

3D Tools for the Desktop

The Third Dimension



Linux normally follows the classic KISS (Keep It Simple, Stupid!) philosophy. The basic approach is to use a combination of small, specialized command-line tools for anything the user wants to do. Users of GUI-based desktops often find it hard to come to terms with this approach, and they typically look for a convenient graphical front-end.

A GUI tool is all the more thrilling if it uses three-dimensional graphics to display user data rather than simply displaying data in a 2D display. As you will learn in this article, even a tool with a very ordinary purpose can use advanced 3D graphics. I will discuss tools that use the graphics adapter's 3D features for a graphic display.

Installation Basics

Your distribution is unlikely to include the programs described in this article. In most cases, you will need to build and install the program from the source code. To build these programs from source, you will additionally need a collection of development packages. The required packages can vary depending on your choice of distribution. On Suse 9.2, the

Command line programs do what they are supposed to do but in a very unspectacular way. If you prefer a more sophisticated look and feel, you'll find a 3D alternative for almost every task. This article examines some of the most interesting 3D tools for the desktop. **BY HAGEN HÖPFNER**

required packages are restricted to the SDL developer package and the Mesa, *libpng*, *SDL_image*, *free_glut*, and *smpeg* libraries.

System Information

Linux is a multitasking operating system; that is, it can execute multiple programs simultaneously. To do so, Linux needs to ensure that each program only receives as much CPU time as it needs to complete. The operating system forks parent processes to create new processes. In other words, the processes form a hierarchy with *init* at its root. If you need an overview of the current processes, console tools such as *ps* or *top* give you the results. If a program freezes, and this does happen occasionally on Linux, running *kill* with the process ID of the frozen process will remove the process.

The *ps3* tool [1] gives you a three-dimensional view of the process table, displaying the processes as blocks on a rotating disk (Figure 1). If you take a closer look, you will see that the name of the process is displayed on one side of

the block. The next side of the block has a number, which is the PID. The height of the block tells you how much memory the process is using. And the color tells you which process is currently stressing the CPU.

You can use the mouse to change the display. Click once with the left mouse button to stop the disk from rotating, and click left again to restart. Hold down the mouse button and drag the mouse left or right to change the direction of rotation. The rotational speed depends on how fast you drag the mouse.

To change the relative height of the blocks, hold down the left mouse button and drag the mouse up or down. Holding down the right mouse button and dragging the mouse changes the view angle. Alternatively, you can press one of the function keys [F1] through [F6] to select a preset view mode. The [Q] key resets the display to the default value.

The *gload* program, which is available from [7], gives you a completely different approach to displaying the system load (Figure 2). You can use the mouse

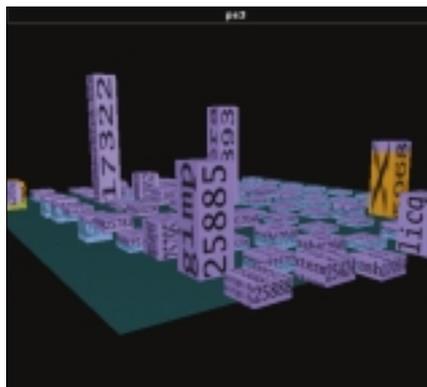


Figure 1: *ps3* provides a 3D view of the process table.

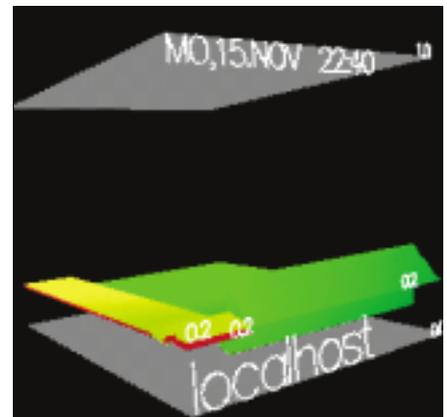


Figure 2: Checking the system load with *gload*.

to modify the 3D view, following a similar approach to *ps3*. Right click on the program window to drop down a context menu where you can toggle the "illuminated" date and time display, labels, and gray boxes on and off.

You Have Mail

Innumerable tools relate to email, which is one of the oldest and most popular Internet applications. Programs that alert users to incoming messages are among the most widespread representatives of this popular program type. The great grandpa of them all, *biff*, has been copied, modified, and fine tuned many times.

glbiff is a 3D rendering of the *biff* tools. Although the program no longer seems to be under active development, you can still pick up the source code at various locations on the Web, such as [2], for example. This said, the effort is only worthwhile if you use *fetchmail* to pick up your mail: *glbiff* does not support POP or IMAP.

Building *glbiff* with a current compiler from one of the major distributions requires a few modifications to the source code, starting in the *cfg.h* header file below the *glbiff* source code directory. You need to comment out the line with

```
#include <stl.h>
```

and add two other header files instead. The file should look like this when you are done

```
//#include <stl.h>
#include <stream.h>
#include <list.h>
```

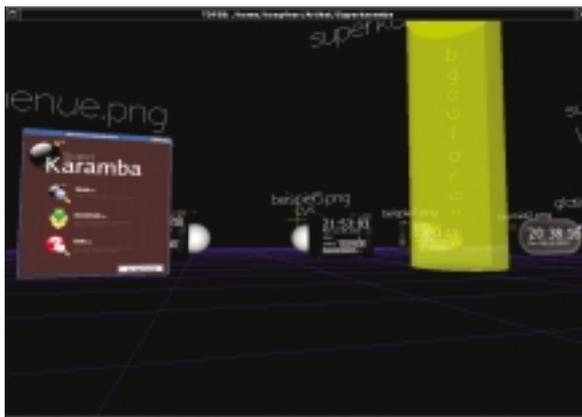


Figure 3: Hunting for files and directories with TDFSB.

Following the same pattern, you should then modify the include line for *stl.h* in the file *cfg.cc* to read as follows:

```
//#include <stl.h>
#include <list.h>
```

After completing the changes, use the normal steps to build *glbiff*, that is, *./configure && make*, and then become root and install the program in */usr/local/bin/* by typing *su -c "make install"*.

The first time you launch *glbiff* in a terminal window, the tool will quit, telling you that it *couldn't open config file /home/username/.glbiffrc*. Fortunately, the source code directory has a *glbiffrc.sample* template, which should help you provide the missing configuration file. Copy the template to your home directory as *.glbiffrc*, and change the values to reflect your mailbox directory – check frequency, mail client, and your preferred alert signal.

Browsing the File System

Although the command line has an abundance of tools for file system management, file managers such as Konqueror or Nautilus give you a more convenient approach to managing data. To add a third dimension to your hard disk inventory, check out TDFSB [3].

This program, which is shown in Figure 3, is reminiscent of 3D games. You can use the mouse and arrow keys for navigation. The mouse changes the view angle – you can turn round and look up or down. The cursor keys allow you to bend over forward or backward. TDFSB displays directories as spheres. To change to a directory, you just step into the corresponding sphere.

Files are displayed in a way that reflects their type. Text files (including HTML) are displayed as yellow columns, with the content scrolling through them. TDFSB displays

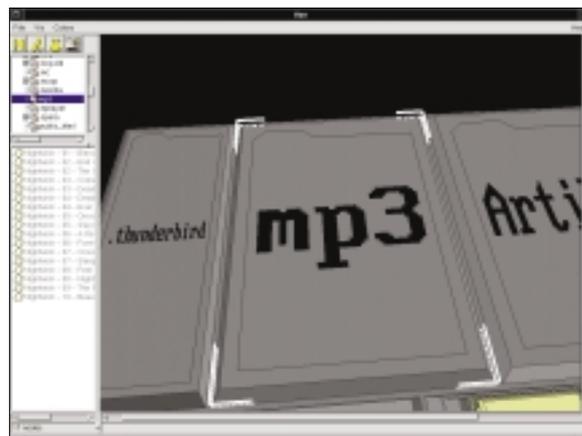


Figure 4: Pyramid-style directories in *fsv*.

images as pictures. MP3 files are displayed as CDs; you can left click an MP3 file, hold down the mouse button, and press [Enter] to play the file. The same applies to MPEG videos.

fsv [4] is not quite as reminiscent of a 3D ego shooter – although it does the same thing as TDFSB; that is, it displays directory content in 3D structures. To build *fsv*, you need the *gtk* and *gtkglarea* developer packages in addition to some of the previously mentioned components.

As you can see in Figure 4, *fsv* is not a purely 3D application. The tool also has a list window (bottom left), and a tree view (top left). However, the large 3D view is used for navigation.

Future

It looks like 3D desktop tools are here to stay. Sun has the Looking Glass project [5], and let's not forget Microsoft with "Avalon."

But the community also has a few projects of its own simmering away, such as the experimental Metisse X desktop [6].

INFO
[1] <i>ps3</i> homepage: http://user.tninet.se/~uhu537u/ps3.html
[2] <i>glbiff</i> resources: http://linux.tucows.com/preview/59877.html
[3] TDFSB Homepage: http://www.determinate.net/webdata/seg/tdfsb.html
[4] <i>fsv</i> Homepage: http://fsv.sourceforge.net
[5] Looking Glass project: http://www.sun.com/software/looking_glass/
[6] <i>Metisse</i> : http://insitu.lri.fr/~chapolis/metisse/