Ex. No.: 12 **231901039**

Date: 16/4/2025

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
c. Hierarchical Directory Structure
d. Directed Acyclic Graph Structure

a. Single Level Directory

ALGORITHM

1. Start
2. Declare the number, names and size of the directories and file names.
3. Get the values for the declared variables.
4. Display the files that are available in the directories.
5. Stop.

PROGRAM:

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

struct Directory {
    char name[20];
    int fileCount;
    char files[10][20];
};

int main() {
    struct Directory dir;
    int i;

    printf("Enter the name of the directory: ");
    scanf("%s", dir.name);
```

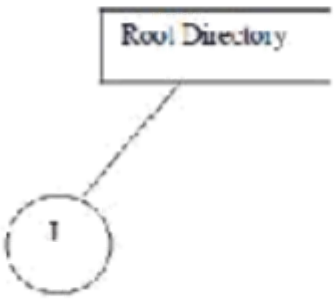
```
printf("Enter number of files: ");
scanf("%d", &dir.fileCount);

for (i = 0; i < dir.fileCount; i++) {
    printf("Enter name of file %d: ", i + 1);
    scanf("%s", dir.files[i]);
}

printf("\nDirectory Name: %s\n", dir.name);
printf("Files:\n");
for (i = 0; i < dir.fileCount; i++) {
    printf("%s\n", dir.files[i]);
}return 0;}
```

OUTPUT:

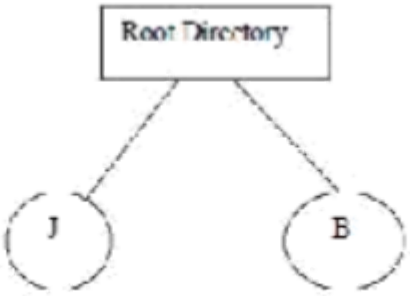
Enter the Number of files

2

Enter the file1 J



Enter the file2 B



b. Two-level directory Structure

ALGORITHM:

1. Start
2. Declare the number, names and size of the directories and subdirectories and file names.

3. Get the values for the declared variables.
4. Display the files that are available in the directories and subdirectories.
5. Stop.

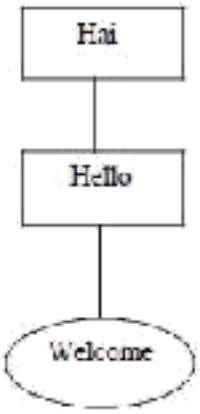
PROGRAM:

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

struct SubDirectory {
    char name[20];
    int fileCount;
    char files[10][20];};

struct Directory {
    char userName[20];
    int subDirCount;
    struct SubDirectory subDirs[10];
};

int main() {
    struct Directory dir[10];
    int userCount, i, j, k;

    printf("Enter the number of users (main
directories): ");
    scanf("%d", &userCount);

    for (i = 0; i < userCount; i++) {
        printf("\nEnter name for user %d: ", i + 1);
        scanf("%s", dir[i].userName);

        printf("Enter number of subdirectories for
%s: ", dir[i].userName);
        scanf("%d", &dir[i].subDirCount);

        for (j = 0; j < dir[i].subDirCount; j++) {
            printf("  Enter name of subdirectory %d:
", j + 1);
            scanf("%s", dir[i].subDirs[j].name);
```

```
            printf("  Enter number of files in
subdirectory %s: ", dir[i].subDirs[j].name);
            scanf("%d",
&dir[i].subDirs[j].fileCount);

            for (k = 0; k <
dir[i].subDirs[j].fileCount; k++) {
                printf("    Enter file %d name: ", k
+ 1);
                scanf("%s",
dir[i].subDirs[j].files[k]);
            }
        }
    }

    // Display
    printf("\n===== Directory Structure =====\n");
    for (i = 0; i < userCount; i++) {
        printf("\nUser: %s\n", dir[i].userName);
        for (j = 0; j < dir[i].subDirCount; j++) {
            printf("  Subdirectory: %s\n",
dir[i].subDirs[j].name);
            for (k = 0; k <
dir[i].subDirs[j].fileCount; k++) {
                printf("    File: %s\n",
dir[i].subDirs[j].files[k]);
            }
        }
    }

    return 0;
}
```

Sample Output:

Enter the name of dir/file(under null): Hai
How many users(for Hai):1
Enter name of dir/file(under Hai):Hello
How many files(for Hello):1
Enter name of dir/file(under Hello):welcome



Result:

Single-Level Directory, Two-Level Directory, Hierarchical Directory Structure and Directed Acyclic Graph Structure have been implemented using C.