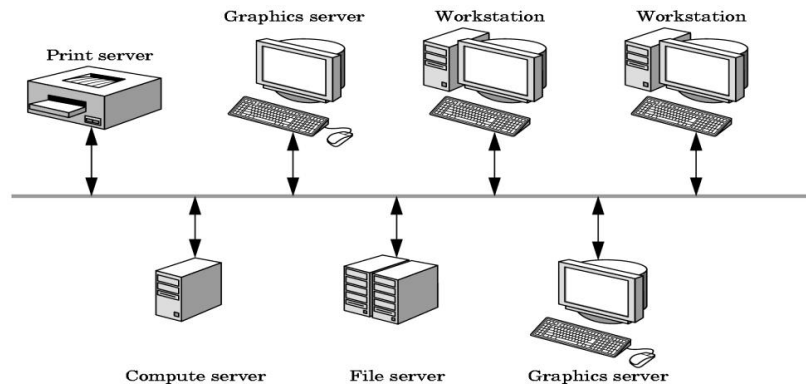


OPENGL - CLIENTS/SERVERS AND DISPLAY LISTS

3.3 Clients And Servers

- The X Window System introduced a client-server model for a network of workstations
 - **Client:** OpenGL program
 - **Graphics Server:** bitmap display with a pointing device and a keyboard



3.4 Display Lists

The Display Processor in modern graphics systems could be considered as a graphics server.

- Retained mode - The host compiles the graphics program and this compiled set is maintained in the server within the display list.
- The redisplay happens by a simple function call issued from the client to the server
- It avoids network clogging
- Avoids executing the commands time and again by the client

- Definition and Execution of display lists:

```
#define PNT 1
glNewList(PNT, GL_COMPILE);
glBegin(GL_POINTS);
    glVertex2f(1.0,1.0);
glEnd();
glEndList();
```

- `GL_COMPILE` – Tells the system to send the list to the server but not to display the contents
- `GL_COMPILE_AND_EXECUTE` – Immediate display of the contents while the list is being constructed.

- Each time the point is to be displayed on the server, the function is executed.
- `glCallList(PNT);`
`glCallLists` function executes multiple lists with a single function call

Text and Display Lists

- The most efficient way of defining text is to define the font once, using a display list for each char, and then store the font on the server using these display lists
- A function to draw ASCII characters

```
void OurFont(char c)
{
switch(c)
{
    case 'O' :
        glTranslatef(0.5,0.5,0.0); /* move to the center */
        glBegin(GL_QUAD_STRIP)
            for (i=0;i<12;i++) /* 12 vertices */
                {
                    angle = 3.14159/6.0 * i; /* 30 degrees in radians */
                    glVertex2f(0.4 * cos(angle)+0.5, 0.4 * sin(angle)+0.5)
                    glVertex2f(0.5 * cos(angle)+0.5, 0.5 * sin(angle)+0.5)
                }
            glEnd();
        break;
    }
}
```

Fonts in GLUT

- GLUT provides a few raster and stroke fonts

- Function call for stroke text :

```
glutStrokeCharacter(GLUT_STROKE_MONO_ROMAN, int character)
```

- Function call for bitmap text :

```
glutBitmapCharacter(GLUT_BITMAP_8_BY_13, int character)
```