**8. Simulate following File Organization Techniques: a) Single level directory b) Two level directory**

**SOURCE CODE:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_FILES 5

#define MAX\_NAME\_LENGTH 20

typedef struct {

char name[MAX\_NAME\_LENGTH];

int size;

} File;

typedef struct {

char name[MAX\_NAME\_LENGTH];

int numFiles;

File files[MAX\_FILES];

} Directory;

Directory singleLevelDirectory;

Directory twoLevelDirectory[MAX\_FILES];

void initializeSingleLevelDirectory() {

strcpy(singleLevelDirectory.name, "Root");

singleLevelDirectory.numFiles = 0;

}

void initializeTwoLevelDirectory() {

for (int i = 0; i < MAX\_FILES; ++i) {

sprintf(twoLevelDirectory[i].name, "Directory%d", i + 1);

twoLevelDirectory[i].numFiles = 0;

}

}

void displaySingleLevelDirectory() {

printf("Single Level Directory:\n");

printf("Directory Name: %s\n", singleLevelDirectory.name);

printf("Number of Files: %d\n", singleLevelDirectory.numFiles);

printf("Files:\n");

for (int i = 0; i < singleLevelDirectory.numFiles; ++i) {

printf("File Name: %s, Size: %d KB\n", singleLevelDirectory.files[i].name, singleLevelDirectory.files[i].size);

}

printf("\n");

}

void displayTwoLevelDirectory() {

printf("Two Level Directory:\n");

for (int i = 0; i < MAX\_FILES; ++i) {

printf("Directory Name: %s\n", twoLevelDirectory[i].name);

printf("Number of Files: %d\n", twoLevelDirectory[i].numFiles);

printf("Files:\n");

for (int j = 0; j < twoLevelDirectory[i].numFiles; ++j) {

printf("File Name: %s, Size: %d KB\n", twoLevelDirectory[i].files[j].name, twoLevelDirectory[i].files[j].size);

}

printf("\n");

}

}

void addFileSingleLevelDirectory(char name[], int size) {

if (singleLevelDirectory.numFiles < MAX\_FILES) {

strcpy(singleLevelDirectory.files[singleLevelDirectory.numFiles].name, name);

singleLevelDirectory.files[singleLevelDirectory.numFiles].size = size;

singleLevelDirectory.numFiles++;

printf("File '%s' added to Single Level Directory\n", name);

} else {

printf("Single Level Directory is full, cannot add file '%s'\n", name);

}

}

void addFileTwoLevelDirectory(char name[], int size, int directoryIndex) {

if (directoryIndex >= 0 && directoryIndex < MAX\_FILES) {

if (twoLevelDirectory[directoryIndex].numFiles < MAX\_FILES) {

strcpy(twoLevelDirectory[directoryIndex].files[twoLevelDirectory[directoryIndex].numFiles].name, name);

twoLevelDirectory[directoryIndex].files[twoLevelDirectory[directoryIndex].numFiles].size = size;

twoLevelDirectory[directoryIndex].numFiles++;

printf("File '%s' added to Directory '%s'\n", name, twoLevelDirectory[directoryIndex].name);

} else {

printf("Directory '%s' is full, cannot add file '%s'\n", twoLevelDirectory[directoryIndex].name, name);

}

} else {

printf("Invalid Directory Index for Two Level Directory\n");

}

}

int main() {

initializeSingleLevelDirectory();

initializeTwoLevelDirectory();

addFileSingleLevelDirectory("file1.txt", 1024);

addFileSingleLevelDirectory("file2.txt", 2048);

addFileSingleLevelDirectory("file3.txt", 3072);

displaySingleLevelDirectory();

addFileTwoLevelDirectory("file4.txt", 4096, 0);

addFileTwoLevelDirectory("file5.txt", 5120, 1);

addFileTwoLevelDirectory("file6.txt", 6144, 2);

displayTwoLevelDirectory();

return 0;

}