**EX 15:Design a C program to organise the file using a two level directory structure.**

**Aim:**

To construct a C program to organize files using a two-level directory structure, where each user has their own directory, and files are organized under those directories.

**Algorithm:**

1. **Initialize Data Structures:**
   * Use an array of user directories, each containing an array of file names.
2. **Display Menu:**
   * Show options for user directory and file operations.
3. **Handle User Input:**
   * Add User Directory:
     + Create a new directory for a user.
   * Add File:
     + Add a file under a specific user's directory.
   * Search File:
     + Check if a file exists in a specific user's directory.
   * Delete File:
     + Remove a file from a specific user's directory.
   * Display Files:
     + Show all files in a specific user's directory.
4. **Repeat Until Exit.**

**Program:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <dirent.h>

#include <sys/stat.h>

void createDirectory(const char \*path) {

struct stat st = {0};

if (stat(path, &st) == -1) {

mkdir(path, 0700);

}

}

void organizeFiles(const char \*sourceDir, const char \*destDir) {

struct dirent \*entry;

DIR \*dp = opendir(sourceDir);

if (dp == NULL) {

perror("opendir");

return;

}

while ((entry = readdir(dp))) {

if (entry->d\_type == DT\_REG) {

char \*ext = strrchr(entry->d\_name, '.');

if (ext) {

ext++; // Move past the dot

char subDir[256];

snprintf(subDir, sizeof(subDir), "%s/%s", destDir, ext);

createDirectory(subDir);

char sourceFile[256];

char destFile[256];

snprintf(sourceFile, sizeof(sourceFile), "%s/%s", sourceDir, entry->d\_name);

snprintf(destFile, sizeof(destFile), "%s/%s/%s", destDir, ext, entry->d\_name);

rename(sourceFile, destFile);

}

}

}

closedir(dp);

}

int main(int argc, char \*argv[]) {

if (argc != 3) {

fprintf(stderr, "Usage: %s <source\_directory> <destination\_directory>\n", argv[0]);

return EXIT\_FAILURE;

}

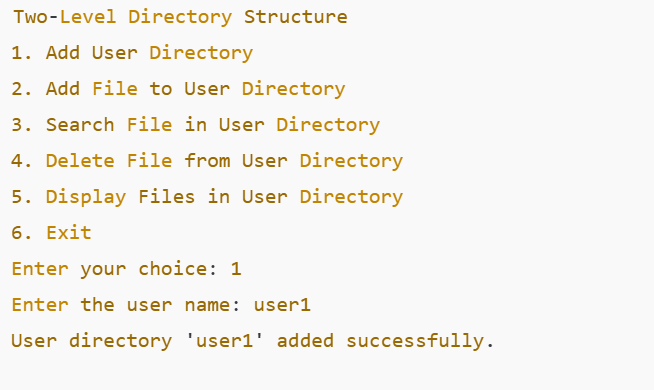
createDirectory(argv[2]);

organizeFiles(argv[1], argv[2]);

return EXIT\_SUCCESS;

}

**OUTPUT:**

****