PROGRAM 8

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

DESCRIPTION:

a) SINGLE LEVEL DIRECTORY:

The directory structure is the organization of files into a hierarchy of folders. In a single-level directory system, all the files are placed in one directory. There is a root directory which has all files. It has a simple architecture and there are no sub directories. Advantage of single level directory system is that it is easy to find a file in the directory.

PROGRAM

```
#include<stdlib.h>
#include<string.h>
#include<stdio.h>
struct
{
char dname[10],fname[10][10];
int fcnt;
}dir;
void main()
{
int i,ch;
char f[30];
dir.fcnt = 0;
printf("\nEnter name of directory -- ");
scanf("%s", dir.dname);
while(1)
{
printf("\n\n1. 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("\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);
```

```
strcpy(dir.fname[i],dir.fname[dir.fcnt-1]); break; } }
if(i==dir.fcnt) printf("File %s not found",f);
else
dir.fcnt--;
break;
case 3: 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);
break;
}
}
if(i==dir.fcnt)


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 -- dir1

1. Create File  2. Delete File  3. Search File 4. Display Files       5. Exit
Enter your choice -- 1

Enter the name of the file -- file1

1. Create File  2. Delete File  3. Search File 4. Display Files       5. Exit
Enter your choice -- 1

Enter the name of the file -- file2

1. Create File  2. Delete File  3. Search File 4. Display Files       5. Exit
Enter your choice -- 4
```

```
The Files are --        file1    file2

1. Create File  2. Delete File  3. Search File  4. Display Files       5. Exit
Enter your choice -- 3

Enter the name of the file -- file2
File file2 is 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 -- file2
File file2 is deleted

1. Create File  2. Delete File  3. Search File  4. Display Files       5. Exit
Enter your choice -- 3

Enter the name of the file -- file2
File file2 not found

1. Create File  2. Delete File  3. Search File  4. Display Files       5. Exit
Enter your choice -- 4

The Files are --        file1

1. Create File  2. Delete File  3. Search File  4. Display Files       5. Exit
Enter your choice – 5
```

DESCRIPTION:

b) TWO LEVEL DIRECTORY

In the two-level directory system, each user has own user file directory (UFD). The system maintains a master block that has one entry for each user. This master block contains the addresses of the directory of the users. When a user job starts or a user logs in, the system's master file directory (MFD) is searched. When a user refers to a particular file, only his own UFD is searched.

PROGRAM:

```
#include<stdio.h>

struct st
{
char dname[10];
char sdname[10][10];
char fname[10][10][10];
```

```
int ds,sds[10];
}dir[10];
void main()
{
int i,j,k,n;
char name[10];

printf("enter number of users:");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("enter user directory %d names:",i+1);
scanf("%s",&dir[i].dname);
printf("enter size of directories:");
scanf("%d",&dir[i].ds);
for(j=0;j<dir[i].ds;j++)
{
printf("enter subdirectory name and size:");
scanf("%s",&dir[i].sdname[j]);
scanf("%d",&dir[i].sds[j]);
for(k=0;k<dir[i].sds[j];k++)
{
printf("enter file name:");
scanf("%s",&dir[i].fname[j][k]);
}
}
}
printf("\ndirname\t\tsize\tsubdirname\tsize\tfiles");
printf("\n*************************************************\n");
for(i=0;i<n;i++)
{
printf("%s\t\t%d",dir[i].dname,dir[i].ds);
for(j=0;j<dir[i].ds;j++)
{
printf("\t%s\t\t%d\t",dir[i].sdname[j],dir[i].sds[j]);
for(k=0;k<dir[i].sds[j];k++)

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

}
```

OUTPUT

```
enter number of users:2
enter user directory 1 names:user1
enter size of directories:2
enter subdirectory name and size:dir1
```

```
2
enter file name:f1
enter file name:f2
enter subdirectory name and size:dir2
3
enter file name:f1
enter file name:f2
enter file name:f3
enter user directory 2 names:user2
enter size of directories:3
enter subdirectory name and size:dir1
2
enter file name:f1
enter file name:f3
enter subdirectory name and size:dir2
2
enter file name:f3
enter file name:f4
enter subdirectory name and size:dir3
4
enter file name:fi
enter file name:f2
enter file name:f3
enter file name:f4

dirname         size   subdirname    size   files
**************************************************
user1           2      dir1          2      f1   f2
                       dir2          3      f1   f2   f3

user2           3      dir1          2      f1   f3
                       dir2          2      f3   f4
                       dir3          4      fi   f2   f3   f4
```