

Name:K.SaiKrishna

Reg-No: 192311106

26.Construct a C program to implement the file management operations.

Aim

To develop a C program to perform basic file management operations: create, read, write, and append data to a file.

Algorithm

1. Start the program.
2. Display a menu for file operations (Create/Write, Read, Append, Exit).
3. Based on the user's choice:
 - o **Create/Write:** Open a file in write mode, input data, and save it.
 - o **Read:** Open a file in read mode and display its contents.
 - o **Append:** Open a file in append mode and add new data.
4. Close the file after each operation.
5. Repeat until the user chooses to exit.
6. End the program.

Procedure

1. Use `fopen()` to create/open a file.
2. Perform operations using `fprintf()` for writing, `fscanf()` or `fgets()` for reading, and `fprintf()` for appending.
3. Handle user inputs and perform error checking (e.g., file not found).
4. Close the file using `fclose()`.

Code:

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
void createFile() {
```

```
    FILE *file = fopen("file.txt", "w");
```

```
    if (file == NULL) {
```

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

```
        return;
    }

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

    fclose(file);
}
```

```
void writeFile() {

    FILE *file = fopen("file.txt", "w");

    if (file == NULL) {

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

        return;

    }

    char data[100];

    printf("Enter content to write into the file: ");

    getchar();

    fgets(data, 100, stdin);

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

    printf("Data written successfully.\n");

    fclose(file);

}
```

```
void readFile() {

    FILE *file = fopen("file.txt", "r");
```

```
if (file == NULL) {

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

    return;

}

char ch;

printf("File content:\n");

while ((ch = fgetc(file)) != EOF) {

    putchar(ch);

}

fclose(file);

}


void appendFile() {

    FILE *file = fopen("file.txt", "a");

    if (file == NULL) {

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

        return;

    }

    char data[100];

    printf("Enter content to append to the file: ");

    getchar();

    fgets(data, 100, stdin);

    fprintf(file, "%s", data);
```

```
    printf("Data appended successfully.\n");

    fclose(file);
}

int main() {
    int choice;

    do {
        printf("\nFile Management System\n");
        printf("1. Create File\n");
        printf("2. Write to File\n");
        printf("3. Read File\n");
        printf("4. Append to 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: appendFile(); break;
            case 5: printf("Exiting...\n"); break;
            default: printf("Invalid choice. Try again.\n");
        }
    }
```

```

    } while (choice != 5);

    return 0;

}

```

Result

The program successfully implements file management operations:

1. **Create File:** Creates an empty file named `file.txt`.
2. **Write File:** Writes user input to the file.
3. **Read File:** Reads and displays the file's contents.
4. **Append File:** Appends additional content to the file.

Output:

The screenshot displays an online C++ compiler interface. On the left is a sidebar with navigation links: 'Welcome, K Sai Krishna', 'Create New Project', 'My Projects', 'Classroom' (marked as new), 'Learn Programming', 'Programming Questions', 'Upgrade', and 'Logout'. The main area shows the source code for a file management program. The code includes functions for creating, writing, reading, and appending to a file named 'testfile.txt'. The output window at the bottom shows the program's execution results, including the success of file creation, the initial content read from the file, and the appended content. The program ends with an exit code of 0.

```

58     close(fd);
59     exit(1);
60 }
61 printf("Data appended to file '%s' successfully.\n", filename);
62 close(fd);
63 }
64
65 int main() {
66     const char *filename = "testfile.txt";
67
68     // Step 1: Create a file
69     createFile(filename);
70
71     // Step 2: Write to the file
72     writeFile(filename, "This is the initial content of the file.\n");
73
74     // Step 3: Read the file
75     readFile(filename);
76
77     // Step 4: Append to the file
78     appendFile(filename, "This is appended content.\n");
79
80     // Step 5: Read the file again
81     readFile(filename);
82
83     return 0;
84 }
85

```

input

```

Data appended to file 'testfile.txt' successfully.
Data read from file 'testfile.txt':
This is the initial content of the file.
This is appended content.

...Program finished with exit code 0
Press ENTER to exit console.

```

Result

The program successfully implements file management operations:

1. **Create File:** Creates an empty file named `file.txt`.
2. **Write File:** Writes user input to the file.
3. **Read File:** Reads and displays the file's contents.
4. **Append File:** Appends additional content to the file.