****

**LAB EXERCISE 13**

**File Organization Techniques**

**Submission Date:30-05-2022**

Name: Jayannthan P T

Dept: CSE ‘A’

Roll No.: 205001049

1. To develop a C program to implement the following file organization techniques

a) Single level Directory

b) Hierarchical Structure.

**Algorithm:**

1. Single Level Directory

a. Maintain a table containing the filename and the starting address location of that file.

b. Give options for creating a new file.

c. When creating the file, check for name collision.

d. Update the table accordingly.

2. Tree Structured Directory

a. Maintain tables for each directory starting from root.

b. Create a structure for a node in tree which contains an array to hold directories and an array to hold files.

c. Limit each directory to have a maximum of three sub-directories and files.

d. For each sub-directory follow the same table structure as described above

**Code:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

**char** files[100][20];

**int** address[100];

**int** cur = 1;

typedef **struct** dir

{

**char** dirname[20];

**char** filename[100][10];

**int** address[100];

**int** curfile;

**struct** dir \*ptr1, \*ptr2, \*ptr3;

}

dir;

dir \* root;

dir\* create(**char** name[20])

{

    dir \*p = (dir\*) malloc(sizeof(dir));

    p->ptr1 = NULL;

    p->ptr2 = NULL;

    p->ptr3 = NULL;

    p->curfile = 0;

    strcpy(p->dirname, name);

    return p;

}

**void** singleLevel()

{

    cur = 1;

    printf("\n\t\t\tSINGLE LEVEL FILE SYSTEM\n");

**int** choice = -1;

    printf("\n\tMenu : \n\t1.Create file\n\t2.Print files \n\t3.Exit\n\t\tEnter Choice:");

    scanf("%d", &choice);

    do

    {   *// if (choice == 3)return;*

        if (choice == 2)

        {

            printf("\nFiles.....\n");

            for (**int** i = 1; i <= cur-1; i++) printf("%d.%s\t \n", i, files[i]);

            printf("\nFiles printed...\n");

        }

        else if (choice == 1)

        {

**char** name[20];

**int** found = 0;

            printf("\n\tName of the file : ");

            scanf("%s", name);

            for (**int** i = 1; i <= cur-1; i++)

            {

                if (strcmp(name, files[i]) == 0)

                {

                    found = 1;

                    break;

                }

            }

            if (found)

            {

                printf("\n\tFile name already exists!\n");

            }

            else

            {

                strcpy(files[cur++], name);

                address[cur - 1] = rand() % 10000;

                printf("\n\tFile created succcessfully!\n");

            }

        }

        printf("\n\tMenu : \n\t1.Create file\n\t2.Print files \n\t3.Exit\n\t\tEnter Choice: ");

        scanf("%d", &choice);

    } while (choice != 3);

}

dir\* find(dir \*p, **char** name[20])

{

    if (p == NULL) return NULL;

    if (strcmp(p->dirname, name) == 0) return p;

    dir \*p1 = find(p->ptr1, name);

    if (p1 != NULL) return p1;

    dir \*p2 = find(p->ptr2, name);

    if (p2) return p2;

    dir \*p3 = find(p->ptr3, name);

    if (p3) return p3;

    return NULL;

}

**void** print(dir \*p)

{

    printf("\nDirectory Name : %s\n", p->dirname);

    if (p->curfile == 0)

    {

        printf("\n\tNO FILES IN DIRECTORY!\n");

    }

    else

    {

        printf("Files in Directory :");

        for (**int** i = 1; i <= p->curfile; i++)

        {

            printf("\n\t%s", p->filename[i]);

        }

    }

    if (p->ptr1 == NULL) return;

    printf("\n\tSubdirectories :");

    if (p->ptr1 != NULL)

    {

        printf("\n\t\t%s", p->ptr1->dirname);

    }

    if (p->ptr2 != NULL)

    {

        printf("\n\t\t%s", p->ptr2->dirname);

    }

    if (p->ptr3 != NULL)

    {

        printf("\n\t\t%s", p->ptr3->dirname);

    }

    printf("\n\n");

}

**void** display(dir \*p)

{

    if (p == NULL)

        return;;

    print(p);

    display(p->ptr1);

    display(p->ptr2);

    display(p->ptr3);

}

**void** hierarchial()

{

    cur = 1;

    root = create("root");

    printf("\n\t\t\tTREE LEVEL FILE SYSTEM\n");

**int** choice = -1;

    printf("\n\tMenu : \n\t1.Create directory\n\t2.Create file\n\t3.Print files \n\t4.Exit\n\t\tEnter Choice: ");

    scanf("%d", &choice);

    do

    {

        if (choice == 1)

        {

**char** name[20];

            printf("\n\tName of directory to be created :");

            scanf("%s", name);

**char** parent[20];

            printf("\n\tParent name : ");

            scanf("%s", parent);

            dir \*p = find(root, parent);

            if (p == NULL) printf("\n\tDirectory not found!");

            else

            {

                if (p->ptr1 == NULL)

                {

                    dir \*temp = create(name);

                    p->ptr1 = temp;

                    printf("\n\tDirectory succcessfully created!\n");

                }

                else if (p->ptr2 == NULL)

                {

                    if (strcmp(p->ptr1->dirname, name) == 0) printf("\n\t Name already exists!\n");

                    else

                    {

                        dir \*temp = create(name);

                        p->ptr2 = temp;

                        printf("\n\tDirectory succcessfully created!\n");

                    }

                }

                else if (p->ptr3 == NULL)

                {

                    if (strcmp(p->ptr1->dirname, name) == 0 || strcmp(p->ptr2->dirname, name) == 0) printf("\n\tDirectory Name already exists!\n");

                    else

                    {

                        dir \*temp = create(name);

                        p->ptr3 = temp;

                        printf("\n\tDirectory succcessfully created!\n");

                    }

                }

                else printf("\nCannot create directory. Space exceeded!\n");

            }

        }

        else if (choice == 2)

        {

**char** file[20];

            printf("\n\tEnter file name : ");

            scanf("%s", file);

            printf("\n\tEnter directory under which you want to create the file : ");

**char** direc[20];

            scanf("%s", direc);

            dir \*p = find(root, direc);

            if (p == NULL)

            {

                printf("\n\tDirectory does not exist!");

            }

            else

            {

**int** found = 0;

                for (**int** i = 1; i <= p->curfile; i++)

                {

                    if (strcmp(p->filename[i], file) == 0)

                    {

                        found = 1;

                        break;

                    }

                }

                if (found) printf("\n\tFilename already exists!");

                else

                {

                    strcpy(p->filename[++p->curfile], file);

                    p->address[p->curfile] = rand() % 10000;

                    printf("\n\tFile succcessfully created!");

                }

            }

        }

        else if (choice == 3)

        {

            printf("\nDisplaying directory structure.....\n");

            display(root);

        }

        else

        {

            printf("\n Enter Valid Choice\n");

        }

        printf("\n\tMenu : \n\t1.Create directory\n\t2.Create file\n\t3.Print files \n\t4.Exit\n\t\tEnter Choice: ");

        scanf("%d", &choice);

    } while (choice != 4);

}

**int** main()

{

**int** option = -1;

    printf("\nMenu : \n1.Single level\n2.Hierarchial \n3.Exit\n\tEnter Choice: ");

    scanf("%d", &option);

    do {

        switch (option)

        {

            case 1:

                singleLevel();

                break;

            case 2:

                hierarchial();

                break;

            default:

                printf("\n Enter Valid Choice\n");

                break;

        }

        printf("\nMenu : \n1.Single level\n2.Hierarchial \n3.Exit\n\tEnter Choice: ");

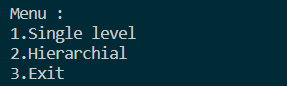
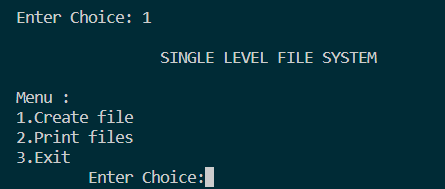
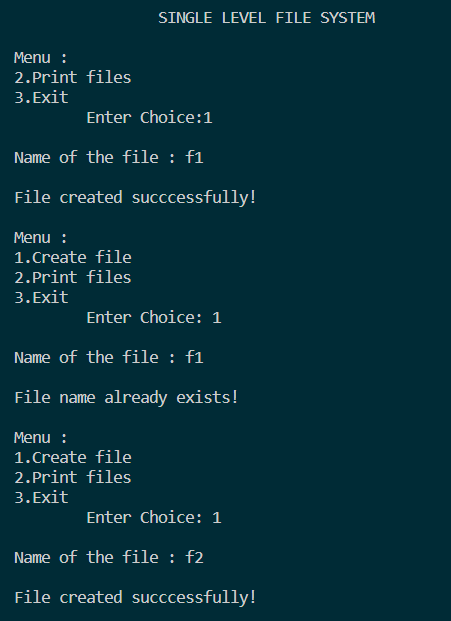
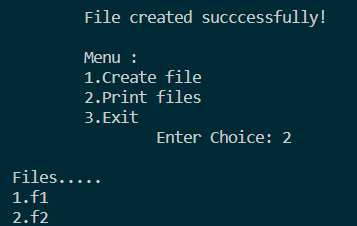
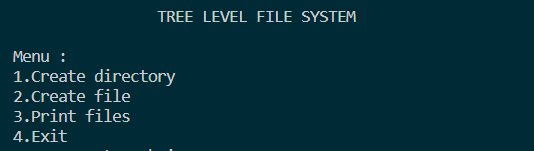
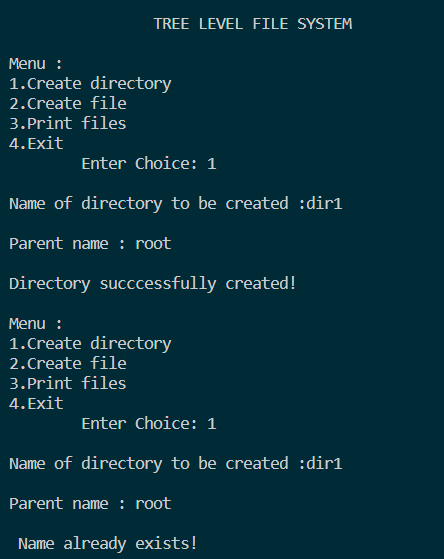
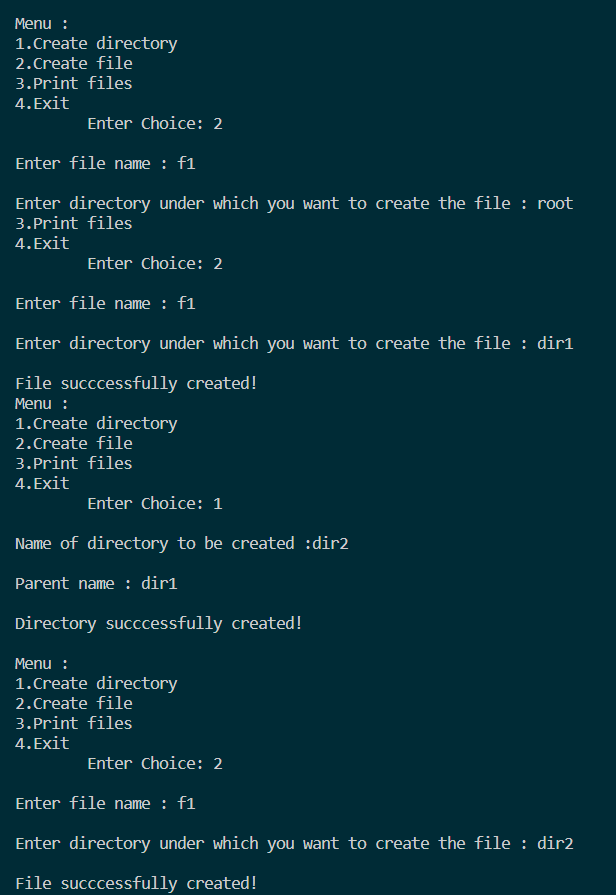
        scanf("%d", &option);

    } while (option != 3);

    return 0;

}

**Output:**

**      **

**Learning Outcome:**

* Learnt about the different directory structures
* Implemented single level directory and tree structured directory using c program