**Sachin Chaudhary**

**230701402**

**Ex. No. 12:**

**File Organization Technique - Single and Two Level Directory**

AIM:

To implement File Organization Structures in C:

a. Single Level Directory

b. Two-Level Directory

c. Hierarchical Directory Structure

d. Directed Acyclic Graph Structure

a. Single Level Directory

ALGORITHM:

Start.

Declare the number of directories, names, and size of the files in the directories.

Get the values for the declared variables.

Display the files that are available in the directories.

Stop.

Program Code (Single Level Directory in C):

#include <stdio.h>

#include <stdlib.h>

void single\_level\_directory() {

int n;

printf("Enter number of files: ");

scanf("%d", &n);

char files[n][50];

// Get file names

for (int i = 0; i < n; i++) {

printf("Enter name of file %d: ", i+1);

scanf("%s", files[i]);

}

// Display files in the directory

printf("\nFiles in the directory:\n");

for (int i = 0; i < n; i++) {

printf("%s\n", files[i]);

}

}

int main() {

single\_level\_directory();

return 0;

}

Sample Output (Single Level Directory):

Enter number of files: 3

Enter name of file 1: file1.txt

Enter name of file 2: file2.txt

Enter name of file 3: file3.txt

Files in the directory:

file1.txt

file2.txt

file3.txt

Result:

The program successfully implements a single-level directory structure, allowing users to input file names and display them in the directory.

b. Two-Level Directory Structure

ALGORITHM:

Start.

Declare the number of directories, subdirectories, and file names.

Get the values for the declared variables.

Display the files that are available in the directories and subdirectories.

Stop.

Program Code (Two-Level Directory in C):

#include <stdio.h>

#include <stdlib.h>

void two\_level\_directory() {

int n;

printf("Enter number of directories: ");

scanf("%d", &n);

char directories[n][50];

// Get directory names

for (int i = 0; i < n; i++) {

printf("Enter name of directory %d: ", i+1);

scanf("%s", directories[i]);

}

// For each directory, get subdirectories and files

for (int i = 0; i < n; i++) {

printf("\nEnter number of subdirectories under %s: ", directories[i]);

int sub\_count;

scanf("%d", &sub\_count);

char subdirectories[sub\_count][50];

// Get subdirectory names

for (int j = 0; j < sub\_count; j++) {

printf("Enter name of subdirectory %d under %s: ", j+1, directories[i]);

scanf("%s", subdirectories[j]);

int file\_count;

printf("How many files in %s: ", subdirectories[j]);

scanf("%d", &file\_count);

char files[file\_count][50];

// Get file names

for (int k = 0; k < file\_count; k++) {

printf("Enter name of file %d in %s: ", k+1, subdirectories[j]);

scanf("%s", files[k]);

}

// Display files in subdirectory

printf("\nFiles in subdirectory %s under directory %s:\n", subdirectories[j], directories[i]);

for (int k = 0; k < file\_count; k++) {

printf("%s\n", files[k]);

}

}

}

}

int main() {

two\_level\_directory();

return 0;

}

Sample Output (Two-Level Directory):

Enter number of directories: 1

Enter name of directory 1: Hai

Enter number of subdirectories under Hai: 1

Enter name of subdirectory 1 under Hai: Hello

How many files in Hello: 1

Enter name of file 1 in Hello: welcome

Files in subdirectory Hello under directory Hai:

welcome

Result:

The program successfully implements a two-level directory structure, allowing users to define directories, subdirectories, and their respective files, then displaying them.