

LAB 4 : File Management System in Operating System

Nirajan Bekoju
PUL076BCT039
2nd March, 2023

I. INTRODUCTION

A file management system in an operating system is responsible for managing the storage and retrieval of files on a disk or other storage device. It provides an interface to create, read, write, delete and modify files.

In a file management system, a file is a named collection of data that is stored in a disk or other storage device. Each file is identified by a unique name, which is used to locate it in the file system.

Reading a file in C programming involves opening the file with the `fopen()` function, which returns a pointer to a `FILE` structure. This structure contains information about the file, such as its name and position in the file. After opening the file, data can be read from it using functions such as `fread()` or `fgets()`.

To write to a file in C programming, the file must be opened in write mode using the `fopen()` function with the "w" parameter. Data can then be written to the file using functions such as `fwrite()` or `fputs()`.

Updating a file involves opening the file in update mode using the `fopen()` function with the "r+" or "w+" parameter. This allows both reading and writing to the file. Data can be read from the file using functions such as `fread()` or `fgets()`, and data can be written to the file using functions such as `fwrite()` or `fputs()`.

In addition to basic file operations, a file management system may also provide features such as file locking, file permissions, and file compression. These features help ensure the security and efficiency of the file system.

II. FILE CREATION

Aim : To write a C program to create a file.

Algorithm :

STEP 1 : Start the program.

STEP 2 : Create the file using create function and assign a variable to it.

STEP 3 : If the value of the variable is less then print file cannot be created ,Otherwise print file is created.

STEP 4 : Stop the program.

Program :

```
#include <stdio.h>

#include<sys/types.h>
#include<sys/stat.h>

int main(){
    int id;
    // create return 0 on success and -1 on error
    if(id = creat("z.txt", 0) == -1){
        printf("cannot create the file\n");
        exit(1);
    }
    else{
        printf("file is created");
        exit(1);
    }
}
```

Ouput :

file is created

```
$ ls
z.txt
```

III. WRITING INTO A FILE

Aim : To write a C program to write the data into a file.

Algorithm :

- STEP 1 : Get the data from the user.
- STEP 2 : Open a file.
- STEP 3 : Write the data from the file.
- STEP 4 : Get the data and update the file.

Program :

```
#include <stdio.h>

int main(){
    char str[100];
    FILE *fp;
    printf("Enter the string : ");
    gets(str);

    fp = fopen("file1.dat", "w+");
    while (!feof(fp))
    {
        fscanf(fp, "%s", str);
    }
    fprintf(fp, "%s", str);
    fclose(fp);
}
```

Output:

```
Enter the string : this is file management system in os.
```

```
$ cat file1.dat
this is file management system in os.
```

IV. READING FROM A FILE

Aim : To create and read data from the file.

Algorithm :

- STEP 1 : Get the data from the user.
- STEP 2 : Open a file.
- STEP 3 : Read from the file.
- STEP 4 : Close the file.

Program :

```
#include <stdio.h>

int main(){
    char str[100];
    FILE *fp;
    fp = fopen("file1.dat", "r");
    while (!feof(fp))
    {
        fscanf(fp, "%s", str);
        printf("%s ", str);
    }
    printf("\n");
    fclose(fp);
}
```

Output :

```
this is file management system in os.
```

V. CONCLUSION

In this lab, we studied file management system in OS. We implemented C program to create, read and write file. Various modes of file and file handling system of OS was explored. We studied various function in C programming for file handling like feof, FILE*, fopen, fclose, fscanf, etc.