1.single directory

Code:

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

struct Directory {

    char dname[10];

    char fname[10][10];

    int fcnt;

};

void main() {

    int i, ch, k, flag = 0;

    char f[30];

    struct Directory dir;

    dir.fcnt = 0;

    printf("\nEnter name of directory -- ");

    scanf("%s", dir.dname);

    while(1) {

        printf("\n\n1. Create File\t 2. Delete File\t 3. Search File \t 4. Display Files\t 5. Exit\n");

        printf("Enter your choice: ");

        scanf("%d", &ch);

        switch (ch) {

            case 1:

                printf("\nEnter the name of the file -- ");

                scanf("%s", dir.fname[dir.fcnt]);

                dir.fcnt++;

                break;

            case 2:

                printf("\nEnter the name of the file -- ");

                scanf("%s", f);

                for (i = 0; i < dir.fcnt; i++) {

                    if (strcmp(f, dir.fname[i]) == 0) {

                        printf("File %s is deleted ", f);

                        for (k = i; k < dir.fcnt - 1; k++)

                            strcpy(dir.fname[k], dir.fname[k + 1]);

                        dir.fcnt--;

                        flag = 1;

                    }

                }

                if (flag == 0)

                    printf("File %s not found", f);

                break;

            case 3:

                flag = 0;

                printf("\nEnter the name of the file -- ");

                scanf("%s", f);

                for (i = 0; i < dir.fcnt; i++) {

                    if (strcmp(f, dir.fname[i]) == 0) {

                        printf("File %s is found ", f);

                        flag = 1;

                        break;

                    }

                }

                if (flag == 0)

                    printf("File %s not found", f);

                break;

            case 4:

                if (dir.fcnt == 0)

                    printf("\nDirectory Empty");

                else {

                    printf("\nThe Files are -- ");

                    for (i = 0; i < dir.fcnt; i++)

                        printf("\t%s", dir.fname[i]);

                }

                break;

            default:

                exit(0);

        }

    }

}

OUTPUT:

Enter name of directory -- nova

1. Create File 2. Delete File 3. Search File 4. Display Files 5. Exit

Enter your choice: 1

Enter the name of the file -- order

1. Create File 2. Delete File 3. Search File 4. Display Files 5. Exit

Enter your choice: 1

Enter the name of the file -- paraboy

1. Create File 2. Delete File 3. Search File 4. Display Files 5. Exit

Enter your choice: 1

Enter the name of the file -- dreamy

1. Create File 2. Delete File 3. Search File 4. Display Files 5. Exit

Enter your choice: 1

Enter the name of the file -- qc

1. Create File 2. Delete File 3. Search File 4. Display Files 5. Exit

Enter your choice: 4

The Files are -- order paraboy dreamy qc

1. Create File 2. Delete File 3. Search File 4. Display Files 5. Exit

Enter your choice: 3

Enter the name of the file -- order

File order is found

1. Create File 2. Delete File 3. Search File 4. Display Files 5. Exit

Enter your choice: 3

Enter the name of the file -- loong

File loong not found

1. Create File 2. Delete File 3. Search File 4. Display Files 5. Exit

Enter your choice: 2

Enter the name of the file -- order

File order is deleted

1. Create File 2. Delete File 3. Search File 4. Display Files 5. Exit

Enter your choice: 4

The Files are -- paraboy dreamy qc

1. Create File 2. Delete File 3. Search File 4. Display Files 5. Exit

Enter your choice: 2

Enter the name of the file -- paraboy

File paraboy is deleted

1. Create File 2. Delete File 3. Search File 4. Display Files 5. Exit

Enter your choice: 4

The Files are -- dreamy qc

1. Create File 2. Delete File 3. Search File 4. Display Files 5. Exit

Enter your choice: 2

Enter the name of the file -- dreamy

File dreamy is deleted

1. Create File 2. Delete File 3. Search File 4. Display Files 5. Exit

Enter your choice: 2

Enter the name of the file -- qc

File qc is deleted

1. Create File 2. Delete File 3. Search File 4. Display Files 5. Exit

Enter your choice: 4

Directory Empty

1. Create File 2. Delete File 3. Search File 4. Display Files 5. Exit

Enter your choice: 5

PS C:\C Programs>

2.two directory

Code:

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

struct Directory {

    char dname[10];

    char fname[10][10];

    int fcnt;

} dir[10];

void main() {

    int i, ch, dcnt, k;

    char f[30], d[30];

    dcnt = 0;

    while (1) {

        printf("\n\n 1. Create Directory\t 2. Create File\t 3. Delete File");

        printf("\n 4. Search File \t \t 5. Display \t 6. Exit \t\n");

        printf("Enter your choice: ");

        scanf("%d", &ch);

        switch (ch) {

            case 1:

                printf("\n Enter name of directory -- ");

                scanf("%s", dir[dcnt].dname);

                dir[dcnt].fcnt = 0;

                dcnt++;

                printf("Directory created");

                break;

            case 2:

                printf("\n Enter name of the directory -- ");

                scanf("%s", d);

                for (i = 0; i < dcnt; i++)

                    if (strcmp(d, dir[i].dname) == 0) {

                        printf("Enter name of the file -- ");

                        scanf("%s", dir[i].fname[dir[i].fcnt]);

                        dir[i].fcnt++;

                        printf("File created");

                        break;

                    }

                if (i == dcnt)

                    printf("Directory %s not found", d);

                break;

            case 3:

                printf("\nEnter name of the directory -- ");

                scanf("%s", d);

                for (i = 0; i < dcnt; i++) {

                    if (strcmp(d, dir[i].dname) == 0) {

                        printf("Enter name of the file -- ");

                        scanf("%s", f);

                        for (k = 0; k < dir[i].fcnt; k++) {

                            if (strcmp(f, dir[i].fname[k]) == 0) {

                                printf("File %s is deleted ", f);

                                dir[i].fcnt--;

                                strcpy(dir[i].fname[k], dir[i].fname[dir[i].fcnt]);

                                goto jmp;

                            }

                        }

                        printf("File %s not found", f);

                        goto jmp;

                    }

                }

                printf("Directory %s not found", d);

jmp :

                break;

            case 4:

                printf("\nEnter name of the directory -- ");

                scanf("%s", d);

                for (i = 0; i < dcnt; i++) {

                    if (strcmp(d, dir[i].dname) == 0) {

                        printf("Enter the name of the file -- ");

                        scanf("%s", f);

                        for (k = 0; k < dir[i].fcnt; k++) {

                            if (strcmp(f, dir[i].fname[k]) == 0) {

                                printf("File %s is found ", f);

                                goto jmp1;

                            }

                        }

                        printf("File %s not found", f);

jmp1 :

                        break;

                    }

                }

            case 5:

                if (dcnt == 0)

                    printf("\nNo Directory's ");

                else {

                    printf("\nDirectory\tFiles");

                    for (i = 0; i < dcnt; i++) {

                        printf("\n%s\t\t", dir[i].dname);

                        for (k = 0; k < dir[i].fcnt; k++)

                            printf("\t%s", dir[i].fname[k]);

                    }

                }

                break;

            default:

                exit(0);

        }

    }

}

OUTPUT:

1. Create Directory 2. Create File 3. Delete File

4. Search File 5. Display 6. Exit

Enter your choice: 1

Enter name of directory -- void

Directory created

1. Create Directory 2. Create File 3. Delete File

4. Search File 5. Display 6. Exit

Enter your choice: 1

Enter name of directory -- order

Directory created

1. Create Directory 2. Create File 3. Delete File

4. Search File 5. Display 6. Exit

Enter your choice: 2

Enter name of the directory -- void

Enter name of the file -- a1

File created

1. Create Directory 2. Create File 3. Delete File

4. Search File 5. Display 6. Exit

Enter your choice: 2

Enter name of the directory -- order

Enter name of the file -- b1

File created

1. Create Directory 2. Create File 3. Delete File

4. Search File 5. Display 6. Exit

Enter your choice: 5

Directory Files

void a1

order b1

1. Create Directory 2. Create File 3. Delete File

4. Search File 5. Display 6. Exit

Enter your choice: 4

Enter name of the directory -- void

Enter the name of the file -- a1

File a1 is found

Directory Files

void a1

order b1

1. Create Directory 2. Create File 3. Delete File

4. Search File 5. Display 6. Exit

Enter your choice: 3

Enter name of the directory -- order

Enter name of the file -- b1

File b1 is deleted

1. Create Directory 2. Create File 3. Delete File

4. Search File 5. Display 6. Exit

Enter your choice: 5

Directory Files

void a1

order

1. Create Directory 2. Create File 3. Delete File

4. Search File 5. Display 6. Exit

Enter your choice: 3

Enter name of the directory -- void

Enter name of the file -- a1

File a1 is deleted

1. Create Directory 2. Create File 3. Delete File

4. Search File 5. Display 6. Exit

Enter your choice: 5

Directory Files

void

order

1. Create Directory 2. Create File 3. Delete File

4. Search File 5. Display 6. Exit

Enter your choice: 6

PS C:\C Programs>

3.tree directory

Code:

#include <stdio.h>

#include <conio.h>

#include <graphics.h>

#include <stdlib.h>

#include <string.h>

struct tree\_element {

    char name[20];

    int x, y, ftype, lx, rx, nc, level;

    struct tree\_element \*link[5];

};

typedef struct tree\_element node;

void create(node \*\*root, int lev, char \*dname, int lx, int rx, int x);

void display(node \*root);

int main() {

    int gd = DETECT, gm;

    node \*root;

    root = NULL;

    clrscr();

    create(&root, 0, "root", 0, 639, 320);

    clrscr();

    initgraph(&gd, &gm, "C:\\Turboc3\\BGI");

    display(root);

    getch();

    closegraph();

    return 0;

}

void create(node \*\*root, int lev, char \*dname, int lx, int rx, int x) {

    int i, gap;

    char temp[20];

    if (\*root == NULL) {

        \*root = (node \*)malloc(sizeof(node));

        printf("Enter name of dir/file (under %s): ", dname);

        fflush(stdin);

        gets(temp);

        strncpy((\*root)->name, temp, 20);

        (\*root)->name[19] = '\0';

        printf("Enter 1 for Dir/ 2 for file: ");

        scanf("%d", &(\*root)->ftype);

        (\*root)->level = lev;

        (\*root)->y = 50 + lev \* 50;

        (\*root)->x = x;

        (\*root)->lx = lx;

        (\*root)->rx = rx;

        for (i = 0; i < 5; i++)

            (\*root)->link[i] = NULL;

        if ((\*root)->ftype == 1) {

            printf("No of sub directories/files (for %s): ", (\*root)->name);

            scanf("%d", &(\*root)->nc);

            if ((\*root)->nc == 0)

                gap = rx - lx;

            else

                gap = (rx - lx) / (\*root)->nc;

            for (i = 0; i < (\*root)->nc; i++)

                create(&((\*root)->link[i]), lev + 1, (\*root)->name, lx + gap \* i, lx + gap \* i + gap, lx + gap \* i + gap / 2);

        } else

            (\*root)->nc = 0;

    }

}

void display(node \*root) {

    int i;

    settextstyle(2, 0, 4);

    settextjustify(1, 1);

    setfillstyle(1, BLUE);

    setcolor(14);

    if (root != NULL) {

        for (i = 0; i < root->nc; i++) {

            line(root->x, root->y, root->link[i]->x, root->link[i]->y);

        }

        if (root->ftype == 1)

            bar3d(root->x - 20, root->y - 10, root->x + 20, root->y + 10, 0, 0);

        else

            fillellipse(root->x, root->y, 20, 20);

        outtextxy(root->x, root->y, root->name);

        for (i = 0; i < root->nc; i++) {

            display(root->link[i]);

        }

    }

}

OUTPUT:

Enter name of dir/file (under root): home

Enter 1 for Dir/ 2 for file: 1

No of sub directories/files (for home): 3

Enter name of dir/file (under home): user1

Enter 1 for Dir/ 2 for file: 1

No of sub directories/files (for user1): 3

Enter name of dir/file (under user1): documents

Enter 1 for Dir/ 2 for file: 1

No of sub directories/files (for documents): 2

Enter name of dir/file (under documents): file1

Enter 1 for Dir/ 2 for file: 2

Enter name of dir/file (under documents): file2

Enter 1 for Dir/ 2 for file: 2

Enter name of dir/file (under user1): pictures

Enter 1 for Dir/ 2 for file: 1

No of sub directories/files (for pictures): 1

Enter name of dir/file (under pictures): photo1

Enter 1 for Dir/ 2 for file: 2

Enter name of dir/file (under user1): music

Enter 1 for Dir/ 2 for file: 1

No of sub directories/files (for music): 1

Enter name of dir/file (under music): song1

Enter 1 for Dir/ 2 for file: 2

Enter name of dir/file (under home): user2

Enter 1 for Dir/ 2 for file: 1

No of sub directories/files (for user2): 2

Enter name of dir/file (under user2): projects

Enter 1 for Dir/ 2 for file: 1

No of sub directories/files (for projects): 2

Enter name of dir/file (under projects): project1

Enter 1 for Dir/ 2 for file: 2

Enter name of dir/file (under projects): project2

Enter 1 for Dir/ 2 for file: 2

Enter name of dir/file (under user2): downloads

Enter 1 for Dir/ 2 for file: 1

No of sub directories/files (for downloads): 1

Enter name of dir/file (under downloads): file1

Enter 1 for Dir/ 2 for file: 2

Enter name of dir/file (under home): public

Enter 1 for Dir/ 2 for file: 1

No of sub directories/files (for public): 1

Enter name of dir/file (under public): shared

Enter 1 for Dir/ 2 for file: 1

No of sub directories/files (for shared): 1

Enter name of dir/file (under shared): file1

Enter 1 for Dir/ 2 for file: 2

