

CSA0405 – OPERATING SYSTEM FOR FILE SYSTEM IMPLEMENTATION

CAPSTONE PROJECT

IMPLEMENTATION OF FILE SYSTEM MANAGEMENT

GOKULRAJ S (192224130)

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 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.

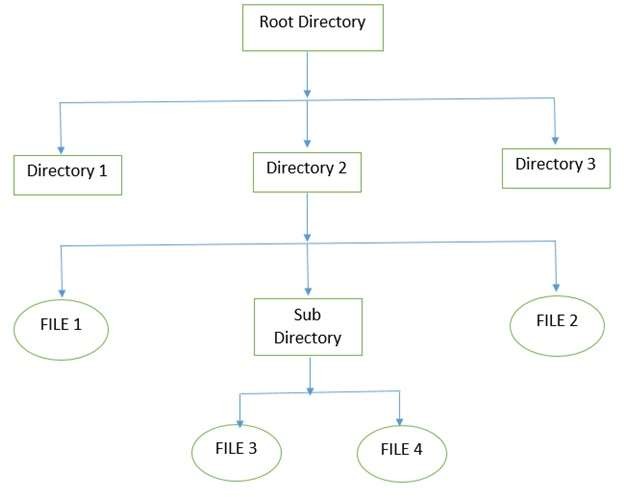
The following are some of the tasks performed by file management of operating system of any computer system:

1. It helps to create new files in computer system and placing them at the specific locations.
2. It helps in easily and quickly locating these files in computer system.
3. It helps to stores the files in separate folders known as directories. These directories help users to search file quickly or to manage the files according to their types or uses.
4. It helps the user to modify the data of files or to modify the name of the file in the directories etc.

File management helps users to organize their valuable documents in a systematic manner for better and efficient use of it.

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.



The above figure shows the general hierarchy of the 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 base storage system ensures better organization of files in the memory of the computer system.

PROGRAM:

#include <stdio.h>

#include <stdlib.h>

#define MAX\_FILENAME\_LENGTH 100

#define MAX\_FILE\_CONTENT\_LENGTH 1000

void createFile();

void deleteFile();

void readFile();

void writeFile();

void listFiles();

void exitProgram();

int main() {

int choice;

do {

printf("\nFile System Management\n");

printf("1. Create a file\n");

printf("2. Delete a file\n");

printf("3. Read a file\n");

printf("4. Write to a file\n");

printf("5. List files\n");

printf("6. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

createFile();

break;

case 2:

deleteFile();

break;

case 3:

readFile();

break;

case 4:

writeFile();

break;

case 5:

listFiles();

break;

case 6:

exitProgram();

break;

default:

printf("Invalid choice. Please enter a number between 1 and 6.\n");

}

} while (choice != 6);

return 0;

}

void createFile() {

char filename[MAX\_FILENAME\_LENGTH];

FILE \*file;

printf("Enter the filename: ");

scanf("%s", filename);

file = fopen(filename, "w");

if (file == NULL) {

printf("Failed to create the file.\n");

return;

}

fclose(file);

printf("File '%s' created successfully.\n", filename);

}

void deleteFile() {

char filename[MAX\_FILENAME\_LENGTH];

printf("Enter the filename to delete: ");

scanf("%s", filename);

if (remove(filename) == 0)

printf("File '%s' deleted successfully.\n", filename);

else

printf("Failed to delete the file.\n");

}

void readFile() {

char filename[MAX\_FILENAME\_LENGTH];

char content[MAX\_FILE\_CONTENT\_LENGTH];

FILE \*file;

int c;

printf("Enter the filename to read: ");

scanf("%s", filename);

file = fopen(filename, "r");

if (file == NULL) {

printf("Failed to open the file.\n");

return;

}

printf("File content:\n");

while ((c = getc(file)) != EOF)

putchar(c);

fclose(file);

}

void writeFile() {

char filename[MAX\_FILENAME\_LENGTH];

char content[MAX\_FILE\_CONTENT\_LENGTH];

FILE \*file;

printf("Enter the filename to write: ");

scanf("%s", filename);

printf("Enter the content: ");

scanf(" %[^\n]s", content);

file = fopen(filename, "w");

if (file == NULL) {

printf("Failed to open the file.\n");

return;

}

fprintf(file, "%s", content);

fclose(file);

printf("Content written to file '%s' successfully.\n", filename);

}

void listFiles() {

system("ls");

}

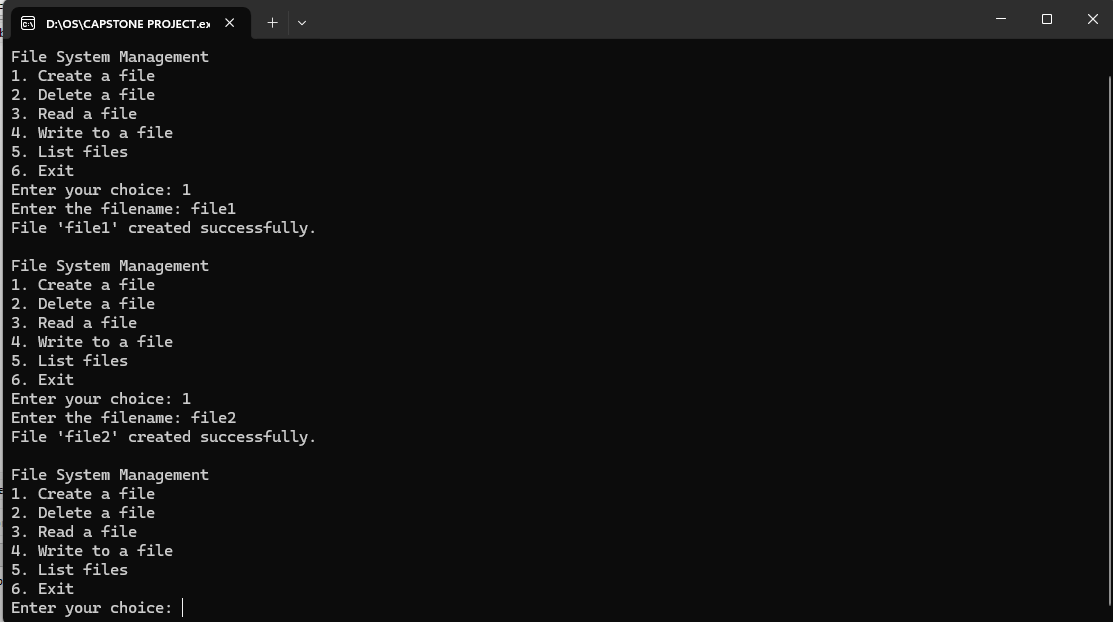
void exitProgram() {

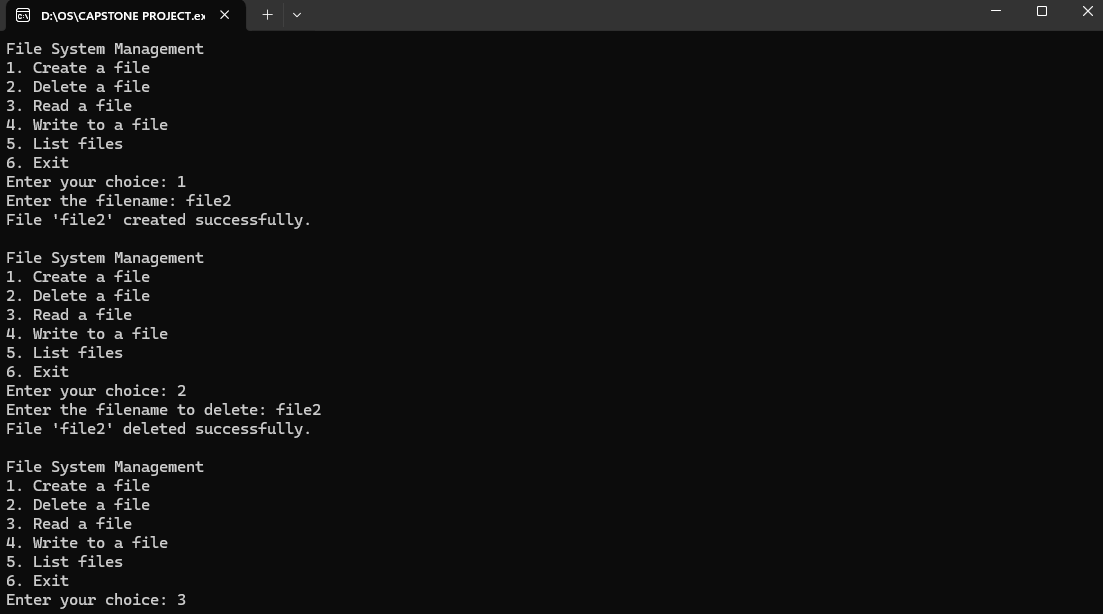
printf("Exiting the program.\n");

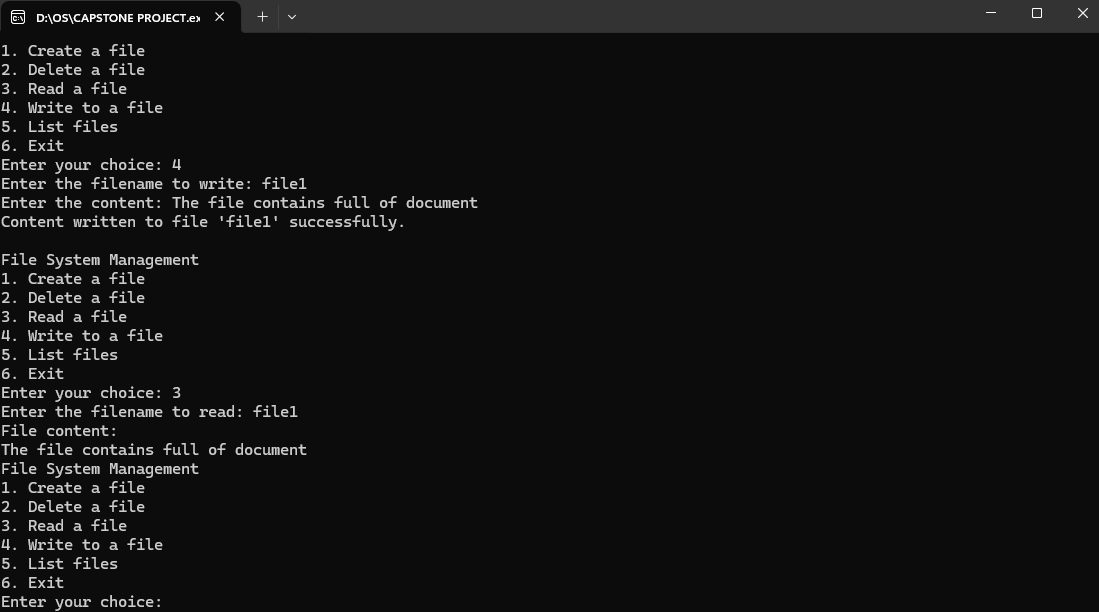
exit(0);

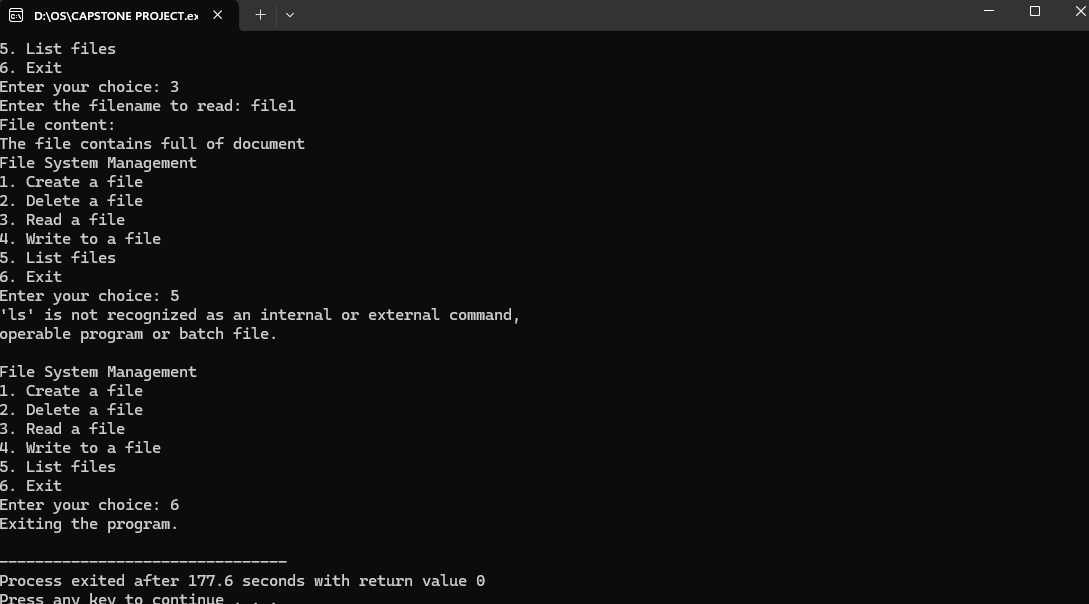
}

OUTPUT:









FUNCTIONALITIES:

The following are some of the functionalities or tasks performed by file management system:

1. Create a new file
2. Delete a file
3. Read a file
4. Write to a file
5. List file
6. Exit

The details of all above functionalities are already explained under MECHANISM AND WORKING heading in the form of code of each function.

REMAINING CODE MODULE, API’S AND PLATFORMS:

No other remaining side work apart from the displayed work above is used in this project. All the functionalities and code of each function is explained above. In this project we use Ubuntu subsystem terminal with C language and bash scripting. So, no others platform, API or plug ins used in this project.

FUTURE SCOPE:

This is the most basic version of file management system. So, in future we can improve the current version’s functionalities and can add more new functionalities to the system. In the current version of files management system there are 06 different options for a user to manage files and directories. In future we can add more choices for users by understanding the advanced concept about file management in Linux operating system. So, this will definitely help users to manage files in a more easy and comfortable manners.

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 system to perform different functions on files.