Ex. No.: 12
Date: 24/4/25

File Organization Technique- Single and Two level directory

AIM:

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 <GL/glut.h>
#include <string.h>
#include <math.h>
#define Max_files 10
 char filename[Max_files][20];
 int file Count =0;
 float file x[Max_files];
```

```
void drawCircle (float x, float y, float rad){
        int i;
        int triAmt =100;
        float angle ;
        gl Begin(GL_TRIANGLE_FAN);
        gl Vertex2f (x,y);
        for (c= string; *c != '\0'; c++)
        {
            glutBitmapCharacter(font, *c);
        }
    }

void display (){
    glClear (GL_COLOR_BUFFER_BIT);
   glColor3f (1.0, 0.0, 1.0);
   gl Begin (GL_QUADS);
         glVertex2f (290,400);
         glVertex 2f (390, 400);
         glVertex 2f ( 390, 450);
         glVertex 2f (290 ,450);
      glEnd();
   gl Color 3f (0,0,0);
   renderBitmap String (315, 420, GLUT_BITMAP_HELVETICA_18
                                                 "Root");
   for (int i=0; i< fileCount; i++){
         float x = filex [i];
         float y = 250;
```

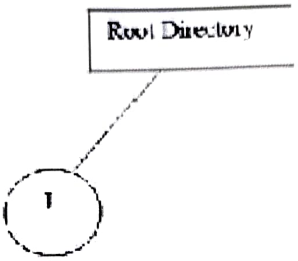
```
            glColor3f(0.0, 0.0, 1.0);
            glBegin(GL_LINES);
                glVertex2f(340, 400);
                glVertex2f(x, y+30);
          glEnd();
         glColor3f(0.2, 0.8, 0.2);
         drawCircle(x, y, 30);
         glColor3f(0,0,0);
         renderBitmapString(x-20, y-5, GLUT_BITMAP_
                                           HELVETICA_12,
                                    filenames[i]);
         }

        glFlush();
 }
void initOpenGL(){
        glClearColor(0.8, 1.0, 0.8, 1.0);
        glMatrixMode(GL_PROJECTION);
        gluOrtho2D(0, 640, 0, 480);
 }
int main(int argc char **argv){
            printf("Enter no.of files(max%d): ",Max_file
            scanf("%d", &fileCount);
            if (fileCount > Max_files)
```

```
              { printf("Too many files. Max is %d\n",
                                        MAXFILES);
              } return1
```

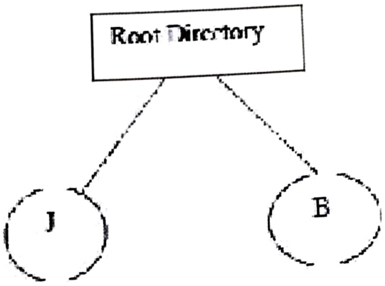
```
for (int i=0; i<fileCount i++){
        printf("Enter name of file %d: ", i+1);
        scanf("%s", filenames[i]);
 }
 int spacing = 640 /(fileCount +1);
 for (int i=0; i<fileCount ; i++){
            fileX[i] = (i+1)* spacing;
 }
glutInit (&argc, argv);
glutInitDisplayMode(GLUT_SINGLE |GLUT_RGB);
glutInitWindowSize (640, 480);
glutCreateWindow ("Single Level File Directory -
                                            OpenGL");

initOpenGL();
glutDisplayFunc(display);
glutMainLoop();
 return0;
}
```

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:

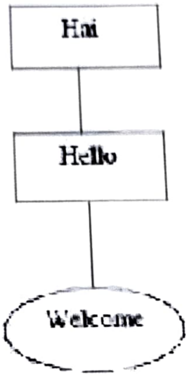
```
#include <GL/glut.h>
#include <stdio.h>
#include <string.h>
#include <math.h>

#define Max_Subdir 5
#define Max_file 10

char subdir [Max_Subdir][20];
char filename [Max_Subdirs][Max_file][20];
int subdir Count=0;
int fileCount [Max_Subdir] ={0};
 float subdir [Max_Subdirs];

 void drawCircle (float x, float y, float rad){
          int i;
          int triangle Amt =100;

          float angle;
```

```
    glBegin(GL_Triangle_fan);
    glVertex2f(x,y);
    for(i=0; i<= triAmt ; i++){
          angle =2.0f *3.1416f *i / triamt_
          glVertex2f(x+cos(angle)* rad, y+
                                  sin(angle)* rad);
    }
    glEnd();
}
void renderBitmapString (float x, float y, void *font
          const char *string){
    Const char *c;
    glRasterPos2f(x,y);
    for(I = string ; *c!='\0'; c++){
          glutBitmapCharacter(font *c);
    }
  }
 void display(){
    glClear(GL_Color_Buffer_Bit);
    glColor3f(1.0, 2.0, 1.0);
    glBegin(GL_Quads);
    glVertex2f(290,400);
    glVertex2f(390,400);
    glVertex2f(390,450);
    glVertex2f(290,450);
    glEnd();
    gl Color 3f (0,0,0);
    renderBitmap String( 315,420, GLUT_BITMAP_
                                 HELVETICA_18, "Root");

for( inti=0; i<subdirCount ; i++){
        float subdir XPos=Subdir X[i];
        float subdirYPos=250;

    glColor 3f (0.0,0.0,1.0);
    glBegin (GL_Lines);
            gl Vertex 2f (340,400);
            glVertex 2f (SubdirXPos, SubdirYPos+30
    glEnd();

   glColor 3f (0.2,0.8,0.2);
   drawCircle (SubdirXPos, subdirYPos, 30);
   glColor 3f (0,0,0);
   renderBitmapString (subdir XPos-20, SubdirYPos-5,
       GLUT_BITMAP_Helvetica-12, subdirs [i]);
  float files XPos = sub dir XPos-50;
 for (int j=0; j< file Count [i] ; j++){
    if(subdir count > Max_Subdir)
     {     printf(" Too many Subdirectories."
              Max is %d\n", Max Subdirs);
           return 1;
     }

 int spacing = 640 /(Subdir Count +1);
 for(int i=0 ; i< subdir count ; i++){
          Subdir [i] = (i+1) * spacing );
   }
   glut Init (&argc, argv);
   glut Init Display Mode (GLUT_Single |GLUT_RGB);
   glut Init Window Size (640, 480);
   glut Create Window ("2 Level Directory" OpenGL");

   init OpenGL ();

   Glut DisplayFunc (Display);

   glut Mainloop ();

   return 0;
 }
```

```
        glColor3f (0.0, 0.0,1.0);
        glBegin (GL_LINES);
              glVertex2f (Subdir XPos, Subdir YPos -30);
              glVertex2f (file XPos, Subdir YPos-80);
      glEnd ();

      glColor 3f (0.8,0.8,0.2);
      draw Circle (filex Pos, Subdir YPos - 80, 20);
      glColor 3f ( 0.0,0,0);
      render BitmapString (file XPos -15, subdir YPos -80
             GLUT_BITMAP_HELVETICA_12, filename
                                                   [i][j]);
      file pos+=60;
   }
}
   glFlush();

void   initOpenGL (){
        glClearColor (0.8,1.0,0.8,1.0);
        glMatrixMode (GL_Projection);
        gluOrtho2D (0,640,0,480);
   }

 int main (int argc, char **argv){
           printf("Enter no of subdirectories (max %d)",
                                 Max_Subdirs);
           scanf("%d", &subdir Count);
```

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:__

Hence the single & Two level Directory Structures are implemented using C.