SSN COLLEGE OF ENGINEERING, KALAVAKKAM (An Autonomous Institution, Affiliated to Anna University, Chennai) Department of Computer Science and Engineering UCS1411 – Operating Systems Laboratory II Year CSE - B Section ( IV Semester) Academic Year 2019-20

Exercise 13-File organization techniques

Name:Prathyush.S

Class:CSE-B

Register No.:185001112

***Program:-***

SingleLevel.c

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

typedef struct files {

char fname[10];

}File;

File\* root[50];

int fcount = 0;

File\* new\_file(char s[])

{

File\* n = (File\*)malloc(sizeof(File));

strcpy(n->fname, s);

return n;

}

int search\_file(char s[])

{

int flag = 0;

for(int i = 0; i < fcount; i++)

{

if(root[i] != NULL) {

if(strcmp(root[i]->fname,s) == 0) {

flag = 1;

break;

}

}

}

return flag;

}

void insert\_file(char s[])

{

if(search\_file(s) == 1) {

printf("File %s already exists!\n", s);

return;

}

root[fcount] = new\_file(s);

fcount++;

printf("Created!\n");

}

void display(File\* d[])

{

printf("Contents of root:\n");

if(fcount == 0) {

printf("Empty!\n");

return;

}

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

if(root[i] != NULL) {

printf("%s\t", root[i]->fname);

}

}

printf("\n");

}

int main()

{

int c;

while(1)

{

printf("1. New File\n");

printf("2. Display all files\n");

printf("3. Exit\n");

printf("Enter choice: ");

scanf("%d",&c);

if(c==1)

{

char s[50];

printf("Enter file name: ");

scanf("%s",s);

insert\_file(s);

}

else if(c==2)

{

display(root);

}

else {

break;

}

}

}

***OUTPUT***

1. New File

2. Display all files

3. Exit

Enter choice: 1

Enter file name: file1.txt

Created!

1. New File

2. Display all files

3. Exit

Enter choice: 1

Enter file name: file1.txt

File file1.txt already exists!

1. New File

2. Display all files

3. Exit

Enter choice: 1

Enter file name: file2.txt

Created!

1. New File

2. Display all files

3. Exit

Enter choice: 1

Enter file name: fil3# #e.txt

Created!

1. New File

2. Display all files

3. Exit

Enter choice: 1

Enter file name: file4.txt

Created!

1. New File

2. Display all files

3. Exit

Enter choice: 1

Enter file name: file5,# #.txt

Created!

1. New File

2. Display all files

3. Exit

Enter choice: 2

Contents of root:

file1.txt file2.txt file.txt file4.txt file5.txt

1. New File

2. Display all files

3. Exit

Enter choice: 3

***Program:-***

Tree.c

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

typedef struct files {

char fname[10];

}File;

typedef struct directory {

char dname[10];

struct directory \*d1,\*d2,\*d3;

File \*f1,\*f2;

}Directory;

Directory \*root = NULL;

void insert\_directory(char s[])

{

Directory\* temp=root;

char \*t = strtok(s,"/");

t = strtok(NULL,"/");

while(t != NULL) {

if(temp->d1 != NULL && strcmp(t,temp->d1->dname)==0)

{

temp = temp->d1;

}

else if(temp->d2 != NULL && strcmp(t,temp->d2->dname)==0)

{

temp=temp->d2;

}

else if(temp->d3 != NULL && strcmp(t,temp->d3->dname)==0)

{

temp=temp->d3;

}

t = strtok(NULL,"/");

}

if(t==NULL)

{

if(temp->d1==NULL||temp->d2==NULL||temp->d3==NULL)

{

char d[10];

printf("Enter the directory name: ");

scanf("%s",d);

Directory\* newdir = (Directory\*)malloc(sizeof(Directory));

newdir->d1=NULL;

newdir->d2=NULL;

newdir->d3=NULL;

newdir->f1=NULL;

newdir->f2=NULL;

strcpy(newdir->dname,d);

if(temp->d1 == NULL)

{

temp->d1 = newdir;

}

else if(temp->d2 == NULL && strcmp(d,temp->d1->dname)!=0)

{

temp->d2 = newdir;

}

else if(strcmp(d,temp->d1->dname) != 0 && strcmp(d,temp->d2->dname)!

=0)

{

temp->d3 = newdir;

}

else if(strcmp(d,temp->d1->dname) == 0 || strcmp(d,temp->d2-

>dname)==0)

printf("Duplicate directories not allowed!\n");

}

else printf("Directory limit exceeded!\n");

}

}

void insert\_file(char s[])

{

Directory\* temp=root;

char temp1[100];

strcpy(temp1,s);

char \*t = strtok(s,"/");

t = strtok(NULL,"/");

while(t != NULL) {

if(temp->d1 != NULL && strcmp(t,temp->d1->dname)==0)

{

temp = temp->d1;

}

else if(temp->d2 != NULL && strcmp(t,temp->d2->dname)==0)

{

temp = temp->d2;

}

else if(temp->d3 != NULL && strcmp(t,temp->d3->dname)==0)

{

temp = temp->d3;

}

t = strtok(NULL,"/");

}

if(t == NULL)

{

if(temp->f1 == NULL || temp->f2 == NULL)

{

char d[10];

printf("Enter the file name: ");

scanf("%s",d);

File \*newfile = (File\*)malloc(sizeof(File));

strcpy(newfile->fname,d);

if(temp->f1 == NULL)

{

temp->f1=newfile;

}

else if(temp->f2 == NULL)

{

temp->f2=newfile;

}

}

else

printf("File limit exceeded!");

}

}

File\* get\_file\_pointer(char s[])

{

char \*t = strtok(s,"/");

char \*g;

Directory \*temp = root;

while(t != NULL) {

if(temp->d1 != NULL && strcmp(t,temp->d1->dname)==0)

{

temp = temp->d1;

}

else if(temp->d2 != NULL && strcmp(t,temp->d2->dname)==0)

{

temp = temp->d2;

}

else if(temp->d3 != NULL && strcmp(t,temp->d3->dname)==0)

{

temp=temp->d3;

}

g = t;

t = strtok(NULL,"/");

if(t==NULL)

{

if(strcmp(temp->f1->fname,g)==0)

return temp->f1;

else if(strcmp(temp->f2->fname,g)==0)

return temp->f2;

else

{

printf("No such file!\n");

return NULL;

}

}

}

return NULL;

}

Directory\* get\_directory\_pointer(char s[])

{

char \*t = strtok(s,"/");

char \*g;

Directory \*temp = root;

while(t != NULL){

if(temp->d1 != NULL && strcmp(t,temp->d1->dname)==0)

{

temp = temp->d1;

}

else if(temp->d2 != NULL && strcmp(t,temp->d2->dname)==0)

{

temp = temp->d2;

}

else if(temp->d3 != NULL && strcmp(t,temp->d3->dname)==0)

{

temp = temp->d3;

}

g = t;

t = strtok(NULL,"/");

if(t == NULL)

{

return temp;

}

}

return NULL;

}

void display\_file(File\* f, char s[])

{

printf("%s\t\t%s\n",f->fname,s);

}

void display(Directory\* r, char s[])

{

if(r!=NULL)

{

strcat(s,r->dname);

strcat(s,"/");

if(r->f1 != NULL)

{

display\_file(r->f1,s);

}

if(r->f2!=NULL)

{

display\_file(r->f2,s);

}

if(r->d1 != NULL) {

char s1[50];

strcpy(s1, s);

display(r->d1,s1);

}

if(r->d2 != NULL) {

char s1[50];

strcpy(s1, s);

display(r->d2,s1);

}

if(r->d3 != NULL) {

char s1[50];

strcpy(s1, s);

display(r->d3,s1);

}

}

}

int main()

{

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

strcpy(root->dname,"root");

root->d1=NULL;

root->d2=NULL;

root->d3=NULL;

root->f1=NULL;

root->f2=NULL;

int c;

while(1)

{

printf("1. Insert a Directory\n");

printf("2. Insert a File\n");

printf("3. Display all files\n");

printf("4. Exit\n");

printf("Enter choice: ");

scanf("%d",&c);

if(c==1)

{

char s[50];

printf("Path format:\n");

printf("root/ (or) root - to insert in root\n");

printf("root/directory/ - to insert into directory in root\n");

printf("Enter the path: ");

scanf("%s",s);

insert\_directory(s);

}

else if(c==2)

{

char s[50];

printf("Path format:\n");

printf("root/ (or) root - to insert file in root\n");

printf("root/directory/ - to insert file into directory in root\n");

printf("Enter the path: ");

scanf("%s",s);

insert\_file(s);

}

else if(c==3)

{

char s[400];

strcpy(s,"");

printf("File\t\tPath\n");

display(root,s);

}

else {

break;

}

}

}

***Output:-***

Tree.c

1. Insert a Directory

2. Insert a File

3. Display all files

4. Exit

Enter choice: 1

Path format:

root/ (or) root - to insert in root

root/directory/ - to insert into directory in root

Enter the path: root

Enter the directory name: D1

1. Insert a Directory

2. Insert a File

3. Display all files

4. Exit

Enter choice: 1

Path format:

root/ (or) root - to insert in root

root/directory/ - to insert into directory in root

Enter the path: root

Enter the directory name: D2

1. Insert a Directory

2. Insert a File

3. Display all files

4. Exit

Enter choice: 1

Path format:

root/ (or) root - to insert in root

root/directory/ - to insert into directory in root

Enter the path: root

Enter the directory name: D3

1. Insert a Directory

2. Insert a File

3. Display all files

4. Exit

Enter choice: 1

Path format:

root/ (or) root - to insert in root

root/directory/ - to insert into directory in root

Enter the path: root

Directory limit exceeded!

1. Insert a Directory

2. Insert a File

3. Display all files

4. Exit

Enter choice: 1

Path format:

root/ (or) root - to insert in root

root/directory/ - to insert into directory in root

Enter the path: root/D1/

Enter the directory name: file2.txt

1. Insert a Directory

2. Insert a File

3. Display all files

4. Exit

Enter choice: 3

File Path

1. Insert a Directory

2. Insert a File

3. Display all files

4. Exit

Enter choice: 4