FILE MANAGEMENT SYSTEM COSC 439: Operating Systems

Karne

Spring 2022

Group Members:

1.Boaz Alemseged
2.Monssef Marrakchi
3.Gideon Giorgis
4.Danayt Teklu
5. Jalen Graham

Project Goal

The goal of this project was to get used to how file management works and how it works in an Operating System. The File management system is the only part of the operating system that has the logical structure of it visible to users and the physical structure hidden from the users. This system communicates with the file management system for accessing, storing, editing, and deleting files.

What is File Management?

Computers use file management systems to store information in an easily accessible environment, allowing for the creation, reading, and writing to files, a file being a storage unit. Files are mapped by the operating system onto physical drives, which are usually nonvolatile, so that the information persists between reboots. Files typically consist of programs or data

File Operations

1. Create - creates a new file in the system

Createf <filename>

2. Write - Either appends (adds) to the file or overwrites it depending on the command

Write append/overwrite <filename>

3. Read - using the cat command, you can read a file even after it has been edited

Cat <filename>

4. Delete - using the delf command you can simply delete any file in the system

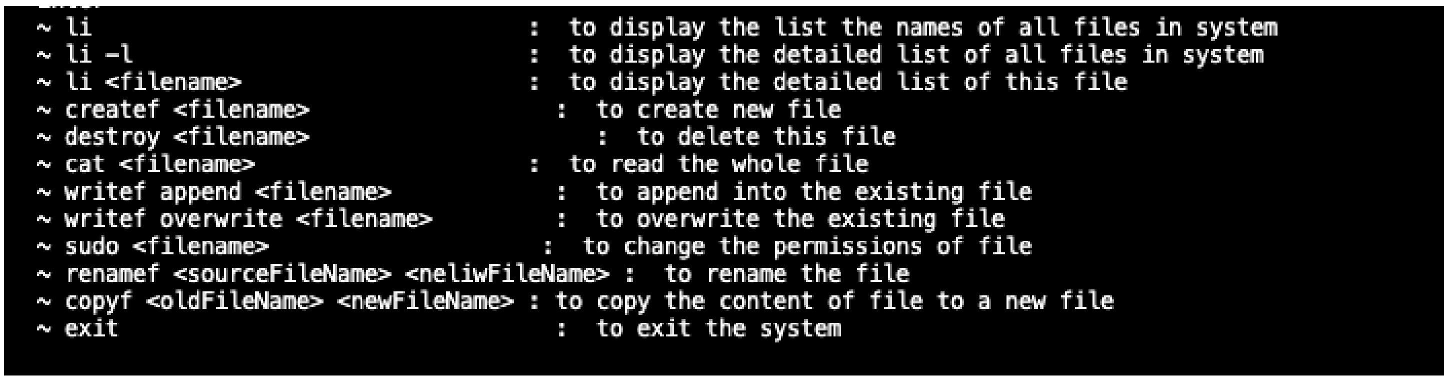
Delf <filename>

5. Display - files may be displayed as a whole or as a detailed list depending on the file

Li, li –l for detailed list

6. Copy - the content of one file may be copied to another file

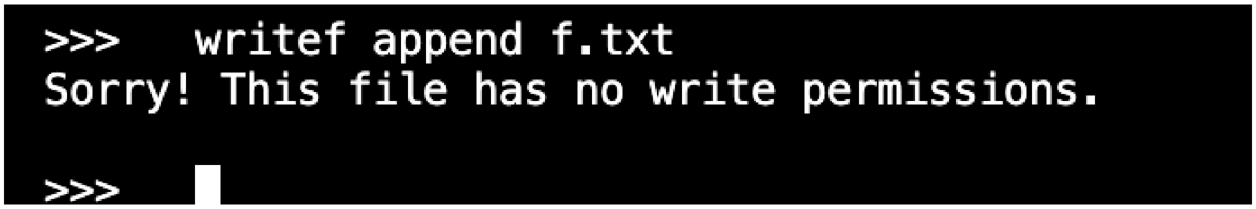
Copyf <oldfilename> <newfilename>



```
~ li                             :  to display the list the names of all files in system
~ li -l                          :  to display the detailed list of all files in system
~ li <filename>                  :  to display the detailed list of this file
~ createf <filename>               :  to create new file
~ destroy <filename>                 :  to delete this file
~ cat <filename>                 :  to read the whole file
~ writef append <filename>        :  to append into the existing file
~ writef overwrite <filename>     :  to overwrite the existing file
~ sudo <filename>                :  to change the permissions of file
~ renamef <sourceFileName> <neliwFileName> :  to rename the file
~ copyf <oldFileName> <newFileName> : to copy the content of file to a new file
~ exit                            :  to exit the system
```

Error Handling

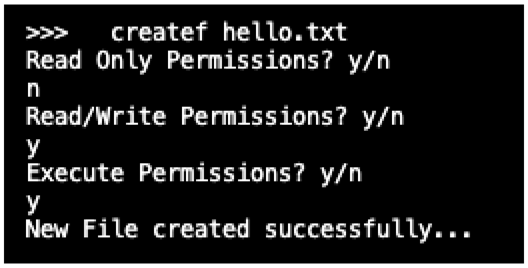
- Since misspelled words & file same file names are common.
- Error handling is found when files that are set to read only are trying to be written to.



```
>>>   writef append f.txt
Sorry! This file has no write permissions.

>>>
```

File Management Simulation Operations

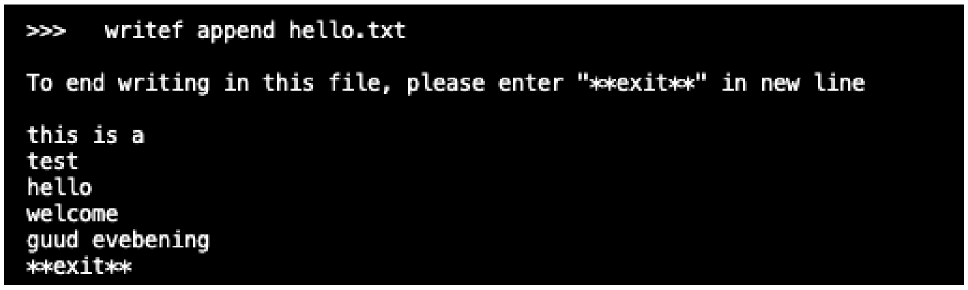
CREATE



```
>>>   createf hello.txt
Read Only Permissions? y/n
n
Read/Write Permissions? y/n
y
Execute Permissions? y/n
y
New File created successfully...
```

When you create a file using the command *createf <filename>*, it asks if you want to do Read Only and Read/Write then execute those permissions. Read Only permissions decides if the file manager lets you read the file. Read/Write permissions means you can read and write into each file. Execute determines if it puts those permissions onto that file.

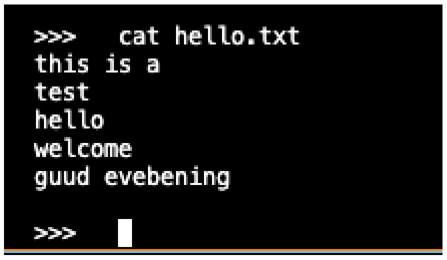
WRITE



```
>>>   writef append hello.txt

To end writing in this file, please enter "**exit**" in new line

this is a
test
hello
welcome
guud evebening
**exit**
```

By using the command *writef append <filename>*, you can write into files that have read and write permissions. You then put **exit to stop writing into files.**

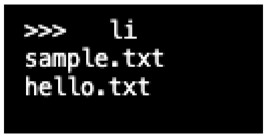
CAT



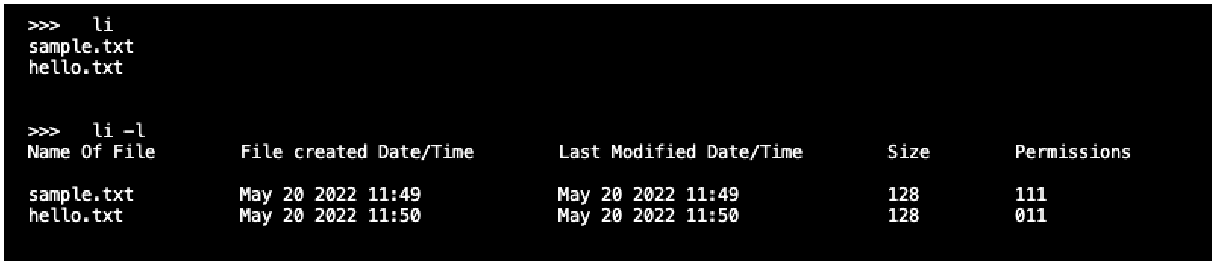
```
>>>   cat hello.txt
this is a
test
hello
welcome
guud evebening

>>>   
```

The *cat* command displays what is inside the hello.txt file and shows what was appended in the previous step.

LIST and LIST ALL



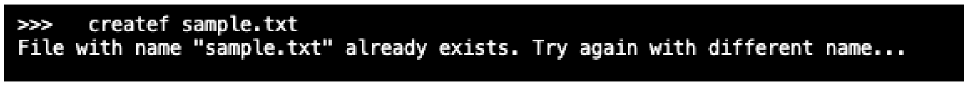
```
>>>   li
sample.txt
hello.txt
```



```
>>>   li
sample.txt
hello.txt


>>>   li -l
Name Of File        File created Date/Time        Last Modified Date/Time        Size        Permissions

sample.txt          May 20 2022 11:49             May 20 2022 11:49              128         111
hello.txt           May 20 2022 11:50             May 20 2022 11:50              128         011
```

The command *li* displays the names of the files. The command li -l displays the names, file creation date and time, last modified date/time, size, and permissions of all files in the system, and currently there are two text files, sample and hello.

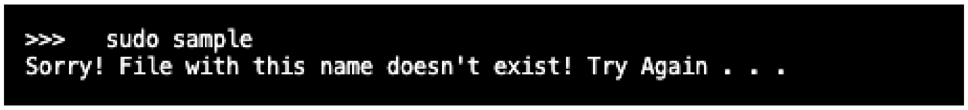
ERROR HANDLING



```
>>>   createf sample.txt
File with name "sample.txt" already exists. Try again with different name...
```

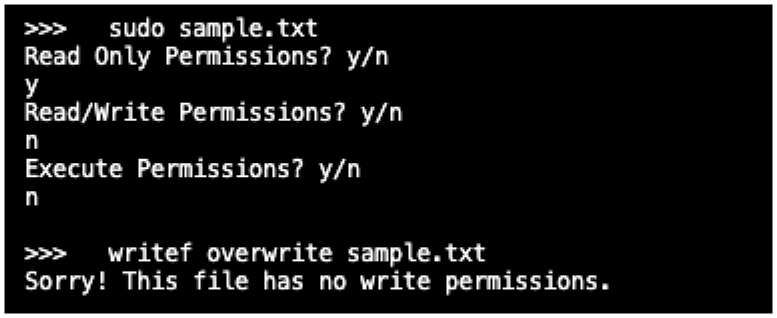
If you try to create a new sample.txt, the error handling will stop it and it will give you an error saying the file already exists because the sample.txt file is already in the system.

SUDO



```
>>>   sudo sample
Sorry! File with this name doesn't exist! Try Again . . .
```

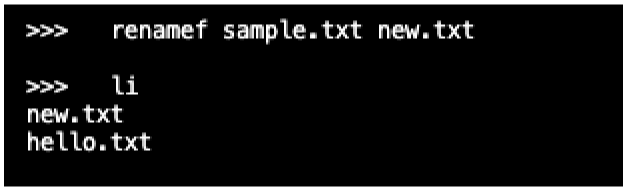
***Sudo* sample does not work because it is not specified as a .txt file and the only files in the system are sample.txt and hello.txt**



```
>>>   sudo sample.txt
Read Only Permissions? y/n
y
Read/Write Permissions? y/n
n
Execute Permissions? y/n
n

>>>   writef overwrite sample.txt
Sorry! This file has no write permissions.
```

***Sudo* sample.txt works fine! However, overwriting the sample.txt file does not work because no was put for the read/write permissions**

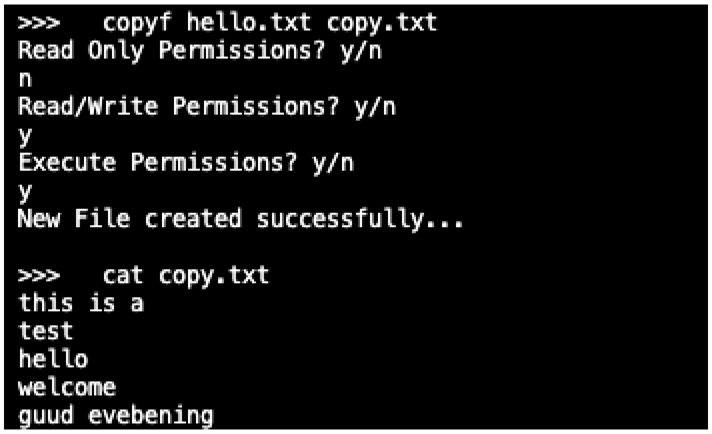
RENAME



```
>>>   renamef sample.txt new.txt

>>>   li
new.txt
hello.txt
```

The first picture shows sample.txt and hello.txt are the files in the system. After renaming sample.txt to new.txt with the *renamef* command, the Li command shows new.txt

<u>COPY</u>



```
>>>   copyf hello.txt copy.txt
Read Only Permissions? y/n
n
Read/Write Permissions? y/n
y
Execute Permissions? y/n
y
New File created successfully...

>>>   cat copy.txt
this is a
test
hello
welcome
guud evebening
```

Hello.txt is copied to copy.txt using the *copyf* command. To show that it works, we used the cat command to display what is in the copy.txt file, and it displays what we appended to the hello.txt file earlier