



Project Title: Bank Fraud Detection Using Machine Learning

Course Title: Operating System Lab

Course Code: CSE324

Daffodil International University

Submitted To-

**Ms. Subhenur Latif
Assistant Professor
Department of CSE
Daffodil International University**

Submitted By-

Md. Nahid Hasan Choudhury

212-15-4125

Sadya Afrin Labonno

212-15-4147

Sec: 59_A

Department of CSE

Daffodil International University

Submission Date: 27-11-2023

File System Management

Introduction:

In this today's modern world, all are our electronic devices like mobile, computer and other devices are running on several operating systems. In this digital age, effective file system management is crucial for maintaining an organized and efficient workflow. Our project aims to develop a simple yet powerful File System Management Tool using Bash scripting. It helps to develop a robust file system management tool using Bash scripting, offering users a versatile solution for interacting with files and directories in a Linux environment. This tool provides users with a range of functionalities to interact with files and directories seamlessly. It also provides an accessible interface for both beginners and experienced users, bridging the gap between simplicity and functionality. It offers various options for listing, creating, deleting, renaming, editing, searching, and sorting files and directories. This project provides an accessible interface for both beginners and experienced users, bridging the gap between simplicity and functionality.

Objective:

The primary objective of this project is to create a user-friendly and efficient interface for managing files and directories through the command line. Also another objective of this project is to develop a script that simplifies file and directory management tasks for users, allowing them to easily navigate and interact with files and directories in a Linux environment. By leveraging Bash scripting, we aim to empower users with various functionalities, simplifying complex file system operations. The tool's design is centered on providing an intuitive experience, especially for those less familiar with the intricacies of the Linux command line. It will specially help those new to the command line, with a straightforward tool for effective file system navigation and manipulation. The objectives include:

- User friendly Interface
- Accessibility for Beginners
- Streamlined Navigation
- Increased Control and Understanding
- Time Efficiency
- User Productivity
- Creative File Operations

These objectives collectively contribute to the overarching aim of creating a File System Management Tool that is accessible, versatile, and empowering for users in their file management endeavors within a Linux environment.

Motivation:

The motivation behind this project is to enhance the user experience when working with files and directories in a Linux environment. By creating a user-friendly interface, users can perform various tasks more efficiently and with less confusion. It arises from the recognition of the challenges users face when dealing with file management in a command-line environment. Many individuals, particularly beginners, find the command line interface intimidating. This project seeks to ease that intimidation by offering a menu-driven tool that captures powerful file management capabilities. The motivation is rooted in making file system navigation and manipulation more accessible and user-friendly. Also the need for a lightweight and accessible file management solution within the command line interface. This tool aims to bridge that gap by providing a menu-driven interface, making file management more natural and less intimidating.

Working Methodology:

The File System Management Tool is implemented in Bash scripting, a widely-used shell scripting language. The script utilizes a while loop to create an interactive menu, allowing users to choose from a variety of file-related operations. Each menu option corresponds to a specific block of code, facilitating the execution of the chosen operation.

- The project initiates a 'while' loop to create an interactive menu-driven interface. The loop continues as long as the counter variable 'i' is less than 100, allowing users to perform multiple file operations in a single session.

```
#!/bin/bash
i="0"
while [ $i -lt 100 ]
do
    echo "=====
    echo "-----FILE SYSTEM MANAGEMENT-----
    echo "=====
    echo " "
    echo "1- List all files and Directories here"
    echo "2- Create New Files"
```

- Users are prompted to choose an option from the menu by entering a numerical value. For example: The 'read opt1' command captures the user's input, and subsequent 'if-elif' statements evaluate the chosen option. Here all the options from menu:
 1. Listing Files and Directories: Utilizes the 'ls' command to display all files and directories.
 2. Creating Files: Offers the creation of .c, .sh, or .txt files using the 'touch' command.
 3. Deleting Files: Implements the 'rm' command to delete specified files chosen by user.
 4. Renaming Files: Employs the 'mv' command for file renaming.
 5. Editing File Content: Uses the 'nano' text editor for modifying file content.
 6. Searching for Files: Utilizes the find command for searching files in specified directories or files as per users search.
 7. File Details: Displays file details using the 'stat' command.
 8. Viewing File Content: Shows file content with the 'cat' command.
 9. Sorting File Content: Sorts file content using the 'sort' command.
 10. Listing Directories: Lists only directories with the 'ls -d */' command.
 11. Listing Files by Extension: Lists files based on the chosen extension (.c, .sh, .txt).
 12. Counting Directories and Files: Counts the number of directories and files using 'echo */ | wc -w and ls -l | grep -v 'total' | grep -v '^d' | wc -l' respectively.
 13. Sorting Files in a Directory: Sorts files in the current directory using 'ls | sort'.
- Each menu option corresponds to a specific file operation, encapsulated within 'if-elif' blocks. These options range from listing files and directories to creating, deleting, renaming, and editing files.
- The project includes common Linux commands such as 'ls', 'touch', 'rm', 'mv', 'nano', 'find', 'stat', 'cat', 'sort', and more to perform these file operations.

```

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

```

- When creating new files, users are prompted to choose the type of file they want to create (.c, .sh, or .txt) and enter a filename. The 'touch' command is then used to dynamically create the specified file with the appropriate extension.

```

echo "Enter File Name without .c Extension"
read filename
touch $filename.c
echo "-----OutPut-----"
echo "File Created Successfully"
echo " "

```

- It includes error handling tools to deal with various scenarios, such as non-existent files during deletion or renaming. For example, the script checks for the existence of a file before attempting to delete or rename it.
- Users receive informative messages, such as "File Does not Exist" or "Successfully Deleted," providing feedback on the outcome of their actions.

```

if [ -f "$delfile" ];
then
    rm $delfile
    echo "Successfully Deleted."
    echo " "
else
    echo "File Does not Exist..Try again"
    echo " "
fi

```

- There are echo statements that provide informative messages to the user, enhancing the overall user experience. For example, when viewing file content or searching for files, the script displays the ongoing process to the user.

- The loop counter 'i' is incremented at the end of each iteration (i=\$((i+1))). This ensures that the script runs until the counter reaches a predefined limit (100 in this case), allowing users to perform multiple file operations in a single session.

```
fi
i=$((i+1))
done
```

- The project also includes a sleep command to add a delay (3 Seconds) between menu options, creating a more responsive user experience. It is particularly useful when you want to control the timing of operations within a project or code.

```
echo "-----"
echo "Total number of Directories"
echo "Loading all directories.."
sleep 3
echo "Counting.."
sleep 3
echo "Number of Directories are :"
echo */ | wc -w
```

- In our code, the dollar sign (\$) is primarily used for variable substitution. For example, in our Bash script, '\$opt1' is used for variable substitution. The variable 'opt1' is assigned the value entered by the user through the read command. The 'read' command is used to take user input and store it in the variable opt1. The '\$opt1' syntax is then used to substitute the value of the variable opt1 in various conditional statements.

```
read opt1
if [ $opt1 == 1 ]
then
```

File System Management Tool to provide users with a seamless, interactive, and dynamic file management experience within a Linux environment. The use of a menu-driven approach and the incorporation of essential Linux commands make the tool accessible to users with varying levels of expertise.

Outcome:

The File System Management Tool successfully provides users with an interactive and efficient means of managing files and directories. Its menu-driven approach simplifies complex command-line operations, making it accessible to users with varying levels of expertise. The tool enhances user productivity and helps in fostering a better understanding of file system navigation. File System Management Tool encompasses the benefits and achievements resulting from its design and implementation.

Code Snippets:

```
ironman784@DESKTOP-4HA3E3P MINGW64 /f/Versity/Fall 23/OS/project
$ bash sys.sh
-----FILE SYSTEM MANAGEMENT-----
1- List all files and Directories here
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
0- Exit
```

It is the first output of our file system management where all option of file system will show. Here user can choose between 1 to 14 for different functions to work. At last, pressing '0' will exit the system management.

```
1
List all files and Directories here..
Showing all files and directories...
Loading..
-----Output-----
'Project Report OS_4125_4147_59-A.docx'  nahid.txt  sys.sh
```

```
2
Create New Files here..echo 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
nahid
-----Output-----
File Created Successfully
```

Here we can see, we can display and create different format of files using this management system.

```
4
-----Output-----
Rename files here..
Enter Old Name of File with Extension..
nhd.c
Checking for file...
Ok File Exist.
Now Enter New Name for file with Extension
nahidd
Successfully Rename.
Now Your File Exist with nahidd Name
```

```
14
-----Output-----
Sort Files here..
Your Request of Sorting file is Generated.
Sorting..
Project Report OS_4125_4147_59-A.docx
nahid.txt
nahidd
sys.sh
system project code.pdf
tuli.c
-Project Report OS_4125_4147_59-A.docx
```

```
5
Edit file content here..
Enter File Name with Extension :
nahid.txt
-----Output-----
checking for file..
opening file..
```

```
MINGW64:/f/Versity/Fall 23/OS/project
GNU nano 7.2
hello there how are you
hello world !
```

```
12
-----Output-----
Total number of Directories here..
Loading all directories..
Counting..
Number of Directories are :
1

13
-----Output-----
Total Numbers of Files in Current Directory here..
Loading all files..
Number of Files are :
7
```

Conclusion:

Thus, this project underscores the effectiveness of using Bash scripting to create a practical and user-centric file management tool. The project's comprehensive feature set, coupled with its accessible design, positions it as a valuable asset for users working in Linux environments. Future iterations may include additional functionalities and an improved user interface, further contributing to the tool's utility and overall user experience.

This project embodies our commitment to simplicity, accessibility, and efficiency in file system management through the command line, addressing the diverse needs of users in the digital age. By implementing a user-friendly interface, users can easily navigate and interact with files and directories, improving their overall experience when working in a Linux environment. The tool is envisioned as a valuable asset for users seeking a user-friendly, educational, and versatile solution for their file management needs.