

ESOF 3655 - Principles Operating System
Fall 2020

Portable File Management System

Contributing Members -
Umang Shah and Khanjan Dabhi

Date submitted – 12/10/2020

Approval

This document has been read and approved by the following team members responsible for its implementation

PRINT NAME: Umang Shah

Signature: US

PRINT NAME: Khanjan Dabhi

Signature: KD

Revisions

INTRODUCTION

Initially, we will clarify that how Portable File Management System is running and how client will utilize this framework. Compact File System runs various capacities. For instance, making a document, erasing, rename a record and duplicate a record. Then again, this program does some various exhibitions on the content. The client can change what client needs, for example, annex text end of the document, embed the content, eliminate the content and show the content. Consequently, when any client utilize that program, client can transform it while utilizing simultaneously. Convenient File System has other element that this program can run all working frameworks, for example, Windows, Linux and IOS. Compact File Management System doesn't give a blunder in all frameworks.

This project was made following Agile software development methodology. We first created, "Create", "Delete" Functions to check an initial menu and other functions were added gradually later on.

This program begins to run with principle screen. This screen shows the order with short code:

E: Exit

C: Create

D: Delete

R: Rename

O: Copy

M: Move

A: Append text

T: Delete text

S: Show text

At the point when client runs the program, the client will see above screen. From that point onward, client will pick the letter and consequently the order. Along these lines, one of orders is run.

MENU SCREEN

This menu tells the client the best way to keep on utilizing that program. The fundamental menu window incorporates all orders. On the off chance that client composes wrong short code, the program will ask an order once more. If it is right, it will pose new inquiry about that order. Nonetheless, while running project memory the board works consummately. We utilized all memory decisions. On the off chance that client sees the blunder, client can utilize 'E' letter and the program gets shut.

```
C:\Users\khanj\source\repos\Operating System\Debug\Operating System.exe
Select a Input:
E : Exit
C : make
D : Delete
R : Rename
O : Duplicate
M : Move
A : Append Text
T : Delete Text
S : Show Text
Input >
```

MAKE FILE

The client makes a document while utilizing the order of making a record. The sort of record can be pdf, txt and so forth. In any case, the client should be compose the index while making a document. Then again, if client need to make a record with a similar name with a document on the PC, the program won't make a record with that name and it will say that this document as of now exist. In any case, there isn't a document that name, Portable file System will make a record and ask another order.

```
C:\Users\khanj\source\repos\Operating System\Debug\Operating System.exe
Select a Input:
E : Exit
C : make
D : Delete
R : Rename
O : Duplicate
M : Move
A : Append Text
T : Delete Text
S : Show Text
Input > C
Enter the name of file
filename > example.txt
The file has been made and the path is example.txt
Input > Input >
```

The function of Make file is running like above.

DELETE FILE

The client can erase the record with Portable File System. This program doesn't take a gander at the sort of the document while erasing. On the off chance that there is a document, program erases that record; then again, if there isn't a record, program gives a mistake.

RENAME A FILE

Portable File System allows renaming a file. While changing a filename, the type of file is not important for the program. Because, this function only does not change the name. It changes name and its directory at the same time.

This capacity begins to ask the order. After client pick the R, the program requests the name from old document. After the client composed the old document the program requests the name from new record. On the off chance that old record is exist, the capacity will work effectively. In any case, if there isn't exist, the capacity won't work.

DUPLICATE FILE

We explain the functions that the program includes. Copy a file is the one of them.

- `printf("Input the file name to be copied: ");`
- `scanf("%s", source);`
- `printf("Input the destination that file will be pasted: ");`
- `scanf("%s", target);`

While above codes, System will duplicate the record. To start with, it requests the name from the record, and it takes the source the name. From that point forward, it asks the objective that record will be glued, and it takes the objective.

At the point when we check the index of the document for the capacity of duplicate a record. The augmentation of the document will be equivalent to the expansion of the duplicate.

MOVE A FILE

The client can move a record unique spot to some place that the client needs. The capacity of move a document has an essential rationale. In the event that somethings move spot to another spot. It should be erased from unique spot and replicated new spot. We made the capacity as an association of the capacity of erase a record and duplicate a document.

This capacity works that first it requests the name from the document that will be moved and from that point onward, it requests the objective from the record that will be moved. This capacity really erases the record the primary spot and duplicate the new spot.

APPEND THE TEXT

This capacity gives to attach text toward the finish of the document. The client can attach something a record. In any case, the record should be existing before the client doesn't attach.

To begin with, program requests the record from name that affix text. Second, program will ask that what will compose toward the finish of the record. The client will add what he needs. From that point onward, post will be included the document. In the event that there is a mistake while running capacity, the capacity will return and alert. Then again it will proceed.

DELETE THE TEXT

This capacity gives to eliminate the substance of the document. Accordingly, there should be a document for eliminating. This capacity additionally works like an annex text. The program requests the document from eliminating, and after entering the name, the capacity will erase the substance of the record.

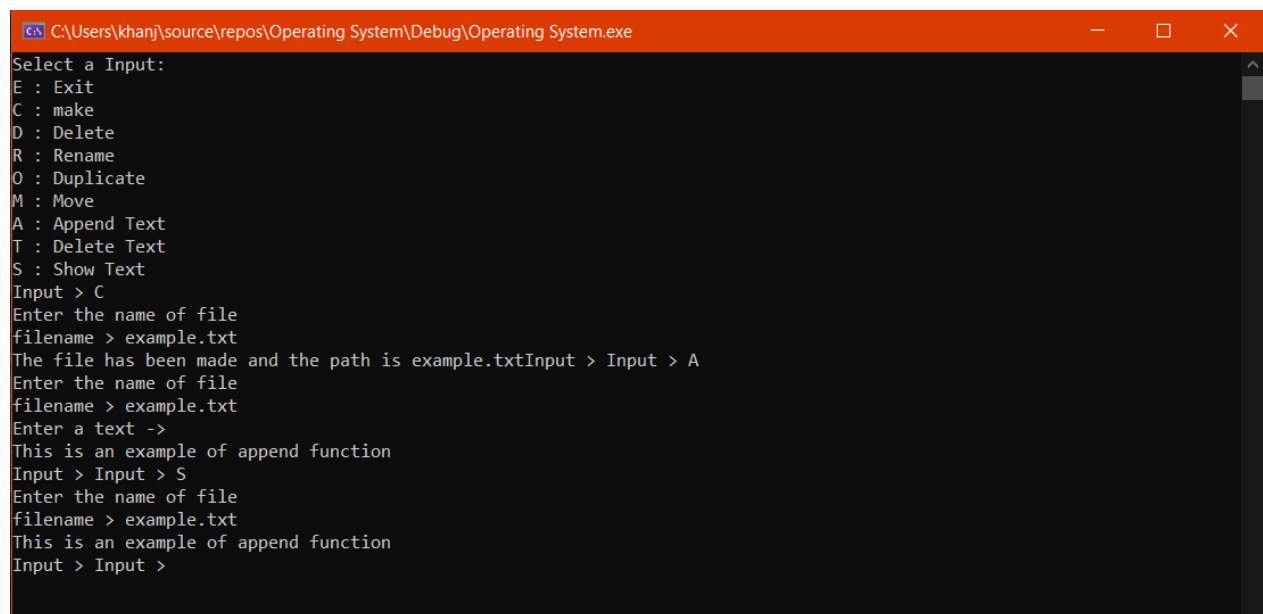
DISPLAY THE TEXT

The client can see the absolute content of the document with the capacity of show the content. This capacity is comparable with the elements of add text and other altering types. If client needs to see explicit line in the content, client can look in the content and after client can pass different lines for perusing.

This capacity asks the client the which document you would like to see, and after client enters the name of the record, client can see the entire content.

HELP

Help utility made for the clients. On the off chance that client can't fathom by what method can utilize the program, assist utility with canning the clients. Help utility incorporates the clarification for all capacities. We will clarify more proficiently in the client manual.



```
C:\Users\khanj\source\repos\Operating System\Debug\Operating System.exe
Select a Input:
E : Exit
C : make
D : Delete
R : Rename
O : Duplicate
M : Move
A : Append Text
T : Delete Text
S : Show Text
Input > C
Enter the name of file
filename > example.txt
The file has been made and the path is example.txtInput > Input > A
Enter the name of file
filename > example.txt
Enter a text ->
This is an example of append function
Input > Input > S
Enter the name of file
filename > example.txt
This is an example of append function
Input > Input >
```