



Malaviya National Institute of Technology Jaipur

Course : OSS Lab

PROJECT REPORT

File Management and Password Cracking System

SUBMITTED TO:

Richa Kumari

SUBMITTED BY:

Naresh Kumar (2021UCP1205)

Suhani Gupta (2021UCP1149)

INTRODUCTION:

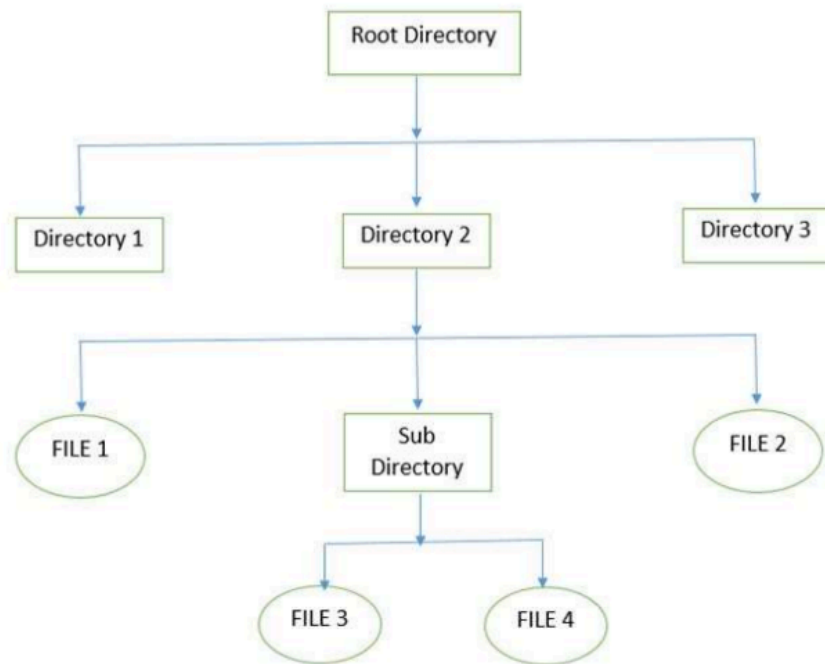
Computer users spend time every day interacting with digital files and folders, including creating, downloading, naming, moving, saving, copying, reviewing, navigating, searching for, sharing, and deleting them. This activity, called file management (FM). File management is an art of storing, naming,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

Provide insights into password cracking methodologies for security assessment.

PROJECT PARADIGM:

The file is actually the collection of associated information. This file-system prearranged into directory for efficient usage. Every directory has a number of files and other directories. The directory is defined as a bit which distinguish 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 manage the storage space and operation of files on media like disks.



MECHANISM AND WORKING:

```
#include <stdio.h>
```

```
int main(void) {
```

```
printf("\033[1m");
```

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

```
printf("..Welcome to File Management and Password Cracking System ..... \n");
```

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

```
printf("\033[0m");
```

```

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("15- Encrypt Password and Save to File\n");
printf("16- Crack Password from File\n");
printf("0- Exit\n");
printf("\nWhat action do you want to perform? Enter 1-16:\n");

return 0;
}

```

Main Code:

```
#!/bin/bash
```

```

encrypt_password() {
    local password="$1"
    local salt="$2"

```

```
    openssl passwd -6 -salt "$salt" "$password"
}
```

```
crack_password() {
    local target_encrypted_password="$1"
    local salt="$2"

    for a in {a..z}; do
        for b in {a..z}; do
            for c in {a..z}; do
                for d in {a..z}; do
                    local password="$a$b$c$d"
                    local generated_encrypted_password=$(openssl passwd
-6 -salt "$salt" "$password")

                    if [ "$target_encrypted_password" =
"$generated_encrypted_password" ]; then
                        echo "Password found: $password"
                        return 0
                    fi
                done
            done
        done
    done

    echo "Password not found"
    return 1
}
```

```
encrypt_and_save_password() {
    clear
```

```
echo "Enter password to encrypt: "  
read -s password  
echo "Enter salt: "  
read salt  
  
encrypted_password=$(encrypt_password "$password" "$salt")  
  
echo "Enter filename to save encrypted password: "  
read filename  
  
echo "$encrypted_password" > "$filename"  
  
echo "Password encrypted and saved to file: $filename"  
echo "Press Enter to return to the main menu..."  
read  
}  
  
# Function to crack password from file  
crack_password_from_file() {  
    clear  
    echo "Enter filename containing encrypted password: "  
    read filename  
  
    if [ ! -f "$filename" ]; then  
        echo "File not found: $filename"  
        echo "Press Enter to return to the main menu..."  
        read  
        return  
    fi  
  
    echo "Enter salt: "
```

```

read salt

target_encrypted_password=$(cat "$filename")

crack_password "$target_encrypted_password" "$salt"
echo "Press Enter to return to the main menu..."
read
}

i="0"

while [ $i -lt 100 ]
do
    gcc project.c -o proj
    ./proj
    read -p "Enter your choice: " opt1

    case $opt1 in
        1)
            echo "Listing all files and directories..."
            sleep 3
            echo "Loading..."
            sleep 3
            echo "-----Output-----"
            ls
            echo " "
            ;;
        2)
            echo "Creating new files..."
            echo "Which type of file do you want to create?"
            echo "1- .c"
            echo "2- .sh"
    esac
done

```

```

echo "3- .txt"
read -p "Enter your choice from 1-3: " filechoice

case $filechoice in
    1)
        read -p "Enter file name without .c extension: " filename
        touch "$filename.c"
        echo "-----Output-----"
        echo "File created successfully"
        echo " "
        ;;
    2)
        read -p "Enter file name without .sh extension: "
filename2
        touch "$filename2.sh"
        echo "-----Output-----"
        echo "File created successfully"
        echo " "
        ;;
    3)
        read -p "Enter file name without .txt extension: "
filename3
        touch "$filename3.txt"
        echo "-----Output-----"
        echo "File created successfully"
        echo " "
        ;;
    *)
        echo "Invalid input.. Try again."
        echo " "
        ;;
esac

```



```
::
```

3)

```
echo "Deleting existing files..."
read -p "Enter name of file you want to delete (please enter full
name with extension): " delfile
echo "-----Output-----"
if [ -f "$delfile" ]; then
    rm "$delfile"
    echo "Successfully deleted."
    echo " "
else
    echo "File does not exist. Try again."
    echo " "
fi
::
```

4)

```
echo "Renaming files..."
read -p "Enter old file name with extension: " old
if [ -f "$old" ]; then
    read -p "Enter new file name with extension: " new
    mv "$old" "$new"
    echo "-----Output-----"
    echo "File successfully renamed to $new"
    echo " "
else
    echo "$old does not exist. Try again with correct filename."
    echo " "
fi
::
```

5)

```
echo "Editing file content..."
read -p "Enter file name with extension: " edit
```

```

echo "-----Output-----"
if [ -f "$edit" ]; then
    nano "$edit"
    echo " "
else
    echo "$edit does not exist. Try again."
    echo " "
fi
;;

```

6)

```

echo "Searching for files..."
read -p "Enter file name with extension to search: " 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
;;

```

7)

```

echo "Displaying file details..."
read -p "Enter file name with extension to see details: " detail
echo "-----Output-----"
if [ -f "$detail" ]; then
    echo "File properties:"
    stat "$detail"
    echo " "
else

```

```

        echo "$detail does not exist. Try again."
        echo " "
    fi
;;
8)
    echo "Viewing file content..."
    read -p "Enter file name: " readfile
    echo "-----Output-----"
    if [ -f "$readfile" ]; then
        echo "File content:"
        cat "$readfile"
        echo " "
    else
        echo "$readfile does not exist."
        echo " "
    fi
;;
9)
    echo "Sorting file content..."
    read -p "Enter file name with extension to sort: " sortfile
    echo "-----Output-----"
    if [ -f "$sortfile" ]; then
        echo "Sorted file content:"
        sort "$sortfile"
        echo " "
    else
        echo "$sortfile does not exist. Try again."
        echo " "
    fi
;;
10)
    echo "Listing all directories..."

```

```
echo "-----Output-----"
echo "Showing all directories..."
echo "Loading..."
sleep 3
ls -d */
echo " "
;;
```

11)

```
echo "Listing files with particular extensions..."
echo "Which type of file list do you want to see?"
echo "1- .c"
echo "2- .sh"
echo "3- .txt"
read -p "Enter your choice from 1-3: " extopt
echo "-----Output-----"
case $extopt in
  1)
    echo "List of .c files:"
    echo "Loading..."
    sleep 3
    ls *.c
    echo " "
    ;;
  2)
    echo "List of .sh files:"
    echo "Loading..."
    sleep 3
    ls *.sh
    echo " "
    ;;
  3)
    echo "List of .txt files:"
```

```

        echo "Loading..."
        sleep 3
        ls *.txt
        echo " "
        ;;
    *)
        echo "Invalid input.. Try again."
        echo " "
        ;;
esac
;;
12)
    echo "Counting total number of directories..."
    echo "Loading all directories..."
    echo "Counting..."
    sleep 3
    echo "-----Output-----"
    echo "Number of directories:"
    echo "$ (ls -d */ | wc -w)"
    echo " "
    ;;
13)
    echo "Counting total number of files in current directory..."
    echo "Loading all files..."
    sleep 3
    echo "-----Output-----"
    echo "Number of files:"
    echo "$ (ls -l | grep -v 'total' | grep -v '^d' | wc -l)"
    echo " "
    ;;
14)
    echo "Sorting files..."

```

```

    echo "Your request for sorting files is being processed..."
    echo "Sorting..."
    sleep 3
    echo "-----Output-----"
    ls | sort
    echo " "
    ;;
15)
    encrypt_and_save_password
    ;;
16)
    crack_password_from_file
    ;;

0)
    echo "Goodbye.."
    echo "Successfully exited."
    break
    ;;
*)
    echo "Invalid input.. Try again."
    ;;
esac

i=$((i+1))

```

done

Output Screenshots:

Main Menu:

```
naresh@naresh-virtual-machine:~$ bash password.sh
=====
-----Welcome to File Management and Password Cracking System-----
=====
Welcome, The Main Menu is given below:
1- List all Files and Directories
2- Create New Files
3- Delete Existing Files
4- Rename Files
5- Edit File Content
6- Search Files
7- Details of Particular File
8- View Content of File
9- Sort File Content
10- List only Directories(Folders)
11- List Files of Particular Extension
12- Count Number of Directories
13- Count Number of Files
14- Sort Files in a Directory
15- Encrypt Password and Save to File
16- Crack Password from File
0- Exit

What action do you want to perform? Enter 1-16:
Enter your choice: █
```

Choice 01 Output:

```
naresh@naresh-virtual-machine:~$ bash password.sh
=====
-----Welcome to File Management and Password Cracking System-----
=====
Welcome, The Main Menu is given below:
1- List all Files and Directories
2- Create New Files
3- Delete Existing Files
4- Rename Files
5- Edit File Content
6- Search Files
7- Details of Particular File
8- View Content of File
9- Sort File Content
10- List only Directories(Folders)
11- List Files of Particular Extension
12- Count Number of Directories
13- Count Number of Files
14- Sort Files in a Directory
15- Encrypt Password and Save to File
16- Crack Password from File
0- Exit

What action do you want to perform? Enter 1-16:
Enter your choice: 1
Listing all files and directories...
Loading...
-----Output-----
Desktop    Downloads  Music    pass1.txt  pass3.txt  pass.txt  Pictures  project.c  Public  sniper  Videos
Documents  helloo.sh  nkp.sh   pass2.txt  pass4.txt  password.sh  proj     project.c.save  snap  Templates
```


Choice 02 Output:

```
=====
-----Welcome to File Management and Password Cracking System-----
=====
Welcome, The Main Menu is given below:
1- List all Files and Directories
2- Create New Files
3- Delete Existing Files
4- Rename Files
5- Edit File Content
6- Search Files
7- Details of Particular File
8- View Content of File
9- Sort File Content
10- List only Directories(Folders)
11- List Files of Particular Extension
12- Count Number of Directories
13- Count Number of Files
14- Sort Files in a Directory
15- Encrypt Password and Save to File
16- Crack Password from File
0- Exit

What action do you want to perform? Enter 1-16:
Enter your choice: 2
Creating new files...
Which type of file do you want to create?
1- .c
2- .sh
3- .txt
Enter your choice from 1-3: 2
Enter file name without .sh extension: neww
-----Output-----
File created successfully
```

Choice 15 Output:

```
Enter password to encrypt:
aaaa
Enter salt:
abc
Enter filename to save encrypted password:
pass5.txt
Password encrypted and saved to file: pass5.txt
Press Enter to return to the main menu...

=====
-----Welcome to File Management and Password Cracking System-----
=====
Welcome, The Main Menu is given below:
1- List all Files and Directories
2- Create New Files
3- Delete Existing Files
4- Rename Files
5- Edit File Content
6- Search Files
7- Details of Particular File
8- View Content of File
9- Sort File Content
10- List only Directories(Folders)
11- List Files of Particular Extension
12- Count Number of Directories
13- Count Number of Files
14- Sort Files in a Directory
15- Encrypt Password and Save to File
16- Crack Password from File
0- Exit
```

Choice 16 Output:

```
Enter filename containing encrypted password:
pass5.txt
Enter salt:
abc
Password found: aaaa
Press Enter to return to the main menu...
```

Learning Outcomes:

Through the File Management and Password Cracking System, users can expect to achieve the following learning outcomes:

Understanding of file management techniques and their practical applications.

Familiarity with password encryption methodologies and their importance in data security.

Awareness of password cracking techniques and the need for strong password practices.

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 and Password Cracking System to perform different functions on files.

—THANK YOU—