OS LAB MANUAL (CS23431)

Roll No:230701258

EX.NO:12

File Organization Technique- Single and Two level directory

Aim: To implement File Organization Structures in C are:

a. Single Level Directory

b. Two-Level Directory

A) Single Level Directory

Program:

```c
#include <stdio.h>
#include <string.h>
int main() {
    int n, i;
    char files[10][30];
    printf("Enter the number of files: ");
    scanf("%d", &n);
    printf("\nCreating Single-Level Directory Structure...\n");
    for (i = 0; i < n; i++) {
        printf("\nEnter the name of file %d: ", i + 1);
        scanf("%s", files[i]);
        printf("\n+----------------+\n");
        printf("|  Root Directory |\n");
        printf("+ ----------------+\n");
```

```c
        for (int j = 0; j <= i; j++) {

            printf("        |\n");

            printf("        +--> [ %s ]\n", files[j]);

        }

    }

    return 0;

}
```

Input:

```
Enter the number of files: 2

Creating Single-Level Directory Structure...

Enter the name of file 1: A
```

Output:

```
Enter the number of files: 2

Creating Single-Level Directory Structure...

Enter the name of file 1: A

+---------------+
|  Root Directory |
+---------------+
        |
        +--> [ A ]

Enter the name of file 2: D

+---------------+
|  Root Directory |
+---------------+
        |
        +--> [ A ]
        |
        +--> [ D ]
```

B) Two-Level Directory :

Program:

```c
#include  <stdio.h>
#include <string.h>
struct File {
    char name[30];
};
struct User {
    char name[30];
    int fileCount;
    struct File files[10];
};
int main() {
    int userCount, i, j;
    struct User users[10];
    printf("Enter the number of users (directories): ");
    scanf("%d", &userCount);
    for (i = 0; i < userCount; i++) {
        printf("\nEnter the name of User %d: ", i + 1);
        scanf("%s", users[i].name);
        printf("Enter number of files for %s: ", users[i].name);
        scanf("%d", &users[i].fileCount);
        for (j = 0; j < users[i].fileCount; j++) {
            printf("Enter file %d name for %s: ", j + 1, users[i].name);
            scanf("%s", users[i].files[j].name);
        }
    }
    printf("\n\nTwo-Level Directory Structure:\n");
    printf("+------------------------------+\n");
    printf("|      Root Directory      |\n");
    printf("+------------------------------+\n");
```

```c
    for (i = 0; i < userCount; i++) {

        printf("       |\n");

        printf("       +--> User: %s\n", users[i].name);

        for (j = 0; j < users[i].fileCount; j++) {

            printf("           |\n");

            printf("           +--> File: %s\n", users[i].files[j].name);

        }

    }

    return 0;

}
```

Input:

```
Enter the number of users (directories): 2

Enter the name of User 1: joe
Enter number of files for joe: 2
Enter file 1 name for joe: A
Enter file 2 name for joe: B

Enter the name of User 2: ram
Enter number of files for ram: 2
Enter file 1 name for ram: A
Enter file 2 name for ram: B
```

Output:

```
Two-Level Directory Structure:
+------------------------+
|      Root Directory    |
+------------------------+
         |
        +--> User: joe
             |
             +--> File: A
             |
             +--> File: B
        |
        +--> User: ram
             |
             +--> File: A
             |
             +--> File: B
```