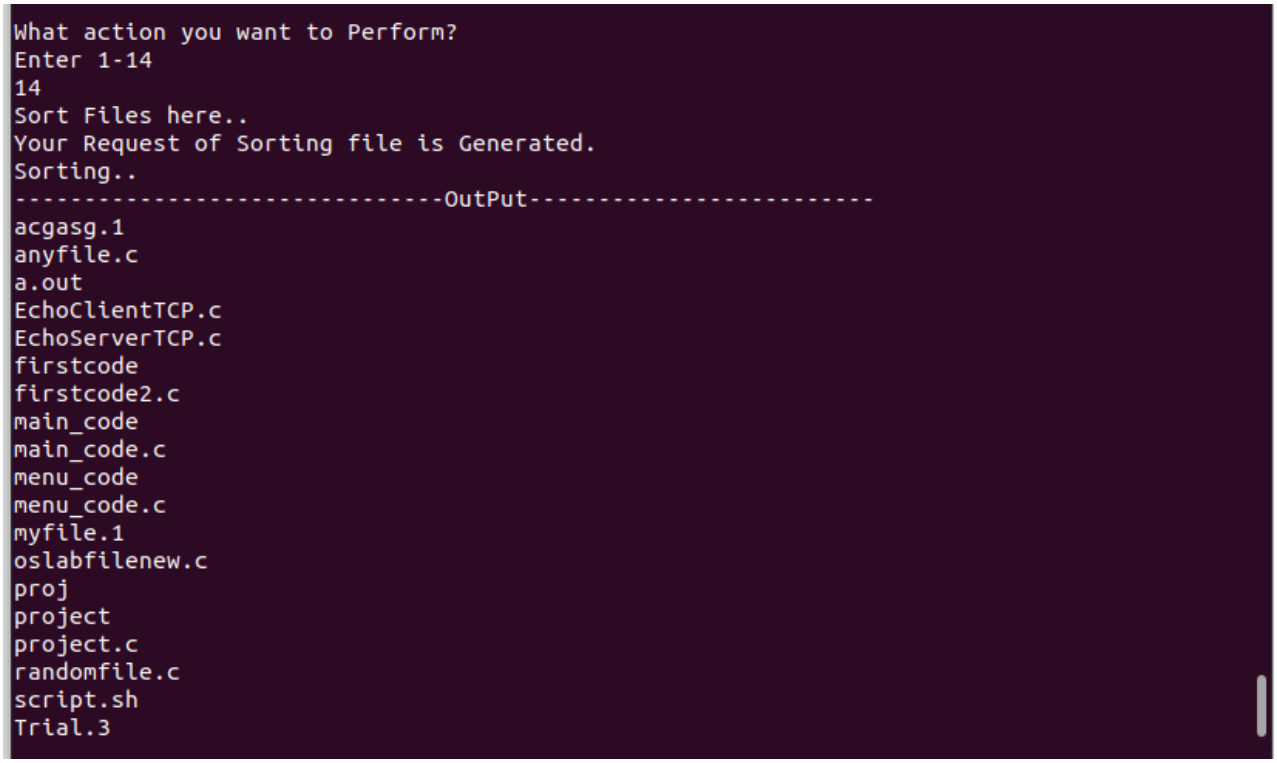
FUNCTION 12: Find total number of directories

```
What action you want to Perform?
Enter 1-14
12
Total number of Directories here..
Loading all directories..
Counting..
------------------------------OutPut-------------------------
Number of Directories are :
1
```

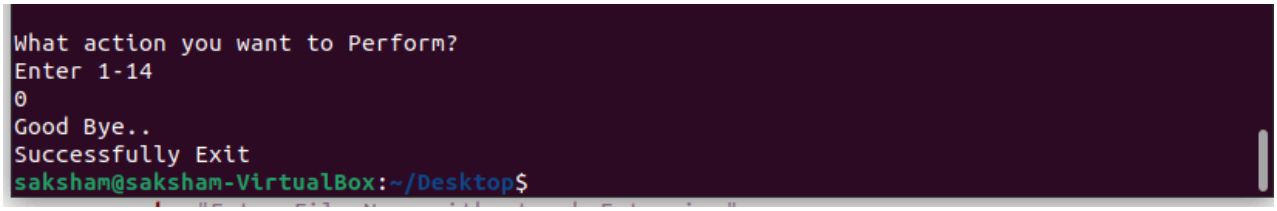
FUNCTION 13: Find total number of files in a directory

```
What action you want to Perform?
Enter 1-14
13
Total Numbers of Files in Current Directory here..
Loading all files..
------------------------------OutPut-------------------------
Number of Files are :
19
```

FUNCTION 14: Sort all files



FUNCTION 0: Exit



Throughout the project lifecycle, we will perform continuous analysis to measure the effectiveness and efficiency of the File and Directory Administration System. This analysis will involve evaluating user feedback, conducting usability tests, monitoring system performance, and comparing this administration system against predefined success metrics. The analysis phase will help identify any gaps or shortcomings and provide insights for further enhancements.

FUTURE ROLES

Following are the roles performed by file management system:

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

This is the most basic version of the File & Directory Administration System. So, in the 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 14 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 of file management in the Linux operating system. 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, deleting existing files, renaming files, editing files, reading or writing files, and so on. All the functionalities work on the basis of the user's input from the keyboard. Users can perform different basic functions 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 a basic knowledge of computers 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- inoperatingsystem.aspx

How to rename a file answer by MazharMIK 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 answered by:
https://linuxhint.com/linux_stat_command/

THANK YOU!