**File Management System**

## Submitted by

**Name of the Students:** Eeshan Ghosh **Enrolment Number: 12022002017082 Section:** A

**Class Roll Number:** 87

**Stream:** CSE

**Subject:** Programming for Problem Solving using C

**Subject Code:** ESC103(pr)

**Department:** Basic Science and Humanities

Under the supervision of Dr. Indrajit De

**Academic Year: 2022-26**

PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE FIRST SEMESTER



**DEPARTMENT OF BASIC SCIENCE AND HUMANITIES INSTITUTE OF ENGINEERING AND MANAGEMENT, KOLKATA**



## CERTIFICATE OF RECOMMENDATION

We hereby recommend that the project prepared under our supervision by Eeshan Ghosh**,** entitled “File Management System” accepted in partial fulfillment of the requirements for the degree of partial fulfillment of the 2nd semester.

Program:

#include <stdio.h>

#include <stdlib.h>

void createFile() {

FILE \*file;

char fileName[100];

printf("Enter the name of the file: ");

scanf("%s", fileName);

file = fopen(fileName, "w");

if (file == NULL) {

printf("Error creating file.\n");

return;

}

printf("File created successfully.\n");

fclose(file);

}

void writeFile() {

FILE \*file;

char fileName[100];

char content[1000];

printf("Enter the name of the file: ");

scanf("%s", fileName);

file = fopen(fileName, "a");

if (file == NULL) {

printf("Error opening file.\n");

return;

}

printf("Enter the content to write (max 1000 characters):\n");

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

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

printf("Content written to the file.\n");

fclose(file);

}

void readFile() {

FILE \*file;

char fileName[100];

char content[1000];

printf("Enter the name of the file: ");

scanf("%s", fileName);

file = fopen(fileName, "r");

if (file == NULL) {

printf("Error opening file.\n");

return;

}

printf("File content:\n");

while (fgets(content, sizeof(content), file) != NULL) {

printf("%s", content);

}

fclose(file);

}

void deleteFile() {

char fileName[100];

printf("Enter the name of the file: ");

scanf("%s", fileName);

if (remove(fileName) == 0) {

printf("File deleted successfully.\n");

} else {

printf("Error deleting file.\n");

}

}

int main() {

int choice;

printf("File Management System\n");

while (1) {

printf("\nSelect an option:\n");

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

printf("2. Write to a File\n");

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

printf("4. Delete a File\n");

printf("5. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

createFile();

break;

case 2:

writeFile();

break;

case 3:

readFile();

break;

case 4:

deleteFile();

break;

case 5:

printf("Thank you for using the file management system. Goodbye!\n");

return 0;

default:

printf("Invalid choice. Please try again.\n");

break;

}

}

return 0;

}

Head of the Department Project Supervisor Basic Sciences and Humanities

IEM, Kolkata

Project Report: File Management System

1. Introduction

The File Management System is a software application developed using the C programming language. The primary objective of this project is to create a user-friendly system for managing files and directories on a computer. The system provides various functionalities such as creating, deleting, renaming, and searching files and directories.

2. Features

The File Management System incorporates the following features:

a) File Operations:

- Create a new file

- Delete a file

- Rename a file

- Copy a file

- Move a file

- Display file contents

- Append content to a file

b) Directory Operations:

- Create a new directory

- Delete a directory

- Rename a directory

- Change current directory

- Display current directory

c) Search Operations:

- Search for a file or directory by name

3. System Design

The system is designed using modular programming techniques. The functionalities are divided into separate functions and grouped into modules based on their related operations. The following modules are implemented:

a) File Module: Implements file-related operations such as creating, deleting, renaming, copying, moving, displaying, and appending file contents.

b) Directory Module: Handles directory-related operations including creating, deleting, renaming, changing the current directory, and displaying the current directory.

c) Search Module: Provides a function to search for a file or directory by name.

d) Main Module: Contains the main function responsible for the user interface, handling user inputs, and invoking the appropriate functions from the other modules based on the user's commands.

4. Implementation

The File Management System is implemented using the C programming language. The system utilizes various standard C library functions and system calls to interact with the underlying operating system.

The program uses data structures such as arrays and structures to store file and directory information. The necessary algorithms are implemented to perform file and directory operations efficiently. Error handling and validation mechanisms are also implemented to ensure the program's robustness.

5. User Interface

The File Management System provides a command-line interface where users can interact with the system by entering commands and parameters. Users can enter specific commands to perform different operations such as creating files, deleting directories, renaming files, etc. The system displays appropriate messages and prompts for user inputs to guide them through the process.

6. Testing

A thorough testing process was conducted to ensure the reliability and accuracy of the File Management System. Various test cases were designed to cover different scenarios, including valid and invalid inputs, edge cases, and boundary conditions. The system was tested extensively to ensure that all the functionalities work as intended and produce the expected results.

7. Conclusion

The File Management System developed in C programming language provides an efficient solution for managing files and directories on a computer. It offers a range of features such as creating, deleting, renaming, copying, moving files, and directories, as well as searching for specific files or directories. The modular design and careful implementation ensure the system's reliability and effectiveness. The system can be further enhanced by incorporating additional features such as file permissions, file compression, and encryption. Overall, the File Management System contributes to improved file organization and accessibility for users.

OUTPUT:



