

Window box

Chris Bidmead goes all GUI over the glorious Gnome's multi-desktop environment.

Since the last column, my relationship with Unix has taken two unexpected turns and my head is still buzzing. Both these 'new developments' relate to the X Window System, which emerged in the mid-eighties. But the stuff I'm going to talk about here is certainly new to me.

The CDE (Common Desktop Environment) of Unix, a mostly failed attempt by commercial Unix manufacturers to give the multifarious versions of the operating system a common look-and-feel, has been a long-running feature on my AIX machine here.

I've also shown you screenshots over the years of various GUIs running on Linux, including Caldera's now-defunct LookingGlass, KDE (which replaces LookingGlass in Caldera's newest OpenLinux 2.2 distribution) and AfterStep, the approximate NeXTStep lookalike which has become my favourite.

KDE is a class above the other Linux user interfaces I've mentioned, incorporating consistent drag-and-drop features along the lines of CDE, and including a collection of applets that approaches a complete desktop environment. In a sense, KDE is an operating system of its own, inviting developers to create 'KDE-aware' applications that may not even run on a non-KDE Linux implementation.

This is one reason I've tended to resist KDE. My preferred GUI on my main workstation, an IBM PC315, remains the more modest AfterStep, which is really not much more than a pretty launcher for standard Linux applications.

Gnome is the newer rival to

KDE. Lagging some 12 months behind KDE in the development cycle, Gnome has been fairly buggy until recently. So although I've been tracking its continued development, I've been keeping it away from the machines on my network.

Gnome's nous and Enlightenment's flair create an infinitely configurable desktop



▲ GARETH WATTS OF NETCOM, NORWICH [SEE PAGE 217] IS ANOTHER VMWARE ENTHUSIAST. HE INTENDS TO USE IT ON A 256K 300MHZ MACHINE RUNNING LINUX, SO THAT HE CAN SUPPORT WINDOWS AND WINDOWS NT USERS WITH CLIENT PROBLEMS. READER EDDIE DE ROOS FROM THE DEPARTMENT OF PHILOSOPHY AT THE UNIVERSITY OF GENT IN BELGIUM <EDDIE.DEROOS@EARTHING.NET> HAS TAKEN VMWARE ONE STEP FURTHER BY RUNNING IT IN CONJUNCTION WITH VNC. THIS SCREENSHOT IS OF A WINDOWS 98 DESKTOP USING VNC AS A WINDOW INTO A REMOTE LINUX MACHINE RUNNING KDE, AND ALSO POPPING UP WINDOWS TERMINAL SERVER IN A VMWARE VIRTUAL MACHINE

Until, that is, SuSE 6.1 and Red Hat 6.0 turned up. Either of these distributions is an easy way of installing the much more stable current version of Gnome. And what Gnome has done with the Linux GUI is my first big buzz of the month.

The two new SuSE and Red Hat distributions make it very easy to install KDE or Gnome. If you have the space, do

what I did on the Siemens Celsius workstation here and install both KDE and Gnome, and you'll then get a choice at log-on

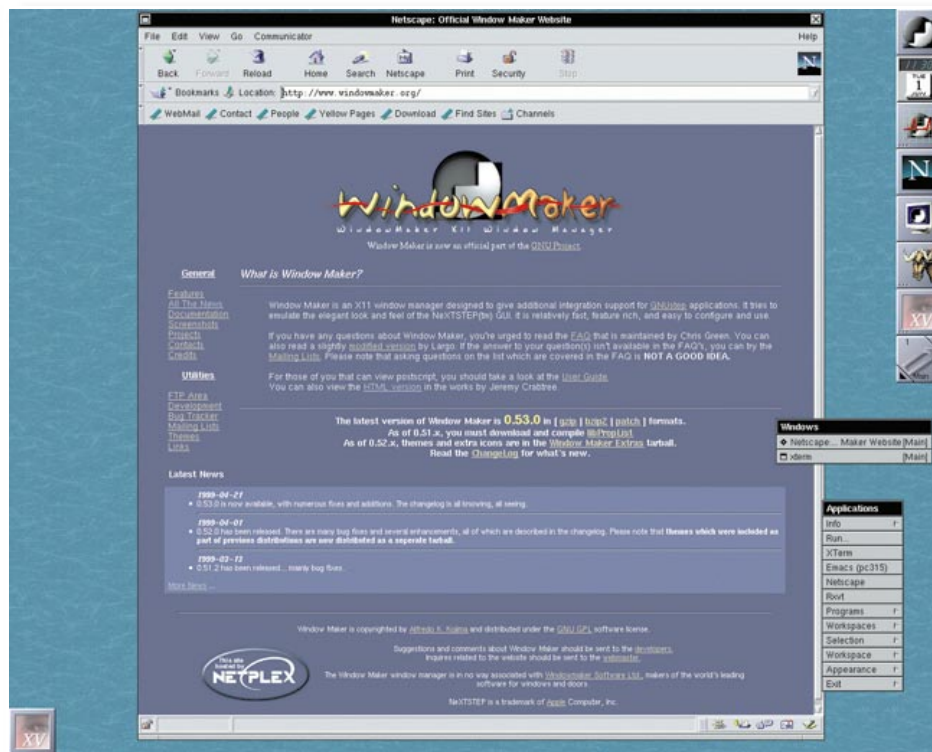
time of which environment you want to run. Or, to anticipate my second big buzz of the month, you can run both of them at the same time!

As soon as I'd installed Gnome, I understood why readers have been

mailing me so enthusiastically about it. And I see too why many of you have been urging me to write more about the Enlightenment window manager.

Enlightenment was initially developed about three years ago by Carsten Haitzler, aka Rasterman <www.rasterman.com>, later joined by Geoff Harrison, aka Mandrake. It's still very much in development — the version distributed with Red Hat 6.0 is 0.15.5 — but it's stable enough to have been adopted as the default window manager for Gnome in the Red Hat 6.0 distribution.

Enlightenment, as regular readers will know from screenshots in this column, is the Matisse of window managers. The combination of Gnome's nous and Enlightenment's artistic flair creates an almost infinitely configurable desktop (or series of desktops — I'm currently switching around between three



◀ **THE ELEGANT WINDOW MAKER [SEE MAIN TEXT] RUNNING WITH ANOTHERLEVEL ON THE SECOND X DISPLAY OF MY PC315 MACHINE, ALTHOUGH THE INSTALLATION IT SHOWS IS ACTUALLY ON THE SIEMENS CELSIUS ACROSS THE ROOM. IF YOU'VE EVER USED THE CLASSIC NeXTSTEP OPERATING SYSTEM, THE LAUNCH BAR ON THE RIGHT AND THE LOOK OF THE WINDOW FRAMES SHOULD BE FAMILIAR**

Gnome/Enlightenment virtual desktops, each one nine times the size of my actual screen...).

The other window manager Gnome offers is Window Maker, version 0.52.0. This is a much more quietly elegant affair, designed, like AfterStep, along the strict lines of NeXTStep. The Red Hat 6.0 distribution allows you to run Window Maker as an alternative window manager for another optional GUI called AnotherLevel (the default being fwm). In either case, with Gnome or with AnotherLevel, the result looks very much like the NeXTStep environment, although lacking the drag-and-drop consistency that you get with NeXTStep's complete set of compliant utilities and applications.

This must sound rather confusing, particularly if you're coming to all this direct from Windows, where the graphical environment is more or less fixed for you. Well, the fact that all this GUI stuff is based on X gives you yet another degree of freedom I've so far

only hinted at, the multi-display capability of XDMCP.

The initials stand for the XDM Control Protocol, and XDM stands for X Display Manager. xdm (in lower case) is also the name of an X utility, one of the 'proof of concept' programs that comes as part of the X bundle. If you're running a network of machines that

use X, even if it's only two machines connected together, you need to know about XDMCP. I'm amazed

X-based GUIs give you yet more freedom — the multi-display capability of XDMCP

that I'd neglected it for so long.

As a Unix user you may already be running xdm, whether you know it or not. Most Linux installations, for example, give you a choice of booting into X directly, or coming up into the command line and running startx manually when you want to bring up X.

Red Hat 6.0 uses runlevel 3 for normal command line execution, and runlevel 5 to give you automatic entry into X. The config file that determines the basics of what happens at the various runlevels, /etc/inittab, includes the following lines:

```
# Run xdm in runlevel 5
x:5:respawn:/etc/X11/xdm
prefdm -nodaemon
```

If you inspect /etc/X11/prefdm using ls with the -l switch, you'll find it's a symbolic link to xdm, or perhaps to kdm or gdm, which are the (somewhat prettier) KDE or Gnome equivalents.

The symbolic link is simply a handy low-level technique for choosing which one you want to run. Other distributions will probably do this differently: my old version of Caldera OpenLinux 1.2 dispenses with the link and simply runs vanilla xdm. SuSE 6.0, on the other hand, uses runlevels 2 and 3 for the command line and X interfaces respectively, and doesn't explicitly start xdm or its equivalent from /etc/inittab, leaving this instead to the runlevel daemon start-stop mechanism in /sbin/init.d.

Xdm as discussed so far manages the X display on your machine. It runs as root and drops you straight into a graphical sign-on that allows you define who you are and, in the case of kdm and gdm, which GUI you want to run. Red Hat 6.0 lets you choose between Gnome, KDE or AnotherLevel.

But now turn to man xdm. The entry begins 'Xdm manages a collection of X displays, which may be on the local host

or remote servers.' Not just a single X display, but a whole collection! I have some of my networked machines here set up so that when I run xdm on one machine, I get a choice of which of the other machines' displays actually comes up on my local screen. And because XFree86 by default supports two X screens, I can run a remote display and the local operating display (or another remote display) at the same time, swapping between the screens by switching virtual consoles using the standard Ctrl-Alt-Fn method. Or (optionally on a single non-networked machine) I can have two variants of the local operating system up simultaneously. Making this work turns out to be much easier than describing it. Next month I'll give you the full details.

The other Netcom

I didn't actually mean to phone Net Communications Limited of Norwich — I was trying to get through to the support line for Netcom UK. But it turns out that the Norwich-based company, founded in the summer of 1995, pipped the Americans at the post and snarfed the netcom.co.uk domain name. Netcom arrived in the UK later the same year, and so has to make do with netcom.net.uk.

I got chatting with Simon Gurney, the MD of Net Communications, about how readers of this column are still having problems connecting their Unix systems to ISPs who assume the whole world only runs Microsoft Windows. He told me that they've been using Linux for their internet-facing servers since the inception.

I've got used to discovering Linux and FreeBSD doing the server grunt work for ISPs, but my ears pricked up at what Simon told me next: the company is converting its whole internal office system over to Linux. Simon himself is the last bastion of Windows in the company, and was about to bite the bullet and go Linux around the time of my call.

The details I got from Net Communications technical manager Gareth Watts were even more intriguing. Gareth reckons he can set up a brand new machine as a Linux workstation in about six minutes. 'What, including installing the operating system?' I asked.

Gareth reckons he can set up a brand new machine as a Linux workstation in six minutes

FINDING THE RIGHT WORDS

There's a fair amount of slack in the various ways people talk about X-based GUIs, and all I'm trying to do here is lay out a simple working vocabulary. If you deeply disagree with how I use any of these terms, I hope you'll drop me a line.

➤ **GUI** The graphical user interface. Roughly describes the totality of what appears on the screen in a mouse-driven windowing environment.

➤ **X** The network based 'glue' that connects the GUI to the operating system. Traditionally X comes with a bunch of small applications, including one or more elementary window managers [*next column*], but they're mostly 'proof of concept' code. X is a GUI-enabler, not in itself

a GUI (people get very confused about this) and as such is essentially invisible.

➤ **Window Manager**

The code that enables X to create windows on your screen, defining how they look and behave. Think of the window manager as controlling the frame of the windows, and the way the windows interact with one another. It shouldn't have any effect on the behaviour of the application within any window (although it might do if, for example, the window manager grabs key combinations before they can reach the application).

➤ **The Desktop** An organising application or set of co-ordinated applications that creates a consistent graphical

working environment, often with a launch bar and/or a method of arranging active icons on the screen backdrop. The environment may include drag-and-drop between windows and possibly the launch bar, although this will only work with compliant applications specially written to the developer guidelines for that desktop.

You can see how X's extra 'proof of concept' add-ons blur the edges between an invisible glue and a working GUI. In the same way, the window manager isn't strictly confined to policing the edges of the application windows. Most window managers support a system of 'hints' that can be used by applications to integrate with the window manager.

'We don't install the operating system,' he said. 'We use Netboot or Etherboot.'

These are two similar techniques for turning a machine into a Linux workstation by pulling the kernel image from a server across the network. A floppy disk, or a specially burned EPROM in the network card, initiates a small loader utility that sends out the ethernet address of the network

card over the network with a request for a response. A BOOTP or DHCP server hears the request and sends the nascent workstation an IP address and hostname. That done, the boot loader code on the workstation uses a simplified version of FTP (called TFTP, or Trivial File Transfer Protocol) to start pulling a copy of the Linux kernel across the network and into

memory. The BOOTP or DHCP server deals out a special version of the kernel called a 'tagged image' — in this case a regular Linux kernel that's been run through a utility called mkmbi-linux (make netboot image, linux version) to add special header information as defined at www.slug.org.au/etherboot/doc/html/spec.html.

The compiled kernel will include a driver for the particular network card the workstation is using, and also the code to run as an NFS (Network File System) client. This will enable the kernel to mount a remote filesystem on the server as its root. So, the workstation doesn't even need its own local hard drive.

PCW CONTACTS

Chris Bidmead welcomes your comments on the Unix column. Contact him via the PCW editorial office or email unix@pcw.co.uk