```
//OS-FILE ORGANIZATION TECHNIQUES IN C
//SINGLE LEVEL DIRECTORY
#include<stdio.h>
#include <string.h>
#include <stdlib.h>
typedef struct filee
{
char fname[10];
int loc;
}file;
typedef struct directory
{
char dname[10];
int count;
file f[10];
}direc;
void main()
{
int i,ch;
int copy;
char f[30];
direc dir;
dir.count = 0;
printf("\nEnter name of directory : ");
scanf("%s", dir.dname);
while(1)
{
printf("\n\n 1. Create File\t2. Delete File\t3. Search File \n 4. Display Files\t5. Exit\nEnter your choice :
");
scanf("%d",&ch);
switch(ch)
{
case 1: printf("\n Enter the name of the file : ");
int fcount=dir.count;
copy=0;
if(dir.count==0)
{
scanf("%s",dir.f[dir.count].fname);
dir.f[dir.count].loc=10;
dir.count++;
}
else
{
scanf("%s",f);
for(i=0;i<dir.count;i++)
{
if(strcmp(f,dir.f[i].fname)==0)
{
copy=1;
printf("FILE ALREADY EXISTS!");
break;
}
}
if(copy!=1)
{
strcpy(dir.f[fcount].fname,f);
dir.f[fcount].loc=dir.f[fcount-1].loc + 15;
dir.count++;
break;
}
}
break;
case 2: printf("\n Enter the name of the file : ");
scanf("%s",f);
for(i=0;i<dir.count;i++)
{
if(strcmp(f, dir.f[i].fname)==0)
{
printf("File %s is deleted ",f);
strcpy(dir.f[i].fname,dir.f[dir.count-1].fname);
break;
}
}
if(i==dir.count)
printf("File %s not found",f);
else
dir.count--;
break;
case 3: printf("\n Enter the name of the file : ");
scanf("%s",f);
for(i=0;i<dir.count;i++)
{
if(strcmp(f, dir.f[i].fname)==0)
{
printf("File %s is found ", f);
break;
}
}
if(i==dir.count)
printf("File %s not found",f);
break;
case 4: if(dir.count==0)
printf("\n Directory Empty");
else
{
printf("\n The Files are : ");
printf("\nFile\tLocation");
for(i=0;i<dir.count;i++)
printf("\n%s\t%d",dir.f[i].fname,dir.f[i].loc);
}
break;
default: exit(0);
}
}
}
//TWO LEVEL DIRECTORIES
#include<stdio.h>
#include<string.h>
```

```
#include<stdlib.h>
struct
{
char dname[10],fname[10][10];
int fcount;
}dir[10];
void main()
{
int i,ch,dcount,k;
char f[30], d[30];
dcount=0;
while(1)
{
printf("\n\n 1. Create Directory\n 2. Create File\n 3. Delete File");
printf("\n 4. Search File \n 5. Display \n 6. Exit \n Enter your choice:");
scanf("%d",&ch);
switch(ch)
{
case 1: printf("\n Enter name of directory:");
scanf("%s", dir[dcount].dname);
dir[dcount].fcount=0;
dcount++;
printf("\nDirectory created");
break;
case 2: printf("\n Enter name of the directory:");
scanf("%s",d);
for(i=0;i<dcount;i++)
if(strcmp(d,dir[i].dname)==0)
{
printf("Enter name of the file:");
scanf("%s",dir[i].fname[dir[i].fcount]);
dir[i].fcount++;
printf("\nFile created");
break;
}
if(i==dcount)
printf("\nDirectory %s not found",d);
break;
case 3: printf("\nEnter name of the directory:");
scanf("%s",d);
for(i=0;i<dcount;i++)
{
if(strcmp(d,dir[i].dname)==0)
{
printf("\nEnter name of the file:");
scanf("%s",f);
for(k=0;k<dir[i].fcount;k++)
{
if(strcmp(f, dir[i].fname[k])==0)
{
printf("\nFile %s is deleted ",f);
dir[i].fcount--;
strcpy(dir[i].fname[k],dir[i].fname[dir[i].fcount]);
goto jmp;
}
}
printf("\nFile %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<dcount;i++)
{
if(strcmp(d,dir[i].dname)==0)
{
printf("Enter the name of the file:");
scanf("%s",f);
for(k=0;k<dir[i].fcount;k++)
{
if(strcmp(f, dir[i].fname[k])==0)
{
printf("\nFile %s is found ",f);
goto jmp1;
}
}
printf("\nFile %s not found",f);
goto jmp1;
}
}
printf("\nDirectory %s not found",d);
jmp1: break;
case 5: if(dcount==0)
printf("\nNo Directory's ");
else
{
printf("\nDirectory\tFiles");
for(i=0;i<dcount;i++)
{
printf("\n%s\t",dir[i].dname);
for(k=0;k<dir[i].fcount;k++)
printf("\t%s",dir[i].fname[k]);
}
}
break;
default:exit(0);
}
}
}
//TREE_STRUCTURE DIRECTORY
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
struct tree
{
char name[25]; //name of dir/file
```

```
int df; //dir-1/file-0
struct tree *main; //main directory
struct tree *sub; //subdirectories
};
struct tree *root[20];
int in=0,count=0;
void create(struct tree *P,int N) //N is the number of child nodes
{
int i;
struct tree *Temp,*T;
Temp=P;
for(i=0;i<N;i++)
{
T=malloc(sizeof(struct tree));
printf("\nEnter name: ");
scanf("%s",T->name);
printf("\nEnter dir(1) or file(0): ");
scanf("%d",&T->df);
if(T->df==1)
{
root[count]=T;
count++;
}
T->main=NULL;
T->sub=NULL;
if(i==0)
{
Temp->main=T;
Temp=T;
}
else
{
Temp->sub=T;
Temp=T;
}
}
}
void display(struct tree *P)
{
int i;
P=P->main;
i=0;
do{
if(P->df==1)
{
printf("\nDIR:\t");
}
printf("%s(%d)\n\t\t",P->name,P->df);
if(P->df==1 && P->main!=NULL)
display(P);
P=P->sub;
i++;
}while(P!=NULL);
}
```

```
void main()
{
int nu,nc;
int i,j,k;
struct tree *tree_dir;
tree_dir=malloc(sizeof(struct tree));
tree_dir->df=1;
tree_dir->main=NULL;
tree_dir->sub=NULL;
create(tree_dir,1);
for(in=0;in<count;in++)
{
printf("\n Enter number of child nodes for %s:",root[in]->name);
scanf("%d",&nc);
create(root[in],nc);
}
printf("\n--Tree Structure Directory--\n(1-Directory\t\t0-File)\n");
display(tree_dir);
}
//DAG DIRECTORY ORGANISATION
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
typedef struct trees
{
char from_d[20];
char file[20];
char to[20];
int flag;
}linkk;
struct tree
{
char name[25]; //name of dir/file
int df; //dir-1/file-0
struct tree *main; //main directory
struct tree *sub; //sub-directories
linkk link; //link
};
struct tree *root[20];
int in=0,count=0;
void create(struct tree *P,int N) //N is the number of child nodes
{
int i;
struct tree *Temp,*T;
Temp=P;
for(i=0;i<N;i++)
{
T=malloc(sizeof(struct tree));
printf("\nEnter name: ");
scanf("%s",T->name);
printf("\nEnter dir(1) or file(0): ");
scanf("%d",&T->df);
if(T->df==1)
{
```

```
root[count]=T;
count++;
}
T->main=NULL;
T->sub=NULL;
if(i==0)
{
Temp->main=T;
Temp=T;
}
else
{
Temp->sub=T;
Temp=T;
}
}
}
void links(struct tree *P)
{
char src[10],dest[20],f[20];
printf("Enter destination directory:");
scanf("%s",dest);
printf("Enter source directory:");
scanf("%s",src);
printf("Enter the filename:");
scanf("%s",f);
P=(P->main)->main;
while(strcmp(P->name,dest)!=0)
P=P->sub;
if(strcmp(P->name,dest)==0)
{
P->link.flag=1;
strcpy(P->link.from_d,src);
strcpy(P->link.to,dest);
strcpy(P->link.file,f);
}
}
void display(struct tree *P)
{
int i;
P=P->main;
i=0;
do{
if(P->df==1)
{
printf("\nDIR:\t");
}
printf("%s(%d)\n\t\t",P->name,P->df);
if(P->link.flag==1)
printf("\n\t\tLink:%s(0)-From dir:%s\n\t\t",P->link.file,P->link.from_d);
if(P->df==1 && P->main!=NULL)
display(P);
P=P->sub;
i++;
```

```
}while(P!=NULL);
}
void main()
{
int nc;
int i,j,k;
int ch;
struct tree *tree_dir;
tree_dir=malloc(sizeof(struct tree));
tree_dir->df=1;
tree_dir->main=NULL;
tree_dir->sub=NULL;
create(tree_dir,1);
printf("\n1.Create fil/dir\n2.Create link\n");
scanf("%d",&ch);
do{
if(ch==1)
{
for(in=0;in<count;in++)
{
printf("\n Enter number of child nodes for %s:",root[in]->name);
scanf("%d",&nc);
create(root[in],nc);
}
printf("\n--Tree Structure Directory--\n(1-Directory\t\t0-File)\n");
display(tree_dir);
}
else if(ch==2)
{
links(tree_dir);
printf("\n--Tree Structure Directory--\n(1-Directory\t\t0-File)\n");
display(tree_dir);
}
else
exit(0);
printf("\n\n1.Create fil/dir\n2.Create link\n");
scanf("%d",&ch);
}while(ch==1||ch==2);
}
/*OUTPUT SINGLE LEVEL DIRECTORY
C:\Users\GAYU\Desktop>gcc -o n singlelevel.c
C:\Users\GAYU\Desktop>n
Enter name of directory : OS
1. Create File
2. Delete File
3. Search File
4. Display Files
5. Exit
Enter your choice : 1
Enter the name of the file : u1_os
1. Create File
2. Delete File
3. Search File
4. Display Files
```

```
5. Exit
Enter your choice : 1
Enter the name of the file : u2_os
1. Create File
2. Delete File
3. Search File
4. Display Files
5. Exit
Enter your choice : 1
Enter the name of the file : u1_os
FILE ALREADY EXISTS!
1. Create File
2. Delete File
3. Search File
4. Display Files
5. Exit
Enter your choice : 4
The Files are :
File Location
u1_os 10
u2_os 25
1. Create File
2. Delete File
3. Search File
4. Display Files
5. Exit
Enter your choice : 2
Enter the name of the file : u2_os
File u2_os is deleted
1. Create File
2. Delete File
3. Search File
4. Display Files
5. Exit
Enter your choice : 4
The Files are :
File Location
u1_os 10
1. Create File
2. Delete File
3. Search File
4. Display Files
5. Exit
Enter your choice : 1
Enter the name of the file : u3_os
1. Create File
2. Delete File
3. Search File
4. Display Files
5. Exit
Enter your choice : 3
Enter the name of the file : u1_os
File u1_os is found
1. Create File
```

```
2. Delete File
3. Search File
4. Display Files
5. Exit
Enter your choice : 4
The Files are :
File Location
u1_os 10
u3_os 25
1. Create File
2. Delete File
3. Search File
4. Display Files
5. Exit
Enter your choice : 5
C:\Users\GAYU\Desktop> */
/*OUTPUT-TWO LEVEL DIRECTORY
C:\Users\GAYU\Desktop>n
1. Create Directory
2. Create File
3. Delete File
4. Search File
5. Display
6. Exit
Enter your choice:1
Enter name of directory:OS
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:OS
Enter name of the file:unit1_os
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:OS
Enter name of the file:unit2_os
File 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:COA
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:COA
Enter name of the file:unit1_coa
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:COA
Enter name of the file:unit2_coa
File created
1. Create Directory
2. Create File
3. Delete File
4. Search File
5. Display
6. Exit
Enter your choice:5
Directory Files
OS unit1_os unit2_os
COA unit1_coa unit2_coa
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:OS
Enter name of the file:unit3_os
File created
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:OS
Enter the name of the file:unit3_os
File unit3_os is found
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:COA
Enter name of the file:unit2_coa
File unit2_coa is deleted
1. Create Directory
2. Create File
3. Delete File
4. Search File
5. Display
6. Exit
Enter your choice:5
Directory Files
OS unit1_os unit2_os unit3_os
COA unit1_coa
1. Create Directory
2. Create File
3. Delete File
4. Search File
5. Display
6. Exit
Enter your choice:6
C:\Users\GAYU\Desktop>
*/
/*OUTPUT TREE STRUCTURED DIRECTORIES
C:\Users\GAYU\Desktop>gcc -o n treelevel.c
C:\Users\GAYU\Desktop>n
Enter name: GAYU
Enter dir(1) or file(0): 1
Enter number of child nodes for GAYU:2
Enter name: OS
Enter dir(1) or file(0): 1
Enter name: COA
Enter dir(1) or file(0): 1
Enter number of child nodes for OS:2
Enter name: unit1_OS
Enter dir(1) or file(0): 1
Enter name: unit2_OS
Enter dir(1) or file(0): 1
Enter number of child nodes for COA:2
Enter name: unit1_COA
Enter dir(1) or file(0): 1
Enter name: unit2_COA
Enter dir(1) or file(0): 1
Enter number of child nodes for unit1_OS:2
Enter name: chap1_unit1_OS
Enter dir(1) or file(0): 0
Enter name: chap2_unit1_OS
Enter dir(1) or file(0): 0
Enter number of child nodes for unit2_OS:2
Enter name: chap1_unit2_OS
Enter dir(1) or file(0): 0
Enter name: chap2_unit2_OS
Enter dir(1) or file(0): 0
Enter number of child nodes for unit1_COA:2
Enter name: chap1_unit1_COA
Enter dir(1) or file(0): 0
Enter name: chap2_unit1_COA
Enter dir(1) or file(0): 0
Enter number of child nodes for unit2_COA:2
Enter name: chap1_unit2_COA
Enter dir(1) or file(0): 0
Enter name: chap2_unit2_COA
Enter dir(1) or file(0): 0
--Tree Structure Directory--
(1-Directory 0-File)
DIR: GAYU(1) //ROOT
DIR: OS(1) //DIR 1
DIR: unit1_OS(1) //SUBDIR 1 under DIR 1
chap1_unit1_OS(0) // FILE under SUBDIR 1
chap2_unit1_OS(0) //FILE under SUBDIR 1
DIR: unit2_OS(1) //SUBDIR 2 under DIR 1
chap1_unit2_OS(0) //FILE under SUBDIR 2
chap2_unit2_OS(0) //FILE under SUBDIR 2
DIR: COA(1) //DIR 2
DIR: unit1_COA(1) //SUBDIR 1 under DIR 2
chap1_unit1_COA(0) //FILE under SUBDIR 1
chap2_unit1_COA(0) //FILE under SUBDIR 1
DIR: unit2_COA(1) //SUBDIR 2 under DIR 2
chap1_unit2_COA(0) //FILE under SUBDIR 2
chap2_unit2_COA(0) //FILE under SUBDIR 2
*/
/*OUTPUT DAG DIRECTORY
C:\Users\GAYU\Desktop>gcc -o n trydag.c
C:\Users\GAYU\Desktop>n
Enter name: GAYU
Enter dir(1) or file(0): 1
1.Create fil/dir
2.Create link
1
Enter number of child nodes for GAYU:3
Enter name: OS
Enter dir(1) or file(0): 1
Enter name: COA
Enter dir(1) or file(0): 1
Enter name: DS
Enter dir(1) or file(0): 1
Enter number of child nodes for OS:2
Enter name: intro
Enter dir(1) or file(0): 0
Enter name: unit1_OS
Enter dir(1) or file(0): 1
Enter number of child nodes for COA:2
Enter name: intro
Enter dir(1) or file(0): 0
Enter name: unit1_COA
Enter dir(1) or file(0): 1
Enter number of child nodes for DS:2
Enter name: intro
Enter dir(1) or file(0): 0
Enter name: unit1_DS
Enter dir(1) or file(0): 1
Enter number of child nodes for unit1_OS:2
Enter name: chap1_unit1_OS
Enter dir(1) or file(0): 0
Enter name: chap2_unit1_OS
Enter dir(1) or file(0): 0
Enter number of child nodes for unit1_COA:2
Enter name: chap1_unit1_COA
Enter dir(1) or file(0): 0
Enter name: chap2_unit1_COA
Enter dir(1) or file(0): 0
Enter number of child nodes for unit1_DS:2
Enter name: chap1_unit1_DS
Enter dir(1) or file(0): 0
Enter name: chap2_unit1_DS
Enter dir(1) or file(0): 0
--Tree Structure Directory--
(1-Directory 0-File)
DIR: GAYU(1)
DIR: OS(1)
 intro(0)
DIR: unit1_OS(1)
 chap1_unit1_OS(0)
 chap2_unit1_OS(0)
DIR: COA(1)
 intro(0)
DIR: unit1_COA(1)
 chap1_unit1_COA(0)
 chap2_unit1_COA(0)
DIR: DS(1)
 intro(0)
DIR: unit1_DS(1)
 chap1_unit1_DS(0)
 chap2_unit1_DS(0)
1.Create fil/dir
2.Create link
2
Enter destination directory:OS
Enter source directory:unit1_COA
Enter the filename:chap1_unit1_COA
--Tree Structure Directory--
(1-Directory 0-File)
DIR: GAYU(1)
DIR: OS(1)
 Link:chap1_unit1_COA(0)-From dir:unit1_COA
 intro(0)
DIR: unit1_OS(1)
 chap1_unit1_OS(0)
 chap2_unit1_OS(0)
DIR: COA(1)
 intro(0)
DIR: unit1_COA(1)
 chap1_unit1_COA(0)
 chap2_unit1_COA(0)
DIR: DS(1)
 intro(0)
DIR: unit1_DS(1)
 chap1_unit1_DS(0)
 chap2_unit1_DS(0)
1.Create fil/dir
2.Create link
0
C:\Users\GAYU\Desktop> */
```