MODEL INSTITUTE OF ENGINEERING AND TECHNOLOGY(MIET),JAMMU

Department of Computer Engineering

Group Members:- 1) Sania Fotedar(2021a1r082)
[GROUP REPRESENTATIVE]
2)Madhu Bala(2021a1r077)
3)Amisha Gurndwal (2021a1r062)
4)Rachna Sharma(2021a1r078)
5)Arjun Kapoor (2021a1r061)

Project Guide: Ms. Swati Goyal

FILE ORGANIZATION TECHNIQUES

FILE ORGANIZATION

File organization refers to the manner in which the records of a file are arranged on secondary storage. The most popular file organization scheme in use today follows.

sequential – Records are placed in physical order. The “next” record is the one that physically follows the previous record. This organization is natural for files stored on magnetic tape, an inherently sequential medium.

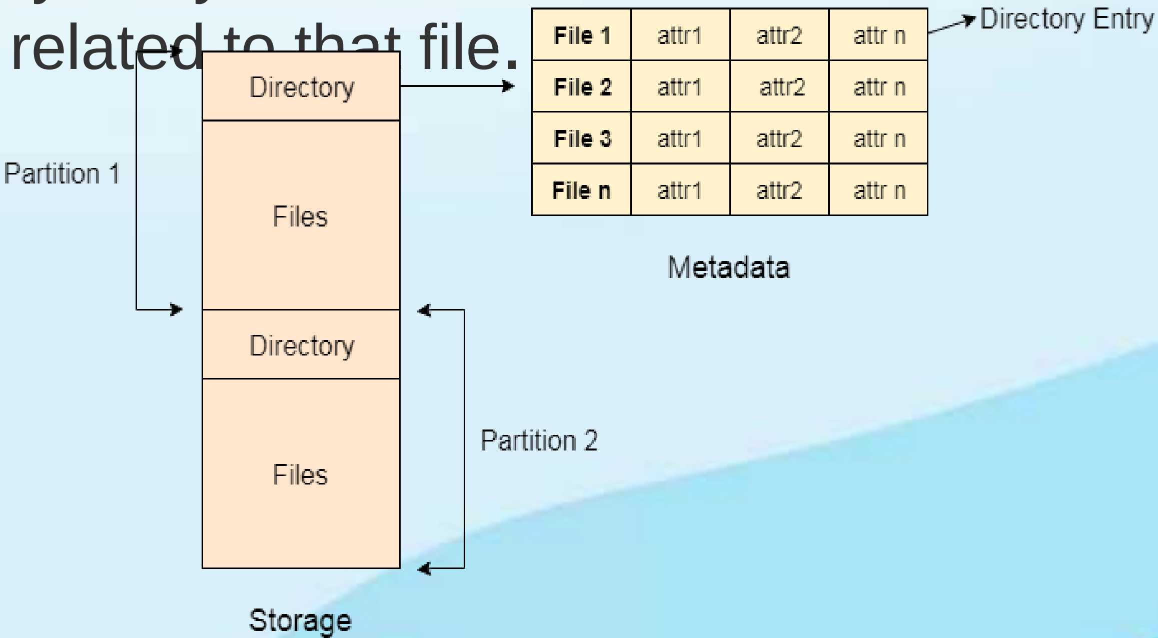
Direct – records are directly (randomly) accessed by their physical addresses on a direct access storage device (DASD).

Indexed sequential – records are arranged in logical sequence according to a key contained in each record. Indexed sequential records may be accessed sequentially in key order or they may be accessed directly.

Partitioned – This is essentially a file of sequential subfiles. Each sequential subfile is called a member. The starting address of each member is stored in the file’s directory.

DIRECTORY STRUCTURE

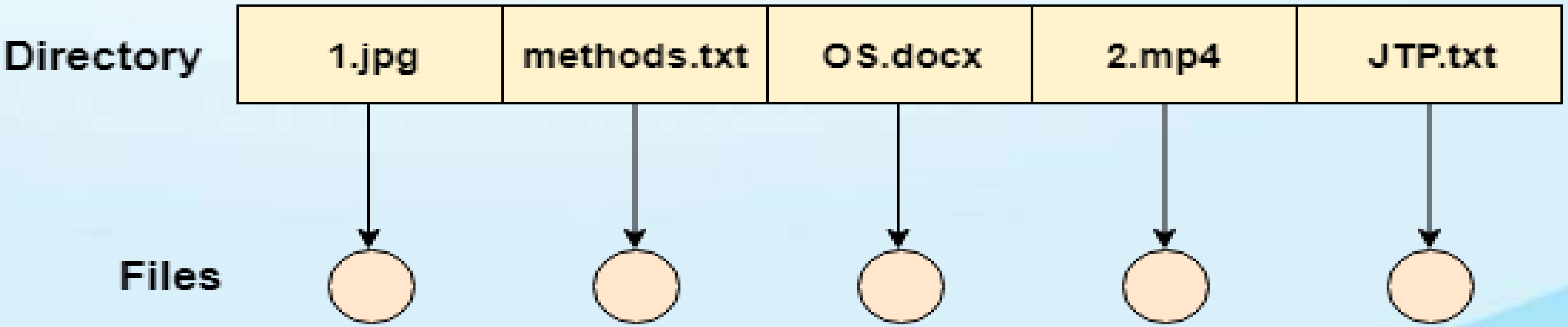
- Directory can be defined as the listing of the related files on the disk. The directory may store some or the entire file attributes.
- To get the benefit of different file systems on the different operating systems, A hard disk can be divided into the number of partitions of different sizes. The partitions are also called volumes or mini disks.
- Each partition must have at least one directory in which, all the files of the partition can be listed. A directory entry is maintained for each file in the directory which stores all the information related to that file.



File 1	attr1	attr2	attr n
File 2	attr1	attr2	attr n
File 3	attr1	attr2	attr n
File n	attr1	attr2	attr n

SINGLE LEVEL DIRECTORY

- The simplest method is to have one big list of all the files on the disk. The entire system will contain only one directory which is supposed to mention all the files present in the file system. The directory contains one entry per each file present on the file system.



Single Level Directory

ADVANTAGES AND DISADVANTAGES

Advantages

1. Implementation is very simple.
2. If the sizes of the files are very small then the searching becomes faster.
3. File creation, searching, deletion is very simple since we have only one directory.

Disadvantages

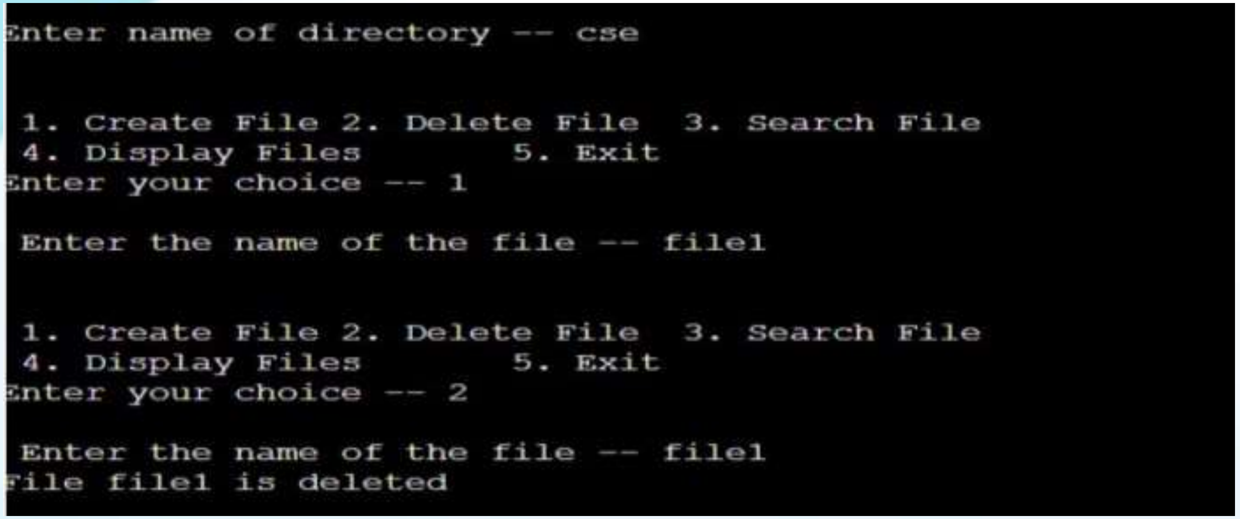
1. We cannot have two files with the same name.
2. The directory may be very big therefore searching for a file may take so much time.
3. Protection cannot be implemented for multiple users.
4. There are no ways to group same kind of files.
5. Choosing the unique name for every file is a bit complex and limits the number of files in the system because most of the Operating System limits the number of characters used to construct the file name

Program to Simulate Single level directory file organization technique

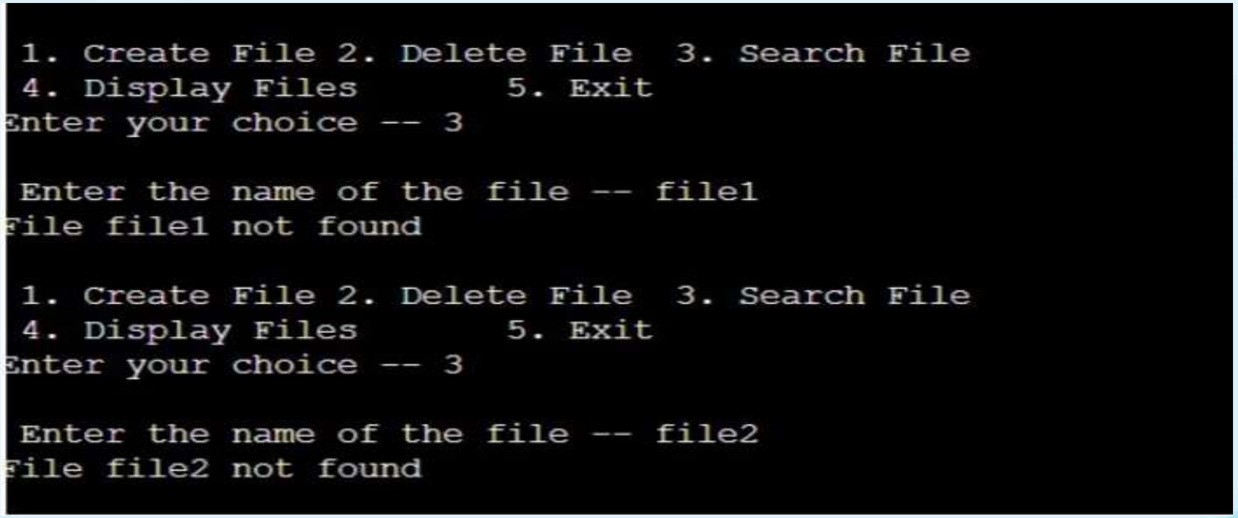
```
#include<stdio.h>
#Include<string.h>
#include<stdlib.h>
struct
{
char dname[10], fname[10][10];
int fcnt;
}dir;
void main()
{
int i,ch;
char f[30];
dir.fcnt = 0;
Printf ("\n Enter name of directory-- ");
Scanf ("%s", dir.dname);
while(1)
```

```
scanf("%d",&ch);
switch(ch)
{
case 1: printf("\n Enter the name of the file -- ");
scanf (“%s",dir.fname[dir.fcnt]);
dir.fcnt++;
break;
case 2: printf ("\n Enter 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("\n Enter 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("\n Directory Empty");
else
{
printf("\n The Files are -- ");
for(i=0;i<dir.fcnt;i++)
printf("\t%s",dir.fname[i]);
}
break;
default: exit(0);
}
}
}
```

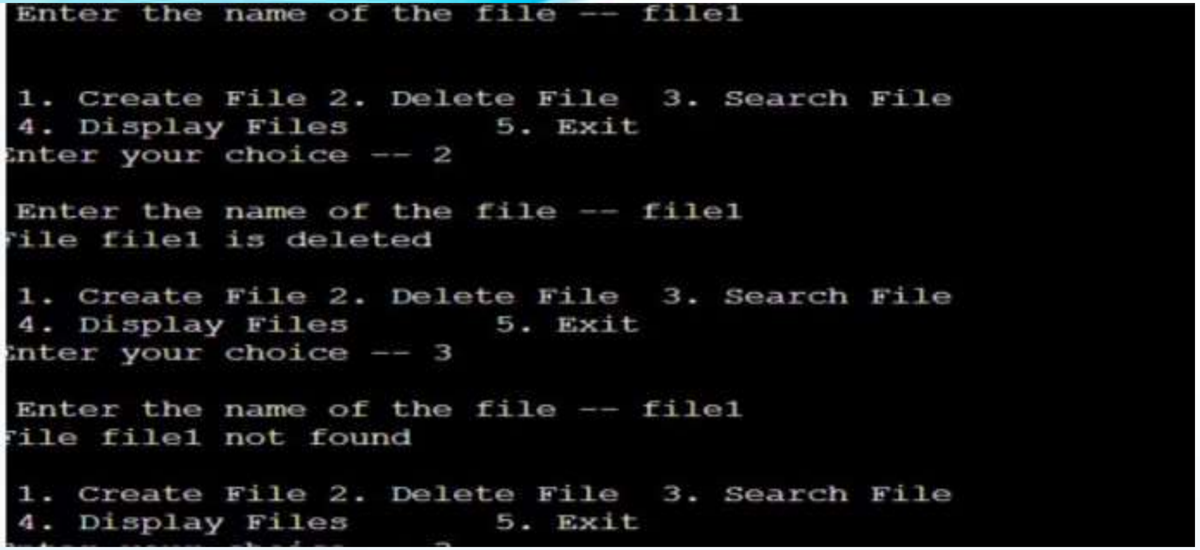
OUTPUT:-



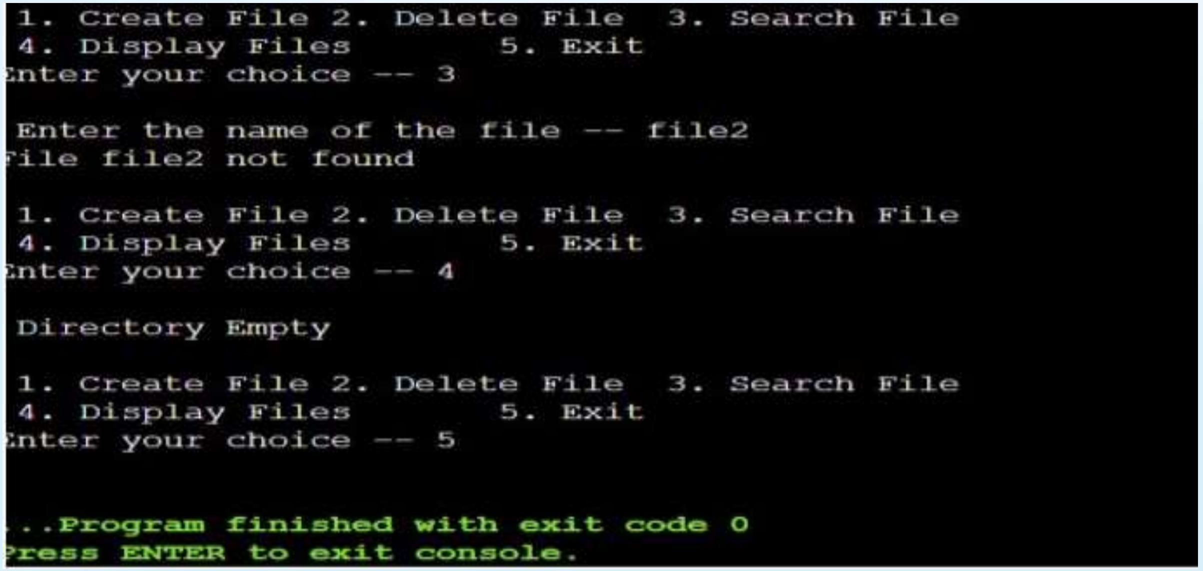
```
Enter name of directory -- cse


 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 -- 2

 Enter the name of the file -- file1
File file1 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 -- file1
File file1 not 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 -- file2
File file2 not found
```



```
 Enter the name of the file -- file1


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

 Enter the name of the file -- file1
File file1 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 -- file1
File file1 not found

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



```
 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

 Directory Empty

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


...Program finished with exit code 0
Press ENTER to exit console.
```

THANK YOU