Ex. No.: 12
Date: 23/4/25

AIM: **File Organization Technique- Single and Two level directory**

To implement File Organization Structures in C are
a. Single Level Directory
b. Two-Level Directory
c. Hierarchical Directory Structure
d. Directed Acyclic Graph Structure

a. Single Level

Directory

ALGORITHM

1. Start
2. Declare the number, names and size of the directories and file names.
3. Get the values for the declared variables.
4. Display the files that are available in the directories.
5. Stop.

PROGRAM:

```
# include <stdio.h>
# include <stdlib.h>
# include <graphics.h>
void main ()
{ int gd = DETECT, gm, count, i, j, num, x1, x;
  char fname[10][20];
  int graphics (&gd, &gm, "C:\\Turbo C3\\BGI");
  clear device ();
  set background color (Green);
  puts ("Enter the no. of files");
  scanf ("%d", &count);
```

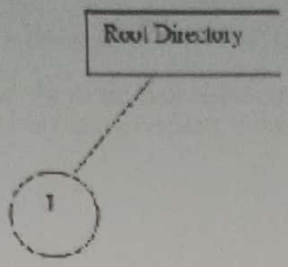
```
for (i=0; i< count ; i++)
  {
     cleardevice ();
     setBKcolor (GREEN);
     printf ("Enter the file %d name : ", i+1);
     scanf ("%s", fname(i));
     setfillstyle (1, MAGENTA);
     mid = 640/count;
     cu_x = mid/3;
     bar3d (270, 100, 370, 150, 0, 0);
     for (j=0; j<=i; j++, cu_x += mid)
        {
           line (320, 150, cu_x, 250);
           fillellipse (cu_x, 250, 30, 30);
           outtextxy (cu_x, 250, fname(j));
        }
  }
  getch ();
  closegraph ();
}
```

OUTPUT:

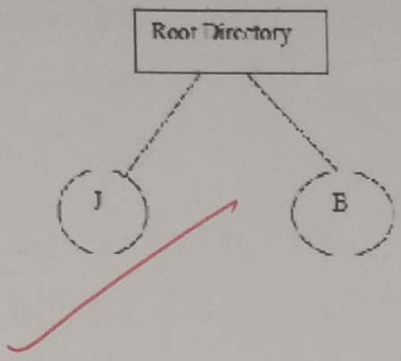
Enter the Number of files

2

Enter the file1 J



Enter the file2 B



b. Two-level directory Structure

ALGORITHM:

1. Start
2. Declare the number, names and size of the directories and subdirectories and file names.
3. Get the values for the declared variables.
4. Display the files that are available in the directories and subdirectories.
5. Stop.

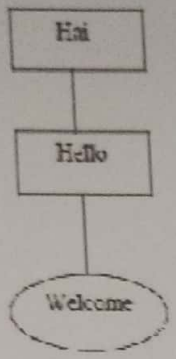
PROGRAM:

```
# include <stdio.h>
# include <stdlib.h>
# include <graphics.h>
void main ()
{
    int gd = DETECT, gm;
    int users, files[10], i, j, uid, uid_user, limit;
    char uname[10][20], fname[10][10][20];
    int it graph1/&gd, &gm, "C:\\Turbo3\\BGI");
    scanf("%d", &users);
    for(i=0; i<users; i++)
    {
        scanf("%s", uname[i]);
        scanf("%d", &files[i]);
        for(j=0; j<files[i]; j++)
        {
            scanf("%s", fname[i][j]);
        }
    }
```

```
cleardevice();
setbkcolor(GREEN);
setfillstyle(1, MAGENTA);
bar3d(370, 100, 370, 150, 0, 0);
setlinestyle(2, 0, 4);
mid = 640/users;
mid_user = mid/3;
for(i=0; cir_x = 50; i<users; i++, cir_x += mid)
{
    for(j=0; j<filled(i); j++)
    {
        int fil_x = cir_x - ((filled(i)-1)*50/2)
                    + (j*50);
        line(cir_x, 230, fil_x, 270);
        fillellipse(fil_x, 270, 20, 20);
    }
}
getch();
closegraph();
}
```

Sample Output:

```
Enter the name of dir/file(under null): Hai
How many users(for Hai):1
Enter name of dir/file(under Hai):Hello
How many files(for Hello):1
Enter name of dir/file(under Hello):welcome
```



Result:

Thus the code for implementing file structure is executed successfully.