



Running hot and cold

After a disappointing initial experience with ISDN, Chris Bidmead has now found an affordable router to speed up connections. Plus, the wise giants of Linux have their say.

Round about this time last year I was boasting to you about having had an ISDN line installed, and the increase in speed it brought to my internet access... Well, mostly just about having the line installed, because if the truth be told, the speed increase over ordinary phone dialup wasn't that great.

There was *some* reduction in the time it took to connect, but I wasn't seeing anything like the theoretical two seconds connection time. My ISDN terminal adapter (TA) was connected via a serial line into one

of my workstations, on which I'd have to run a dialup app which would trigger my ppp daemon, which would then tickle the TA into action, and I'd get a connection in about 20 seconds.

During connection, all the packets would have to go through the somewhat sluggish ppp daemon on my NeXT machine, and along the serial wire, which can be a bottleneck at these kind of speeds. Yes, it was still faster than a 33Kbps modem, but frankly not much and nowhere near the theoretical rate of 64Kbps for my ISDN line.

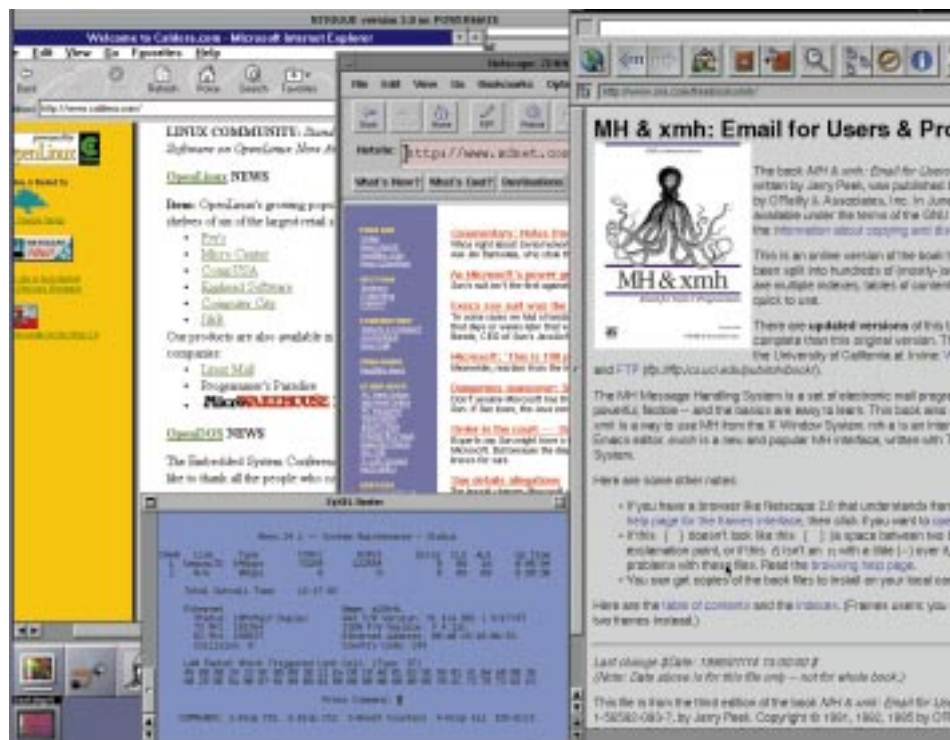
The reason I'm putting all this in the past tense is that I've just installed a new gizmo which changes everything. It makes me feel like someone who's been boasting about having a mains water supply installed and has been taking friends down to the end of the garden to see the tap sticking out of the ground. It's called an ISDN router. It's a class of product that has been around for several years, but what's new is the price, which brings it within reach of the ordinary user. This one comes from the Taiwanese company ZyXEL, and is distributed in the UK by P&L Systems.

I'll tell you more about the router next month. In the meantime I'll just round off the rather crude "mains water" analogy by saying that I now have hot and cold running water throughout the house. (See the screenshot, left, for further clues.)

Up and running

I can't say I always enjoy helping readers through their installation problems (read all the relevant HOWTOs at www.caldera.com/LDP/HOWTO/HOWTO.index), but there's a lot of satisfaction when, after a series of letters, I receive correspondence saying: "I'm up and running." To add to the fun, this can be a multi-stage process: "Great, I'm up and running... but now I need to install X." And then: "Thanks, Chris, I have a complete Unix system... Now help me set up my ppp connection to my provider."

Microsoft and Mac users who read this column to find out what's happening on the other side of the fence are welcome, and I'm keen



I couldn't resist showing off. Here are three different browsers running on three different machines (Netscape and Microsoft Explorer are X Windows, on the Apricot LS550 and the NEC PowerMate respectively, and my NeXT machine which is running its own OmniWeb browser). The browsers are all live, each with their own set of TCP/IP connections to the ZyXEL router which in turn connects them to the 64Kbps ISDN line to my service provider

Chris's question time: Finding your way

Q Chris, I'm lost. I have a working Linux system, but I've no idea what programs are available on it, or where to find important files.

A That's OK. We all start out feeling like that when coming to Unix from, say, the DOS world. It reminds me of the tribe of pygmies who had a habit of getting lost in the long grass. Here in the Unix world, the long grass is certainly very tall, but we don't just jump up and down in it; we sit on the shoulders of giants. And where do you find these giants? Where else, but right inside your Unix system. If you've been following my column, you'll have heard of the daemons in there; those

tireless slaves who take care of the grunt-work. But the wise giants are all in there, too, ready to give you a lift any time you want to wake them up.

If you find this personification of Unix rather twee, try to get used to it. As with the ancients, personification turns out to be a useful way of making sense of a complex world. Let's start with the command "man man": it evokes the manual page about the manual system. Read that, and you'll know how to find out about the whole Unix system and all the installed utilities — and the programming stuff, too. It will tell you how to use "man -k" or "apropos" (which means the same thing) to find

manual entries on everything relevant to a particular subject: booting, managing the keyboard, or whatever.

If finding files is your problem, "man find" will tell you all about the find command. This will find files, although it's most important use is for delivering a qualified list of files names for you to do something else with, like feed into another command. But "man find" will also point you in the direction of the locate command, which uses a database for fast look-up of file locations. "Man locate" will tell you more about this, particularly how you maintain the file database, and will direct you to the Info chapter on Finding Files.

Bidmead's books: *UNIX Power Tools* (2nd edition)

Authors: Tim O'Reilly, Mike Loukides & Jerry Peek
Price: \$59.95
ISBN: 1-5659-2260-3
Publisher: O'Reilly

This second edition of the classic tome, first published in 1993, is a substantial revision of the original work, incorporating hundreds of readers' suggestions and corrections, and is now slanted more heavily towards the ubiquitous GNU flavours of Unix, including Linux. This toolkit of tips and tricks is a shapeless collection of short articles by a variety of authors. You'll search in vain for a narrative line or a personal philosophy.

It's big, too. Including the bundled CD-ROM of freeware source and binaries for the most common Unices, it weighs three and a half pounds and runs to over 1,000 pages! It's an unglamorous book which deals in the lowdown on the sometimes unlovely entrails of Unix, with tips on taming vi, backing up files with tar, and the messy nitty-gritty of file ownership flags. It tells you how and when to kill processes (and when not to). It contains valuable hints on improving your shell programming (avoiding the C Shell like the plague). It's also a sampler for other O'Reilly books, with extensive borrowings from O'Reilly's *Nutshell* range of handbooks.

If you don't care much about the real Unix that lurks under the veneer of X, this book is probably not for you. Neither will this be the right book if you're looking for a general overview of Unix. However, if you work regularly with the operating system and are ready for a fundamental seminar on, say, how files are connected to filenames, then this dumpy tome is the bible to turn to. As co-author Mike Loukides jokes: "The ability to tumble about inodes is the key to social success at a Unix gurus' cocktail party." This book is just such a cocktail party.



message is probably the easiest way to discover the device name of your CD-ROM drive, which varies from system to system and from drive to drive.

If you're using Linux, Shift-PageUp will scroll the boot screen back to find it, as long as you do this as soon as the boot has completed. (The Caldera OpenLinux implementation offers a pause point in the install that suggests you do this.) But if you're like me, you won't notice that the CD-ROM drive is missing until you try to access it, and by then you've probably lost those initial screens out of the console buffer.

So how do you read these messages again? Simple — that's what the "dmesg" command is for. Read the dmesg manual pages. It was clear from the dmesg output that Jonathan's system wasn't finding the CD-ROM device, so I directed him to Jeff Tranter's CD-ROM HOWTO and suggested he take a look at the Modules HOWTO as well. Armed with those two documents he was able to install the CD-ROM with no further help from me.

Yes, it's all working fine, he wrote in his next email... except the most important thing: how do I connect to the internet?

I sent Jonathan back to the good old HOW-TOs, particularly the PPP-Client-HOWTO and Winfried Trumper's excellent PPP-over-minicom mini-HOWTO.

that as many readers as would like to should bring up their own live Unix systems. So if you're struggling with an intractable install, drop me an email and we'll fix it. For instance, there are some elementary installation queries which repeatedly crop up. The geometry of large hard drives is one of them (take a look at the BootPrompt HOWTO and the Large Disk mini-HOWTO).

Another old favourite is exemplified by reader Jonathan Murphy's query: "When accessing my drives, I can't get my CD-ROM to mount. Under Slackware it was

'mount /dev/hdd (dir to mount to)' but when I try this under Linux Pro it reads 'mount: /dev/hdd is not a block device /dev/hdd: No such device'.

I told Jonathan about "dmesg", a great installation diagnostic tool. It took me ages to discover it, but when you know about it life is a lot simpler. "Remember those initialisation messages that scroll up your screen too fast to read when you're booting up?" I wrote back. "Among them will be a line announcing that a probe has found (or perhaps not) your CD-ROM drive." That

PCW Contacts

Email Chris Bidmead at unix@pcw.co.uk

ZyXEL ISDN router from P&L Systems
info@pls.co.uk



Fuss in boots

Chris Bidmead presents another solution to problems with booting up. Using Boot Managers lilo and loadlin, he explores everything from master boot records to keeping kernels.

Richard Moore describes himself as a “daily user of a number of UNIX operating systems (mainly Solaris and Linux)”. Richard <moorer@cs.man.ac.uk> has reminded me about a great Linux utility that should be particularly interesting to readers who dual boot between Linux and Windows — I know from your mail that a lot of you do. It’s no dark secret, but an obvious solution that I’ve overlooked in the past when talking boot problems.

When you install Windows 95, it likes to snoop into your hard drive’s master boot record (MBR). If it’s a dual-boot installation, and another operating system is using the MBR to boot, Windows 95 regards this an error. It graciously and silently “repairs” your MBR, and you lose the ability to boot

into your other operating system. The simplest solution is to use the loadlin DOS utility that comes with most Linux distributions. The author and maintainer of loadlin, Hans Lermen, also calls it “the safest way to boot Linux from your hard disk if you have a bootable DOS-partition besides Linux on your machine”.

Unlike lilo, the Linux boot manager written by Werner Almesberger that comes with most Linux distributions, loadlin is used to boot a Linux kernel from the DOS prompt (optionally with the standard boot arguments — see the Boot Prompt HOWTO), but can also be set up to launch automatically, using Windows 95’s multiple config arrangement. One essential role for loadlin is where you have hardware that

depends on the supplied DOS driver to initialise it into a particular known state (SoundBlaster-compatible cards, for example). Loadlin can also save space on your local hard drive by pulling in the kernel image from a server on the network.

Richard’s tip is that when upgrading to Win95 (or reinstalling it), make sure you keep loadlin and a copy of your kernel on the DOS filesystem. If (or do I mean when?) you trash your MBR, all you have to do is boot into DOS and restart your Linux system by running loadlin. From inside Linux you can tell “use lilo” to rewrite the MBR.

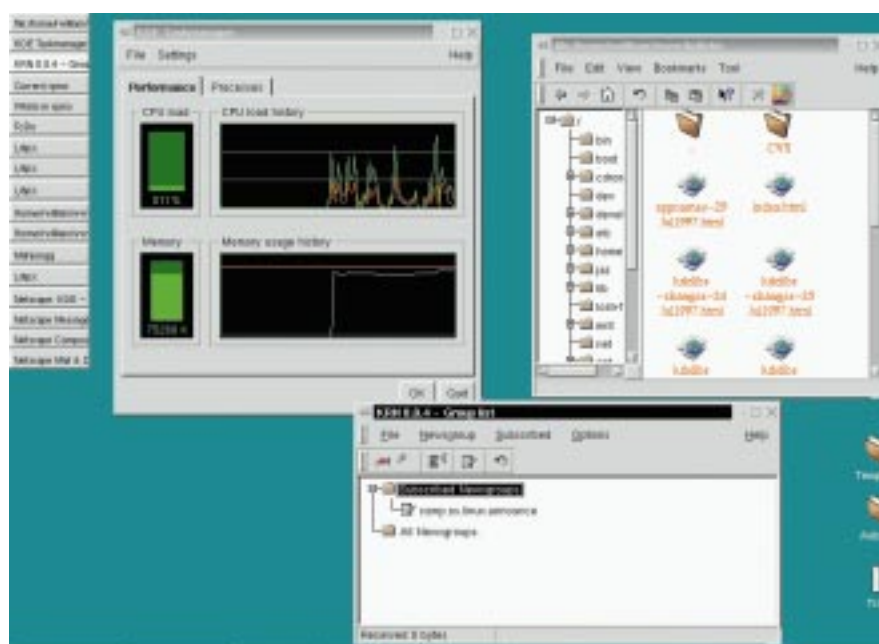
Says Richard: “Loadlin is reliable and I’ve used it in the past as the normal way to start my system in Linux via a DOS start-up menu. (Incidentally, Red Hat’s DOS autoboot.bat script for installing Linux direct off its CD from DOS is just a wrapper for loadlin.)”

If you want the full grizzly details about booting and master boot records, a good place to start is the README that comes with lilo. The PartitionMagic (a commercial boot manager) manual is excellent, too. If you *don’t* want the full grizzly details, please follow Richard’s tip or stick to my own more rigorous rule, which is: *Don’t mess with the MBR*. In other words, put lilo anywhere but the master boot record, unless you really know what you’re doing.

There are more loadlin, lilo and Linux boot utilities than you can shake a stick at: a good place to start is sunsite.unc.edu/pub/Linux/system/boot/.

A new free desktop

It was particularly good to hear from Richard, because (along with Kalle Dalheimer, one of the programmers of Star Office) he’s part of the team developing an exciting new desktop for Linux and other



The KDE desktop depends on a specially written set of widget and support libraries that are shared by all the applets. This helps keep memory usage low, and a Linux system that uses it can be run in 8Mb of RAM (see “A new free desktop”, alongside)

Chris's question time

Q Thanks for your notes on how to connect to BT Internet from a UNIX system, but I'm still not getting any support from the helpdesk.

A I've had a surprising amount of email from readers about the problems of connecting to BT Internet. Mostly I've managed a personal reply, but a few of your responses got lost in my mail system. My loss, because some of these have turned out to be very helpful. One reader, Toby Riley <toby@handc.btinternet.com>, a network support analyst, tells me that as a result of similar experiences of his in trying to hook up Linux to BT Internet, a news group <news.btinternet.linux> has been set up. He tells me that on BT's ftp site there's also a FAQ on how to connect to BT using UNIX.

Q I've managed to install Linux, but can't seem to get X working with my video card, which uses an "obscure manufacturer" chipset.

A Congratulations on having installed Linux. You're now running an

industrial-strength OS. There's a lot you can do with and learn from raw Linux, and I spent the first six months of my Linux life in console mode because I, too, couldn't get X running.

What have you done to try to get it working? If the install fails for whatever reason and you've had no joy with the XF86Setup utility that comes with XFree86, see what the command line utility SuperProbe says it thinks your chip is. (Alternatively, run X by itself with the "-probe" argument.) If you have a whizzy accelerated chip that XF86 doesn't know about, it may be that SuperProbe can detect it as a more generic subset and you could enter that into the setup.

If this fails, take a look at the excellent XFree86 HOWTO. It comes with a full list of supported chipsets (which grows; so check the latest version) and more information than you'd ever get from me about how to set up your XF86Config file.

You might have to switch out the video sub-section that came with your machine and add a plug-in card that uses a chipset

in the supported list. When you buy it, specify that you want it for Linux (in the UK, legally this makes it easier to return). It may cost a few quid, but it will be worth it because the XFree86 HOWTO points out that: "In general, XFree86 on a Linux system with an accelerated SVGA card will give you much greater performance than that found on commercial UNIX workstations (which usually employ simple frame buffers for graphics)".

■ Dear readers

Could you help me out by being a little more explicit in your subject headers? I don't want to do anything to discourage your email, but please try to make the subject line as specific as you can.

Richard Moore <moorer@cs.man.ac.uk>, clearly an old hand at this sort of thing, heads his mailing "Subject: Using 'for' to remove filenames in a file". That does two things: it tells me what to expect, and distinguishes itself from dozens of others so I can refer to it easily.

Thanks — CB.

Unices. It's called KDE (Richard doesn't tell me what this stands for — could it be the Kalle Dalheimer Experience?) and it's a set of integrated applets including a window manager, file manager, terminal, editor and mailer that makes up a drag-and-drop desktop along the lines of CDE.

Richard tells me that currently there are 30-40 applications and applets that are part of KDE, and new ones are being developed all the time. "Using a number of small applets rather than one huge application," he says, "is very much the UNIX philosophy and is one we've followed in KDE."

KDE is available under the GNU Public Licence and there are source binary RPMs (the RedHat distribution format, also used by the Caldera and Debian distributions) for a number of platforms. Says Richard: "Linux is currently the best supported platform but KDE is being used on Solaris and several other versions of UNIX (there's even a guy who's got some of it to work on OS/2)".

Although KDE is still a "work in progress", Richard says it's stable enough for use on a home machine. "But it's very much under development so don't expect perfection." Users of Win95 should feel at home, because the look and feel is similar,

although the project is by no means an attempt to develop a clone interface. "We've taken ideas from all over," Richard tells me, "and if we didn't like the way other systems did things, we've done it our own way." Anyone foolish enough to attempt to copy Windows would find themselves playing a continuous game of catch-up.

The KDE team has designed the desktop to be "network transparent", which means that anywhere a filename is expected, you can specify a URL instead. You can browse the web using the KDE file manager, in the way that Microsoft is promising for Windows 98, and then use drag-and-drop to drag a document into the editor. You can get a copy at www.kde.org.

Batch deletion again

Another hardened UNIX expert, Mark Wooding <mdw@excessus.ebi.ac.uk>, seems to have been pleasantly amused by my ambling adventures in trying to solve the batch deletion problem (PCW, October).

If you remember, I was working out the best way, under the bash shell, of feeding a list of files (in a text file called killem) into rm for deletion. What I ended up with was:

```
rm -f `cat killem`
```

Mark points out there's a special UNIX command for feeding a list into another command. The command substitution trick works fine in the example I've given, but it can run into problems where the expansion exceeds the limits of the command line length. Command substitution automatically expands shell metacharacters, which is probably not what you intend.

Mark proposes the xargs command. I've used xargs before, but only by copying it parrot-fashion, and I've never understood it. Xargs can do complicated things. Sadly, the examples that demonstrate its use in the textbooks tend to be complex as well. The way Mark uses it below is a clear demonstration of the principles.

At its simplest, xargs says, "treat the first argument that follows as a command to be run, and use the remaining arguments to construct an argument list to be supplied to that command". Here's how Mark uses it:

```
xargs rm -f <killem
```

Thanks, Mark.

PCW Contact

Email Chris Bidmead at unix@pcw.co.uk.



The Scenic route

Chris Bidmead is going away, and along with his shorts and sunscreen, he packs that most essential holiday essential; a laptop with Linux. Here's how he made sure it would travel well.

I'm off on holiday, so I'm in the process of refurbishing a laptop (a smart little Scenic Mobile) with an updated version of Linux to take with me: the pressure is on and I don't need amusing installation quirks at this stage. So, I turned to Caldera's latest release, OpenLinux Standard 1.1 <www.caldera.com>. The previous version went on like a dream once I'd got help via the internet with the XF86Configfile that defines the X Window settings.

It took only a couple of hours to do the install and get the new operating system bedded-in, including installing several

hundred megabytes of additional software, setting up the networking via an Acctron 2212-2 PCMCIA card and getting the thing to dual boot with Windows 95, using Partition Magic.

OpenLinux Standard isn't based on the latest version of Linux: out of the box it installs the 2.0.29 kernel, but frankly, on this holiday I need a machine that will do real work, not a bleeding-edge operating system with entertaining nuances. Even so, taking a close look at this relatively old "stable version", I realise how nicely the Linux development is coming along. The air isn't thick with marketing promises about "Linux

98"; unpaid Linux and Gnu developers around the world are just getting on with improving their software.

For example, the technique of configuring

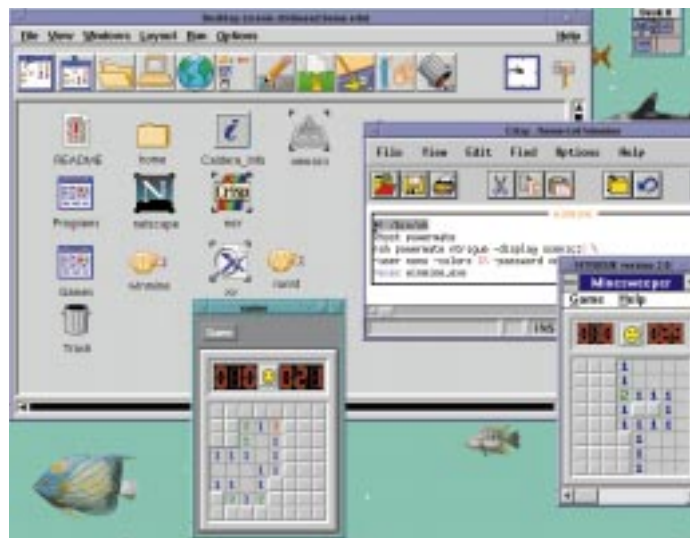
kernel function by dynamically loading modules (rather than having to recompile) is now well advanced. The install process automatically detected my Acctron network card and loaded the PCMCIA handler, pcmcia_core, and the network modules needed to drive it. In the early days of Linux you'd either have started with an overblown kernel pre-compiled with unwanted stuff, or you'd have had to recompile to tailor the kernel to your system requirements.

Exploring modules

Okay, I know Windows 95 does all this, too, or tries to, but the thing I like about Linux is that none of this need be mysterious. If you want to explore Linux' module handling for yourself, a good way to start (well, the module HOW-TO, but that's too obvious) is to run the lsmod command. This not only shows you the modules running on your

Fig 1 Running the lsmod command

Module	#pages	Used by
vfat	3	1
nkfs	3	2
pcnet_cs	2	1
ds	2	[pcnet_cs] 3
i82365	3	2
pcmcia_core	7	[pcnet_cs ds i82365] 4
ipx	3	5
psaux	1	1
misc	1	[psaux] 1
ppp	5	0
slip	2	0
slhc	2	[ppp slip] 0
lp	2	0
isofs	5	0
nfs	12	6
sg	1	0
st	6	0
sr_mod	4	0
sd_mod	3	0
scsi_mod	10	[sg st sr_mod sd_mod] 4
8390	2	[pcnet_cs] 0



I'm writing this in the garden, networked to my office machines so I can run mission-critical applications like MineSweeper from my NTrigueserver (the Winnine script I knocked up to evoke it is in the window behind)

Questions & Answers

Q What is UNIX, and why should I care?

A UNIX, like any other piece of software, is just a bunch of ones and zeros. It is an operating system (nearly 30 years old) that grew out of a radical, rebellious spirit. But then it got fat, put on a suit and made a lot of money. This excites some people but probably not your average PCW reader. What is exciting is the emergence of son-of-UNIX in various guises including Linux, BSD, and Apple's imminent Rhapsody.

Q What is Apple doing here? This isn't Hands On Macintosh.

A No, but I'm talking about the underlying BSD-like operating system of NeXTStep which, with some tweaks, is set to become Apple's industrial-strength alternative to MacOS. After two massively wallet-vacuuming attempts to come up with a next-generation operating system, it has admitted defeat and fallen back on something that, like Linux, is essentially UNIX in all but name. This move of Apple's

has profound implications.

Q Steady on — it's only ones and zeros, remember...

A Yes, quite right. And that's the point. The failure here is not just Apple's. Over the past decade, Apple (IBM, Microsoft and others) got caught up in a whirlwind of escalating aspirations about what operating systems should do. This turns out to be a Bad Idea. Moore's Law says that hardware keeps getting smaller and faster but the same rules don't apply to software: software bloats. The move by Apple to UNIX signifies a return to the "keep it simple" principles on which much of UNIX is founded.

Q Er... but UNIX is very complicated.

A Windows 95 and NT are at least as complicated. The difference is that Microsoft has a mission to make software appear as simple as possible; an honourable enough goal, but hiding complexity means you sacrifice choice

and configurability. An operating environment can be bulging with function, but unless the ordinary user can get at that function and put functions together to do unique new things the designers didn't think of, you have to stick to the tramlines.

Q But Microsoft is continually producing software with more function.

A Laying more and more tramlines: "Where do you want to go today?... OK. That'll be in the next version." And because its earnings come mostly from upgrades, Microsoft has an imperative to keep us trading-up to increasingly function-rich new versions.

Q So this column is anti-Microsoft?

A No, it's pro-choice. This column isn't about being UNIX hermits: most of us have at least a bit of Microsoft in our lives and many of us run systems that dual-boot into UNIX or Windows — like the laptop I'm taking on holiday.

system, but also lists the dependencies.

Fig 1 shows what mine looks like.

You can see from this that pcmcia_core is used by three modules called pcnet_cs, ds and i82365. But how are they loaded during startup? The startup files are kept under /etc in a directory called rc.d. (There are two rather different philosophies of boot design, System V and BSD. Linux uses the System V approach.) Rc.d contains a script called rc.modules. You'd need to know a lot more than I about shell scripting to understand intimately what rc.modules is doing, but it's not hard to get the general idea: it loads a bunch of modules that are listed in a file. A complete beginner might have some difficulty identifying the location and name of this file, because rc.modules sets it up with the line:

```
MODULE_LIST="/etc/modules/`uname r`  
/`uname -v`$Sep$1"
```

This is an example of "command substitution" (which I touched upon last month). Right in the middle of a file path we have a command wrapped in back-quotes, so the string returned by that command becomes part of the path. But as you can see, this happens twice: once in the path and once in the file name itself. The expression finishes by throwing in a couple

of shell variables: \$Sep and \$1 (these boil down to ".default" if rc.modules is evoked with the parameter "default").

Try running **uname** from the command line, first without a parameter, then with each of these two parameters in turn. Unadorned, the command returns the string "Linux", and the parameters respectively give you something like:

"2.0.29"

and

"#1 Tue Feb 11 20:36:48 MET 1997"

I say "something like", because these strings will vary depending on the version of the operating system you're using, and its release date. As you can find out from **man uname**, the command is a GNU utility that returns information about the machine and operating system on which it is running.

The point of using these strings to name key files is to make sure they don't get evoked by a version of the operating system that can't handle them. Left to someone like me, the module list would probably be called something imaginative like "module.lst". Then when I upgrade the kernel to a new version that doesn't support the earlier modules, things would start breaking and the boot might even hang. Using command substitution like this makes

sure that rc.modules is only pulling in a module list designed for the current version.

Linux also puts all its filesystem handling into modules, and there are plenty of them, which is why it's such a good operating system in a mixed environment. As well as most of the standard UNIX flavours of filesystem, Linux can handle partitions that belong to OS/2, DOS and even the Amiga. In a dual-boot machine like the Scenic I normally establish a permanent mount so that Linux can read and write to the whole of the DOS partition, and I did this by setting up **/etc/fstab**, the file that sets out what is mounted where. But the FAT filesystem I was using truncates the name of files written by Windows 95 to 8.3 format. No matter. Load the VFAT file system with the command **insmod vfat.o** (or better still, add it to the list we discovered above, so it automatically loads at boot time). Then create a new directory, **/mnt/vfat** (say) and you can mount the Windows partition in a way that shows the full filenames:

```
mount -t vfat /dev/hda1 /mnt/vfat
```

PCW Contact

Email Chris Bidmead at unix@pcw.co.uk.



Weird and wonderful

Chris Bidmead's aim is to inform you of the many treats in Unix, so this month he returns to a Procmal poser from Peter who has discovered Popover. Plus, Bash comes to the rescue.

Peter Balfie from University College London thinks I was "very silly" to mention Procmal (*PCW*, August 1997) because *"It makes punters like me, who've never been anywhere near it, due to the huge amount of 'unixspeak', try to use it!"*

Excellent! It reminds me of the poet Christopher Logue's alleged response to that visa waiver form they hand you on aeroplanes: against the section that asks whether you're coming to the US to peddle drugs, overthrow the government or engage in acts of moral turpitude, he reportedly wrote: "Sole purpose of visit".

People are put off Unix because it looks weird, it's old, and it doesn't come wrapped in yards of marketing. The sole purpose of this column is to alert you to the fact that it's bristling with possibilities within surprisingly easy reach. What you can do with a few lines of Procmal is a case in point. Except that Peter couldn't do anything. He put the .forward file into his home directory, as the documentation suggested, but it seemed that Procmal wasn't being called.

The first thing is to take out the .forward file and send an email to yourself. If you're using sendmail on a dial-up connection you can do this while you're offline. Check this mail item has arrived and write a .forward line that simply names an arbitrary file:

```
/Users/bidmead/Test.forward
```

The file can be anywhere and doesn't need to even exist, but make sure it's in a writeable directory. Quote the pathname in full because sendmail's forwarding mechanism won't expand shell conventions like "~" as an alias for your home directory.

Now send a mail item to yourself again. It shouldn't arrive in your usual mailbox. If the forwarding mechanism is working it will have created a new file to receive the

message, or will append your message to an existing file of that name.

On Peter's system this didn't work, and we found that this was because he wasn't using sendmail to do SMTP (Simple Mail Transfer Protocol) but had a POP (Post Office Protocol) account. The latter is appropriate for a standalone machine that only intermittently connects to the internet, because it leaves the mail transport entirely to the remote server. POP just collects a "mail drop" at each logon.

Peter's setup was different. *"I'm on a campus POP3 mailserver for email,"* he told me, *"and I use 'Popover' run from a crontab file to pull down new messages, which end up in usr/spool/mail."* Peter's mail-user agent picks them up from here and displays them in the usual way. What's needed here is for Popover to pass the mail directly to Procmal. Peter found the instructions for doing this in the help files for Popover.

Popover lets you name the directory in which you want mail to be deposited, but instead of a location you can use a pipe to another program, like this:

```
|/usr/local/bin/formail -e -s
Procmal
```

This puts Procmal in the loop, and now Peter's managing his mail across multiple mailboxes and finds he can also use Procmal and Formail to handle the mounds of historic mail in his mailboxes. "This is a potentially huge hidden benefit," says Peter. "My 'in' mailboxes together add up to 2,500 messages, stretching back over 17 years."

Peter is working with NeXTStep, the officially defunct Unix-based operating system set to be resurrected by Apple as Rhapsody. So his Procmal pipes its output through the NeXTStep utility, appnmail, rather than delivering it directly into a file or

directory. Appnmail transforms the mail into a form NeXTStep's MUA can handle, and also takes care of the file locking.

Last month, I teased you with the question: "Why don't my own Procmal recipes include the trailing colon that requests Procmal to manage automatic lock files?" Well, it's because in the NeXTStep environment, appnmail does all the lock handling for me.

While I was experimenting with the .forward file I discovered that you can put multiple lines in there, and sendmail's forwarding mechanism will respect all of them. A handy use for this is to create a raw backup file of all your mail before Procmal gets its hands on it. In the early days of your experiments with Procmal, this should guarantee an ability to roll back on your incoming mail in case anything goes wrong. To the single .forward line I suggested in the August column, add a second line mentioning a filename ([Fig 1, p260](#)).

The mail will go to both locations, and if you ever need to recover it from the file, you can use formail to pipe it back into your mail-handling system ([Fig 2](#)).

Batch deletion

Several of you have written in asking how to delete a set of files whose names are stored in an ASCII file. "Man rm" doesn't give much in the way of clues about this. You might think this is remiss of the designers of the rm command, but you'd be missing the point: in Unix, you combine the talents of the various utilities to achieve what you want: we want to write a command line that feeds the names into rm, individually.

One suggestion, from someone using the Bash shell, was to use the built-in "read" command, so, just for fun, let's explore that



Quantrix is one of a dozen NeXTStep apps for which Lighthouse is giving away two-user licences (see [this page](#))

filenames for the filenames themselves. So, instead of Fig 6 we'll write Fig 7.

The backquotes (don't confuse them with single quotes ' ') inform Bash that it should carry out the command substitution.

BT Internet revisited

Last month I relayed the saga of trying to connect a Unix workstation to BT Internet and my frustration at the refusal of BT Internet's support people to face up to the cross-platform implications of the net.

Since then, I've had a long chat with Raj Kanthan, BT Internet's product manager, who confirmed that the intention is definitely to offer a platform-independent service.

"I can see what's happening here," he says. "When you say Unix, they say: 'Oh, we only support Windows'. That's true for the client software, but it's a service that will work with any machine."

What the help desk people don't tell you is that you don't need BT Internet's Windows software to register. The company seems to have revamped its registration procedure since my early struggles, and if you can borrow somebody's internet connection from a machine running any kind of web browser, you can sign up by visiting www.btinternet.com. Raj says: "You can get on with any client and operating system that supports IP and HTML and CHAP."

Once you've registered, and at any time after that, the full platform-independent details you'll need to set up your chap-secrets file will be available as a personal web page that you can access with your password. They're also sent to you in a welcome email.

Lighthouse

Devoted NeXTStep users have just received a considerable bonanza. Lighthouse, the long-standing NeXT application provider now absorbed by Sun, is giving away free two-user licences on a dozen of its commercial-class NeXTStep applications, all of which are downloadable from www.lighthouse.com/CurrentProductVersionNumbers.html. Included is the magnificent Quantrix (pictured, above), the best multi-dimensional spreadsheet I've come across on any platform.

```
/Users/bidmead/Test2.forward "IFS=' '&&exec
/usr/local/bin/procmail -f-||exit 75 #bidmead"
```

```
formail -s /usr/lib/sendmail -t < /Users/bidmead/Test.forward **
```

```
read: read [-r] [-p prompt] [-a array] [-e] [name ...]
```

```
bash> while read < killem ; do echo $REPLY ; done
```

```
bash> while read ; do echo $REPLY ; done < killem
```

```
bash> rm -f file1 file3 file3 file4...
```

```
bash> rm -f `cat killem`
```

possibility. Bash is available across most platforms in a GPL'd distribution and I've just installed it on my NeXTStep system in the interests of consistency.

One nice feature for beginners is the way you can pick up tips on individual aspects of the shell behaviour by typing "help" followed by the command in question: a sort of online mini-manual. For example, Fig 3 shows what "help read" reveals.

One line is read from the standard input, and the first word is assigned to the first NAME, the second word to the second NAME, and so on, with leftover words assigned to the last NAME. Only the characters found in \$IFS are recognised as word delimiters. The return code is zero unless end-of-file is encountered. If no NAMES are supplied, the line read is stored in the REPLY variable.

To read line by line you might be tempted to turn to the Bash shell's "for" construction, but it doesn't work the way BASIC or C programmers might expect. "While...do...done" is more promising, and

you can test this non-destructively with a command line using "echo" instead of "rm" (Fig 4). But consider why this doesn't work: all that happens is the "while" loop keeps reading the first line of the "killem" file, and never exits. But why? Well, the line loops between "while" and "done" and each time it comes back to the "while", it reconsiders the proposition of opening the file called "killem" and reading it.

This is exactly what we've told it to do and is why only the first line gets read. If you want to feed the file to the loop, you have to take the file out of the loop (Fig 5).

This gets the job done, but in Unix there's usually a simpler method just around the corner. You can extract your filenames from the list directly with the "cat" command and what you want now is some way of laying out these filenames so that rm is able to read them. This calls for an often-used trick known as "command substitution".

We'll use rm -f to "force" deletion so we're not interrupted by any prompts, and substitute the command to create the



Help or hindrance?

Our Unix chap has some trouble with CHAP and is forced to encounter the BT Internet helpline. Chris Bidmead recounts his efforts. And, caught red ANDed? A return to procmail.

UNIX is a user-friendly operating system; it's just choosy about its friends. That's often taken to be a joke but I actually think a modern UNIX implementation like Linux is more ready to yield up its secrets to the first-time user than other, more familiar operating systems. The core philosophy of UNIX is to have no secrets (from authorised users). Windows, on the other hand, positively goes out of its way to conceal what's going on behind a sometimes completely impenetrable façade of good-looking screens.

I was reminded of this when I tried to make use of the guest account offered to me by BT Internet. I wanted to connect via my NeXT machine, but BT Internet insists you use its own client software, at least for the initial registration, and, its client software only runs on... you guessed it... Microsoft Windows. (Oh, and Apple Macs.)

Regular readers of this column will remember that, last year, we went through a similar routine with another internet service provider (ISP). The ISP in question was using PAP (Password Authentication Protocol) to validate user names against passwords. But the ISP's help desk didn't understand the principle of this simple cross-platform protocol and could only talk me through the steps necessary to set it up on a Windows machine.

This involved mindlessly filling in replies to dialog boxes, which gives you no overall picture of what you're actually doing. The

data is then squirrelled away inside your system — securely? Retrievably? Capable of being backed up? Who knows? Certainly not the people on the help desk. And *voilà!* it just works. Or maybe not, in which case you will be invited to reinstall the software and start again.

It was odd to find myself going through all this again with BT Internet. Perhaps I'm naïve, but I was astonished that a company with the global reach of BT seems not to appreciate that the internet is a cross-platform interconnect, not just a way of getting Windows machines to talk to one another, or occasionally to Macs.

There was nothing for it but to log in initially from a Windows machine. I used the Siemens-Nixdorf portable, which is currently set up to dual boot into Windows 95 or Linux. I thought it would be simple, once I had registered and acquired my user ID and password, to transfer those across to my NeXT machine. Silly me.

BT Internet uses CHAP, or Challenge Handshake Authentication Protocol, to validate its users. CHAP itself wasn't the problem, as you'll see, because CHAP is a core part of PPP and any properly implemented PPP daemon ought to know how to deal with CHAP challenges. PAP works like an ordinary username-password login procedure and is therefore vulnerable to anybody who may be listening in. CHAP is more cunning. As a challenge it sends a randomly generated string of characters,

agreed password specially encoded against the randomly generated string I've just received. The thing that makes CHAP particularly secure is that this challenge and response can be carried out at any time during the connect period.

You might think this all sounds complicated to implement. In Windows it was. I'm not quite sure where, when or how the CHAP stuff got set up but it was all part of a lengthy initial login and registration which, as it happened, didn't go at all smoothly. When I first installed the BT software it apparently took a look at my hard drive's dual boot arrangement, disliked what it saw and decided to fix a few bytes for me. On rebooting, the machine simply hung and refused to come back to life until I revived it with a DOS system disk. FDISK revealed that the Boot Manager partition was mangled.

OK, I fixed that, and at last I'm talking to a BT Internet web registration page that asks me my corporate ID, my personal registration code, my user ID and my mother's maiden name (yes, really!) and finally returns a number of variously mysterious strings. These are stuffed away inside Windows, so that next time I log on I'm automatically authenticated.

Now, to set up CHAP on the NeXT machine. Happily there's a standard UNIX way of doing this and it's simple. Operating as root, you create a file called chap-secrets, under /etc/ppp. It's a text file (so make sure that only root has read and write permission on it) and it's laid out like **Fig 1**.

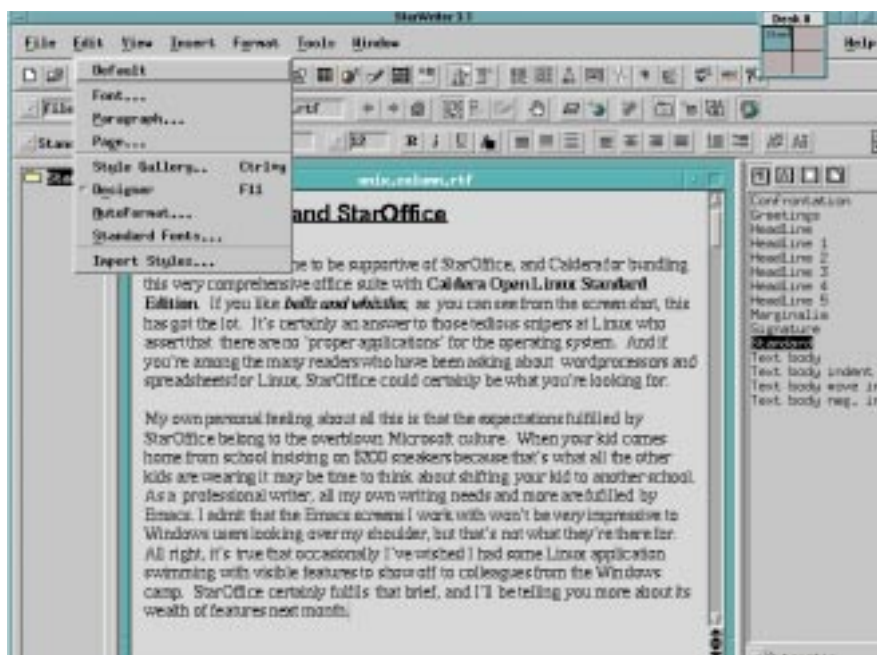
The first line is a header (as with most UNIX config files, the # makes it a comment) just to remind me what the fields mean. There can be any number of lines below this, one for each CHAP-protected server I

p232 ➤

Fig 1: Creating "chap-secrets"

```
# client  server  secret  address
nextmachine  LondonT-NAS9  "Ice Cold in Alex"
```

and its own hostname. If I'm trying to log in, my correct response is to return my own hostname together with — and this is the cunning part — our



Caldera's OpenLinux Standard version has arrived in my office. It is bundled with the StarOffice suite. This is the word processor, StarWriter. I've incorporated my own comments into this screenshot

Windows-to-Mac, service.

Careless colons in Procmail

I'm afraid I haven't left room to fulfil my promise of more on procmail this month, but I must make space here for a warning about what I wrote last month. The warning (which shouldn't be necessary for seasoned readers of this column) is RTFMP: read the flipping manual pages. Regard my remarks here as mere aperitifs to a proper digestion of the authentic facts the manual pages offer. For instance, I suggested that multiple rules in a procmail recipe are ORed. They're not. They're ANDed.

I was somewhat careless with my colons in referring to the 0: flag when I meant the :0 flag. Procmail wants to see :0 as an indicator of the beginning of a new recipe, but that trailing colon, omitted in the recipes I quote from my own .procmairc, is worth explaining because it has a special meaning.

In a multitasking operating system like UNIX you may have multiple instantiations of procmail running simultaneously. What happens if several procmail processes find themselves writing to the same mail file at the same time? Garbage is what happens. The classic way to avoid this is to have the first procmail process that opens the file put a lock on it. If procmail starts writing to a file called "mymail" you can have it create a temporary file called "mymail.lock". This only has meaning while procmail observes the rule that it won't try to open any file called <something> as long as there's also a file called <something>.lock.

Procmail is written to work like this if you so choose. To kick this into action all you need do is open your recipe with :0: instead of just 0:. It's as simple as that. So why don't I use that trailing colon in my own .procmairc on the NeXT machine? I'll tell you why not, and more, next month.

intend to log in to. Currently there's only one, the BT server, which returns the string "LondonT-NAS9" as its server name. I'm "nextmachine", so the first two fields should be self-explanatory. "Ice Cold in Alex" is my password, or more accurately, the secret string shared between my system and LondonT-NAS9. The address field at the end is optional — I can put the dotted quad address (of the form "192.168.1.4") of the remote server there as a belt and braces security measure, if I happen to know it. I don't, so I'll leave it blank.

There are, as it happens, several problems with this. Firstly, whatever BT Internet knows me as, it doesn't know me as "nextmachine". I need my BT username in that first field. The registration process has returned several strings that might be my username and there's no way of telling which is the correct one. I'll have to ask BT Internet's help desk about that.

The second problem is that the server only returns LondonT-NAS9 as its name the first couple of times I contact it (I can see this, because I have PPP debugging switched on to show me the exchange of strings). Subsequently it shows up as LondonT-NAS12, or some other number, indicating that I'm actually talking to several servers. A CHAP system really shouldn't do this. One solution, to keep things secure, would be to set up a separate line for each of BT Internet's various servers. So I need to ask the helpline how many I'm likely to run into, and what their names are. In the meantime, there's a useful cheat I can do. I put a "*" in that field and it will match any server name.

Field three. Another problem. Which of the many strings returned to me during the registration process is my "chap secret"? Is it the original password I chose? Or a hashed version of that password which ended up lodged in one of the Windows dialog boxes? Or (ah!) could it be the hex string labelled "password" that I discovered in a Windows config file called "pctcp.ini"? I needed to ask the BT helpline about that.

I rang the BT Internet so-called helpline with these three simple questions and they had no idea what I was talking about. They said they didn't support UNIX. I said I didn't need any UNIX support — I have a fully working UNIX system, thanks. What I needed was support for their implementation of CHAP. They didn't know anything about that. Would it help, I suggested, if I sent the questions as email? Yes, they replied, good idea. So I did. And here is their response: "Hello. I am sorry, but we only support our software on Win95, 3.1 and Apple Macs. As we do not support it we do not get trained in that area so we are unable to help you."

As it happens I have figured out how to connect to BT Internet from UNIX, but I won't give you the details here. At this stage, encouraging readers of this column to adopt BT Internet as their service provider would clearly put an intolerable burden on BT's help desk. What I will do, though, is talk to some senior people at the company and try to understand why it calls itself an ISP when all it feels qualified to offer is a Windows-to-Windows, or at best

PCW Contacts

You can email **Chris Bidmead** at unix@pcw.co.uk.

Due to space limitations, the Unix column has been reduced to two pages instead of three. Please email any comments to ellie@vnu.co.uk.



Mail order

Don't chuck your old email messages – they are a valuable information bank. If you need to keep them for future reference, Chris Bidmead explains how to do it in an orderly manner.

If your mailbox is anything like mine, it's being increasingly polluted by advertising spam.

Even without this dross, you've probably found your mailbox becoming more unmanageable as the volume of traffic increases. The simple solution is to delete all incoming mail as soon as you've read it. But if you do this, you're often throwing away valuable information. I've found old email very useful as a way of tracking ideas, people and events, and with hard disk prices as low as they currently are, I like to keep everything. The trick is to organise it.

I'm assuming in what follows that you've already got a working mail system. If you haven't, check out man -k mail to discover the elements you'll need to swing into action. For most of us this will imply something like pppd to connect to the outside world, a mail transport daemon like sendmail or qmail to manage that connection, and a mail client like elm or pine to act as the user front-end.

The system I use most, on my NeXT machine, has a graphical interface called Mail.app which seamlessly handles rich media like voice and pictures. It looks exotic (Fig 1) but it's much the same Unix setup we're discussing here.

A typical mail client stores all incoming mail in a single "mailbox", actually a file, or in the case of NeXTMail, a hierarchy of directories. Most mail clients also allow you to shift email across a selection of mailboxes and you can use this to do manual sorting. Some come with a built-in filtering system allowing you to automatically direct incoming mail into different mailboxes, depending on the various criteria you set up. You may be able to achieve something similar by messing around with

your mail transport daemon, but I wouldn't recommend it. The best way is to leave the filtering to a specialist tool, as you will see.

For an overview of filtering, take a look at the Filtering Mail FAQ maintained by Nancy McGough <www.ii.com/internet/faqs/launchers/mail/filtering-faq/>. Procmal, a marvellous piece of free software conceived by Stephen R van den Berg <srb@cuci.nl> is the most powerful mail filter available, and here's a simple run-down of how I use it. The idea is to interpose procmal between the mail transport daemon and your mail agent. There's more than one way to do this, but the simplest is probably to use the mail transport system's automatic

forwarding system. With sendmail this consists of a control file called ".forward" which is kept in your home directory. One standard use of .forward is to invoke a utility called "vacation" that sends automatic replies to messages when you are on holiday, but you can also use it to redirect your incoming messages into procmal before your mail client gets to them. My .forward file contains a single line:

```
"IFS=' '&&exec /usr/local/bin/
procmal -f-|exit 75 #bidmead"
```

The line is processed by the shell and it first makes sure that the internal field separator (IFS) is a space, then executes procmal with the -f parameter (which in this

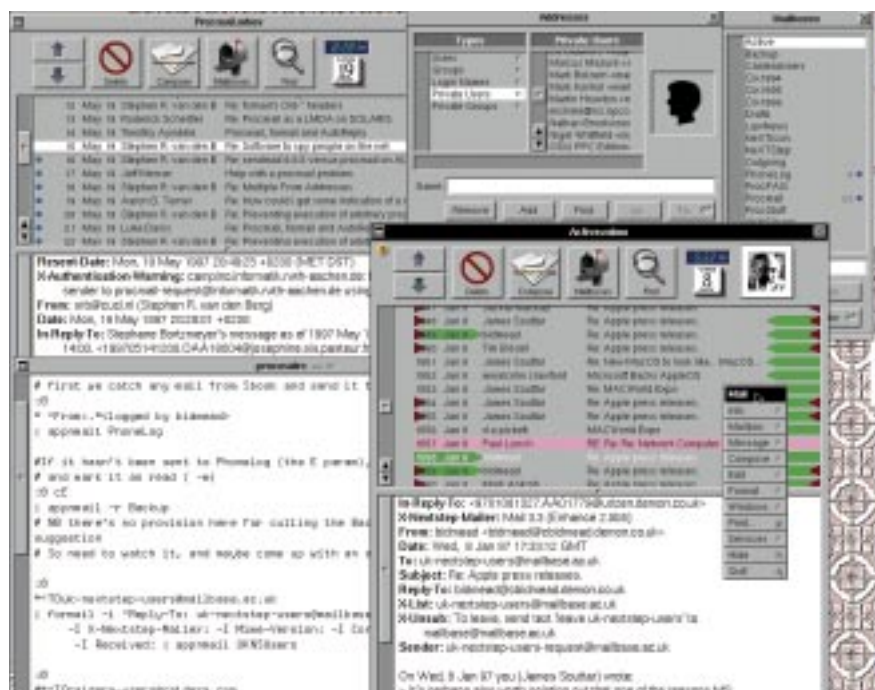


Fig 1 NeXTStep's luxurious Mail.app, augmented with some free add-ons (EnhanceMail and Colorizer) becomes a very useful information filing system. The (invisible) addition of procmal automates the sorting of incoming mail into multiple mailboxes

case effectively guarantees there's a From line in the header of the incoming mail, as required by most mail clients). The "||" that follows sets up a graceful exit if procmail isn't found. The #USERNAME comment at the end is a bit of a puzzle but the procmail manual page encourages its use as "a necessary kludge against over-optimising sendmail programs".

If the Unix-ness of this line seems somewhat overwhelming, don't worry. I copied it verbatim from the procmail manual pages and have been using it for months.

When procmail springs into action it looks for a file called .procmailrc, also in your home directory. Once you've set it up, this .procmailrc control file is the only thing you need to mess around with. The file consists of two parts. The first (which you'll probably set once and for all) establishes some environmental variables (Fig 2).

The second part comprises all the

Fig 2 Establishing environmental variables

```
PATH=/bin:/usr/bin:/usr/local/bin
MAILDIR=$HOME/Mailboxes #this directory must exist
LOGFILE=$MAILDIR/procmail.log #where procmail writes its log
```

redirection rules that you're going to set up for the various mailboxes. You can have as many or as few of these as you please and the rules can be as simple or as complicated as you like. The first simple rule I wrote for my system instantly repaid me for the work of installing procmail. It looks like Fig 3. The first line is a comment. SBook is a NeXTStep phone dialler that optionally logs phone use by sending an email for each call (Fig 4). Without a filter this is just messy; with procmail catching and filing the output, it's a joy. The second line announces that the following lines are a list of rules, or "conditions", as the manual calls them. I use a single rule here but you can have as many as you like and they will be connected by "or" operators.

The asterisk in the following line characterises it as a rule, which in this case can be read as "look for 'From:' at the beginning (^) of a header line and test whether what follows includes the string '<logged by bidmead>'". If this rule succeeds (and only if) the action line (the first of the following lines that don't begin with *) is carried out. In this example, the action we

trigger is to pipe the output from procmail, the message in question, through a utility called appnmail into the appropriate mailbox. Appnmail is a special NeXTStep

tool created by NeXT guru Carl Edman specifically to link NeXT's mail client,

Mail.app, with procmail. It will create the target mailbox if it doesn't already exist and will parse any multimedia content the message may hold, preparing it to be read by NeXT's Mail.app. In a non-NeXTStep system you would omit

the "| appnmail" and just supply the name of the target mailbox. But as the appnmail example illustrates, a power user could use the action line to carry out any number of activities. The whole block, condition plus action, is called a "recipe". Recipes can be clever without being much more complex:

```
:0 c
| appnmail -r Backup
```

Here, we're taking the elementary precaution of backing up all incoming mail that hasn't come via SBook (which has already been diverted), so we send it to a mailbox called Backup.

Fig 3 A simple redirection rule

```
# catch any mail from SBook and send it to the PhoneLog :0
* ^From:.*<logged by bidmead>
| appnmail PhoneLog
```

Fig 4 SBook call logging

An SBook call logging email looks like this:

```
Date: Tue, 22 Apr 97 16:52:09 +0100
From: Mike Pritchard, Euromation <logged by bidmead>
To: bidmead
Subject: Call received from Mike Pritchard, Euromation
Call received from Mike Pritchard, Euromation at Tue Jul 22 16:51:16 1997
Mike says that Euromation will be delighted to fly me to Cannes at their expense if I promise to write a glowing article about their products. I graciously declined.
Duration: 04:55
```

The simple "To:" line (omitting my full internet address) ensures that the email stays local and isn't relayed out through my service provider. The "From:" line shows the (alas, in this case, entirely fictional) originator of the call. The timeline is added automatically by SBook and the note that follows is made by me at the time of the call.

To make sure this is caught by procmail and sent to SBook's own mailbox, I need to look for a feature in the headers that is unique to an SBook-originated email. This turns out to be the "<logged by bidmead>" part of the "From:" line.

As we saw with the SBook recipe, processed mail normally won't fall through to the next recipe. But this time, we don't want to intercept the mail, just make a copy of it. This is achieved by the "c" parameter that follows the "O:" flag. (Appnmail, which by default marks the mail it handles as unread, needs a -r parameter so that the copied mail in Backup doesn't continue to clamour for our attention.) With just those two recipes in place, all my incoming mail is copied to Backup and falls through to my standard mailbox. Now it's safe to experiment by appending some new

```
:0^T0caldera-users
| formail \
-I X-Nextstep-Mailer: -I Mime-Version: -I Content-Type: \
-I Received: | appnmail CalderaUsers
```

recipes. Let's capture all that useful traffic from the caldera-users' mail list [above]. The recipe introduces new ideas which we'll look at next month. Meanwhile, you can join the procmail mail list by mailing procmail-request@Informatik.RWTH-Aachen.DE with the word "subscribe" in the header. It's busy, so you might need this:

```
:0^T0 .*procmail@Informatik.RWTH-Aachen.DE
Procmail
# or, if you're using NeXTStep.
# | appnmail Procmail
```

PCW Contacts

You can email **Chris Bidmead** at unix@pcw.co.uk.

Due to space limitations, the Unix column has been reduced to two pages instead of three. Please email your comments to ellie@vnu.co.uk.



/bin there, done that

Chris Bidmead throws light on Unix directories — /bin, /dev, etc. It's an informal layout, but you *can* get a feel for how it's put together. Plus, there's ping, SCO, and the great awk.

Last month I described how the XFree86 configuration keeps its information in a file called XF86Config, which is usually to be found in the /etc directory. When you're just getting started with UNIX, finding your way around the various directories can be confusing. A number of them have been set aside, by convention, for more or less specific uses, but the catch is that the set of directories and their uses tend to vary between different flavours of Unix. So, as for much else in this column, what follows isn't prescriptive, but is to help you get a handle on how it works.

- Expect to find the standard binary files intimately associated with the operating system itself in /bin.
- Devices (pseudo files which are actually connections to device drivers) are in /dev.
- /tmp is reserved for temporary files.
- The /usr directory will typically contain nothing but a number of other directories, like /usr/sbin, for binaries used for system administration, and /usr/bin, for frequently used binaries that aren't already in /bin.

The distinction between /bin and /usr/bin is fairly arbitrary. You might think that putting them under /usr implies they are the operating system enhancing binaries added by users, but these belong more properly (on many systems) in /usr/local/bin.

The /etc directory sounds as if it might be for everything else, but it, too, has a (roughly) defined role. These days the /etc location tends to be the place where all the configuration files are kept. But for historical reasons, /etc is also where you'll find networking utilities like ping and ifconfig. Oh, and the executable init scripts are probably in there, too. Yes, it's a bit of a mess, but it's a widely understood mess!

More recently, there's been a drive to formalise this directory tree more strictly, but in the meantime hold on to the idea that there is at least some rhyme and reason behind the layout. You get a feel for where the various files are lurking, and you can always supplement this with the system search utilities /bin/find and /usr/sbin/whereis. Or is that /usr/bin/find and /bin/whereis?

Sign of the times

I may not get the chance to reply to all your email (I do try) but feedback from you is definitely the making of this column, so please keep it coming.

There is, however, one kind of email I really don't like getting. No, I'm not talking about abusive email (rare, I'm glad to say) but the all-too-common mailing that expects me to do all the work and doesn't give me anything to go on.

Here's an example: *"I'm trying to connect my new Windows 95 PC to our IBM AIX database server for client-server access. The IBM is expecting a TCP/IP connection to service the client-server request. I've set up my PC with Microsoft's TCP/IP as one of its protocols (using Network Neighbourhood Add & Properties). What do I need to do now to establish a connection?"*

My first question is, who are you? The email isn't even signed. I had to fish about in the headers to discover the name, Ian MacDonald. What does it cost to spend a line or two introducing yourself? For example, "Hi, Chris, I'm a nuclear physicist at Cern, and we're using FreeBSD to control our particle accelerator. I happened to pick up a copy of PCW..."

Another point that is almost mandatory in mailings, not just to me but to

newsgroups or whoever (and we talked a bit about this last month) is to indicate that you've gone some way towards trying to solve the problem yourself and you're not just leaning back, expecting someone else to do all the work. Even if you were, you'd need to define the problem rather better than Ian does here: *"The IBM is expecting a TCP/IP connection..."* doesn't help much. If there is such a connection and nothing's working, you have one set of problems. If you haven't managed to make the TCP/IP connection at all, that's a different problem.

The power of ping

The way you test whether a simple TCP/IP connection exists is to access the AIX machine's IP address from the client machine using the elementary utility called "ping" which should be available on any TCP/IP setup, and is certainly there in Windows 95. Now, I acknowledge there's a real problem for Windows 95 users: if they don't happen to know about the existence of ping (and why should they?) it's very hard to find out about it. Despite the extensive so-called "Help" Microsoft offers, there's no way (that I can see) for the complete novice to get from "network" to "ping". Contrast this with any decent UNIX operating system, where entering "apropos network" at the command line will return a list of associated commands: a little research among these will lead you to ping.

So ping may be all it takes to diagnose Ian's problem, or it may not. Who could possibly tell on the basis of the skeletal information provided?

SCO — sorted

By way of contrast, here's the sort of email I much prefer: *"I read your column regarding*



Paul Rowlands' email mentions the HTML documentation that comes with SCO Open Server. Here's what it looks like. The help screen is a simple web browser (I've added Netscape 2.0.2 to my own Open Server setup, downloadable from the SCO web site) and the underlying HTTP server seems to be Apache, so this is more or less the same setup you get with Linux distributions these days

the problems you were having running SCO and DOS on the same (Apricot) system: seems like you had more trouble than I did ... I have SCO OpenServer 5 (free), NT4, and DOS 6/Windows 3.11 all on the same system ... My only problems are:

- 1. When I boot, I have to use fdisk to set either the Unix or the two Microsoft partitions active. The NT loader doesn't want to know about Unix.*
- 2. SCO doesn't support my graphics card (a Number 9 Reality 332) so I have to run it in 640 x 480 mode — a bit grim, since it can support 1024 x 768 x 32K colours under NT and Windows...*

"Another gripe about SCO is that it seems impossible to feed back problems or get support. I know the product is free (but for a 'small handling charge'), but what is it there for? Is it to encourage me to buy it in my professional capacity?"

"Have you noticed that SCO comes with an HTTP server already configured and running (dunno what particular flavour it is, but it contains O/S documentation)? Anyway, all the best, keep it up — we need a non-Microsoft perspective on the world."

Paul Rowlands

Paul also provided details of his hardware and made a few other useful points about SCO, which I've omitted for the sake of brevity. This is the kind of mail to which I can respond. I suppose the most

serious point is the one about SCO's unapproachability. The good news is that this shouldn't be the case for readers of this column because Dave Gurr, marketing development manager at SCO UK, has been very helpful to me over setting up Open Server, and he's allowed me to publish his phone number and email address (see "PCW Contacts", p269) for any of you who need further help.

Please remember that the same rules (about which I've been writing) apply here, too: do your homework first. You'll find help about Open Server installation at www.sco.com, so please investigate here before contacting Dave.

As luck would have it, several other readers have written in with help for the problem about needing fdisk to switch between bootable partitions. My apologies for not having spotted this myself, but SCO has actually included some elementary dual boot software in the distribution. Robert Warner robert@softdesign.demon.co.uk discovered this when setting up a disaster recovery server that used two partitions, one for SCO and one for Windows 95.

He says: "I installed SCO first, having the usual drivers problem because I was using a non-Adaptec SCSI controller..."

(Minor technical carp, Robert. These aren't controllers, they're host bus adapters. With SCSI, the controller is embedded into the drive device itself.)

"...Once I had that up and running, I proceeded to install Win95. This all worked fine except that when I booted the PC, it went straight into Win95 instead of getting the SCO boot prompt..."

This is because Windows 95 assumes that it is the only operating system on the disk, and on install sets its own partition as the active partition. Robert fixed this using fdisk to make the SCO partition active. This gives you the SCO Unix boot prompt when you switch on, and if you just hit return at this point you'll be booted into Unix.

"But," writes Robert, "if you enter 'dos' and press return, it will boot from the Win95 partition. You get the best of both worlds."

That weasel word, "Open", again

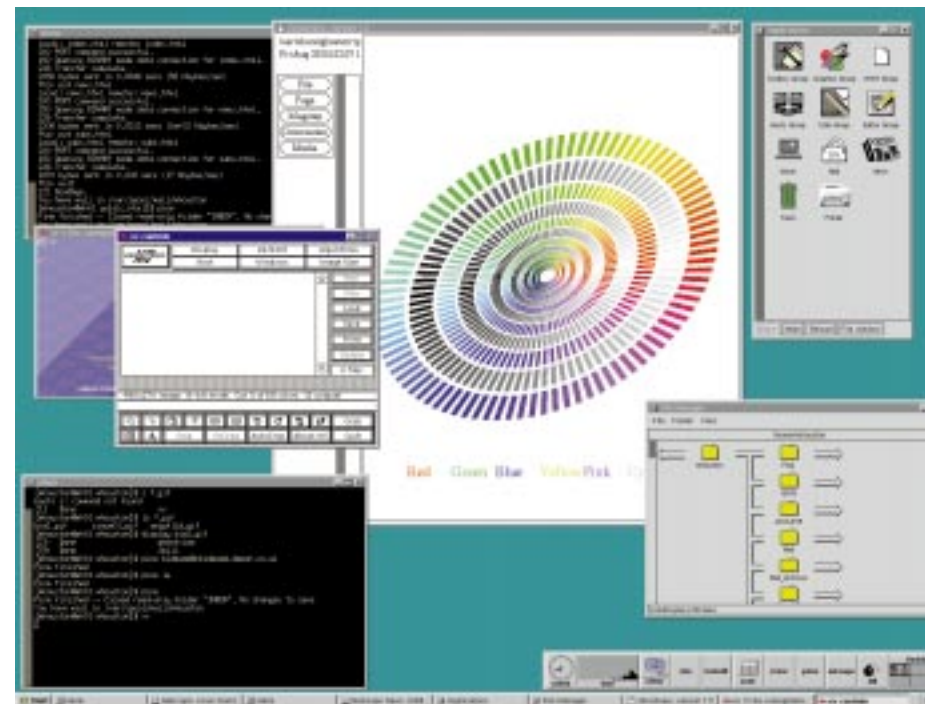
A couple of months ago I quoted SCO's Michael Tilson, who was complaining that: *"The term 'open' has been hijacked. Because 'open' is good, everyone labels whatever they sell as 'open'."*

It's a point I had raised with Caldera when it first came out with OpenLinux™ at the beginning of this year. But more recently, I found myself becoming increasingly irritated by this new development because of a supposed Caldera product called "OpenLinux Lite".

My initial objection to "OpenLinux" was that it capitalises on an established (non-proprietary) brand, but somehow manages to suggest that it is more "open" than Linux. Depending on your interpretation of the word "open" you might not accept this. A plain Linux implementation like the RedHat one, on the PCW CD a couple of months ago, can be shared and copied without restriction; Caldera OpenLinux can't. So which is the more "open"?

Well, the argument goes: Open with a capital "O" refers to adherence to standards and is not about whether or not you get the source code, or a reference to unrestricted redeployment. The new Caldera distribution is being aimed towards full UNIX compliance according to the specifications laid down by The Open Group. This is something most Linux users don't care about, so in this sense their plain old Linux is arguably less "Open" than Caldera's.

The OpenLinux Base product that currently sells for about £55 includes the Looking Glass desktop, the Netscape Navigator 2.02 web browser, Metro-X's proprietary accelerated X-Window system, the CrispLite text editor and other stuff. The version stripped of all this, and so freely



More adventures with awk

As regular readers know, I like "awk". It's a good, simple, general-purpose language and much better than BASIC, in my view, as a way to get started in programming. It also comes free, in the shape of GNU's "gawk". The awk examples we've had in this column in the past have always required you to evoke gawk explicitly and add the program file and datafile as parameters, like this:

```
gawk -f progfile datafile
```

Recently, I decided I wanted to write an awk program as a utility that you can run directly as an executable in its own right. With shell files (written in csh, bash or whatever) the trick is to use what's called the "shebang notation". You do this by adding a magical first line to your script that tells the system where to find the language file that's going to run it.

```
#!/usr/bin/csh
```

The "#" (pronounced "hash" or "sharp") indicates that the line is a comment, and the immediately following "!" (or "bang") triggers the system magic. Most modern UNIX systems support this.

You can do the same thing with an awk script but what happens next isn't what you want: gawk loads and then tries to treat your script as if it were the data file. The trick (not immediately obvious from any of the documentation) is to use the -f indicator, just as you would at the command line.

Here's a skeletal example that also collects parameters from the command line:

```
#!/usr/local/bin/gawk -f
```

```
# How to pass command line params into gawk using ARGV, ARGC
```

```
# The ARGV array starts at 0, but that value will be "gawk"
# So we start at ARGV[1]
```

```
BEGIN {for (x = 1; x < ARGC; ++x)
        print ARGV[x]
        print "There were " ARGC - 1 " arguments,
not counting ARGV[0]"
}
```

There's an amusing catch to this, though, if you're also trying to pass one or more data files into gawk from the command line, and I'll deal with that next month.

This screenshot, supplied by Martin Houston, organiser of the Linux branch of the UK UNIX User group (www.mh01.demon.co.uk) shows RedHat Linux running the latest fvwm97 window manager (the resemblance to windows from another well-known operating system is not accidental). You'll find a number of optional Linux interfaces at www.PLiG.org/xwinman/

distributable is, as I understand it, to be called "OpenLinux Lite". This seems to me to imply that it's somehow a cut-down version of Linux, which it isn't.

Perhaps a more serious objection to OpenLinux Lite is that despite promises made at the beginning of this year, no mention of it has yet appeared on Caldera's web page and it doesn't seem to be obtainable (although you can download OpenDOS, Caldera's own version of DOS, bought in from Novell). I hope this will have changed by the time you read this. You can check www.caldera.com to find out, and if it still isn't there you can write to nancy.pomeroy@caldera.com to ask why not.

By way of contrast, I notice SCO has made good its promise to release a free single-user version of UnixWare on much the same basis as its currently very popular (to judge from your emails) and free Open Server. Find details of both operating systems at www.sco.com/offers/index.htm.

Why two operating systems? SCO is the largest commercial vendor of Intel-based UNIX systems, and when Novell's efforts to make a decent business out of UNIX fell apart a couple of years ago, SCO stepped in towards the end of '95 and bought UNIX from them. (At a bargain price, by the way: SCO needed to bring its own ageing version of UNIX up to date and paid Novell \$60m for something that Novell had paid AT&T \$320m for two years earlier!)

SCO's intention is to combine its Open Server with Novell's UnixWare, and the result of this will be appearing later this year as "Gemini". The free single-user version of UnixWare looks like something worth investigating. I've put in my own request for a copy, so expect news in this column soon.

PCW Contacts
Chris Bidmead is a consultant and commentator on advanced technology. Email him at unix@pcw.co.uk.
SCO UK, Dave Gurr 01923 813548; email davidgu@sco.com



The joy of X

Chris Bidmead tells how he got X up and running on his Scenic Mobile 700 notebook, accepts praise on his technique, and gets down to some heavy vetting of Linux books.

Of course, it may be that you've all been well brought up, but a great deal of the email I get these days begins with burbles of praise for this column. And a frightening number of you are also writing in to tell me you've been encouraged by my ramblings here to junk Microsoft Windows and install one or other of the UNIXes, or at least create a dual boot system. For once, words fail me. I don't know how to express the sense of excitement, trepidation, pride and responsibility this gives me.

On the other side of the coin are the people who tell me that Windows is "good enough" and "if it ain't broke, don't fix it." I'm sympathetic about this too. Microsoft has done a remarkable job of making computing accessible to everybody. And if by contrast you're left with the impression that UNIX is an unintelligible collection of arcane keystrokes — as somebody recently put it in an electronic conference — I do see your point (and thanks for reading the column this far). My response to this was: if evolution had shied at arcane keystrokes, we'd all still be rocks.

Martin Diedrich, from the Department of



Cross-platform interoperability isn't just about workstations and servers. I really like the idea of being able to control the network printer from any workstation on the network — security permitting. The Tektronix Phaser 350 has its own built-in web server, so any workstation running a browser can read its status and reset its parameters

Economics at Keele University writes: "Dear Chris, First of all, my congratulations on your excellent Unix column in PCW." (See, I'm not making this up!) "After two years of growing interest in Linux, I have finally gathered enough courage to get started... For various reasons it will be convenient for me to install Linux on a laptop... but being a Unix novice I am worried that I might do damage to my screen when setting up X."

I get a lot of mailings asking about UNIX

and laptops, which is how I got into installing Caldera's latest release, Caldera Open Linux, onto a Siemens Nixdorf Scenic Mobile 700 portable, as I mentioned last month. I was very impressed — and I hope you were too — by the way the install boots straight off the Caldera CD-ROM. It's the first time I've ever been able to do this with a PC-type machine, and tribute must go to Siemens Nixdorf for implementing this feature into the BIOS, and to Caldera for

CBOR — Chunky books ooze reassurance

The day that Windows NT finally established itself, I remember thinking at the time, was the day that the Windows NT Resource Kit arrived — three chunky volumes accompanied by a CD-ROM. It doesn't matter what my views of Windows NT are, I recall thinking, or what I know about Microsoft's support for its new baby ("Problems? Have you tried rebooting? Ah, OK, then the best thing is to reinstall..."), or indeed whether Windows NT fulfils its promises or not. The three chunky volumes ooze the kind of reassurance that is exactly what corporate customers need with a product like this. Microsoft knew this, of course, which is why the Microsoft Press produced them.

At the time I never believed that even its greatest fans could feel the same way about a freely distributable operating system like Linux. Sure, there's a ton of documentation out there on the web or buried inside the installation CDs. But these are nerdy monographs with spellings like "kernal", not glossy volumes that sit on your bookshelf glowing with confidence.

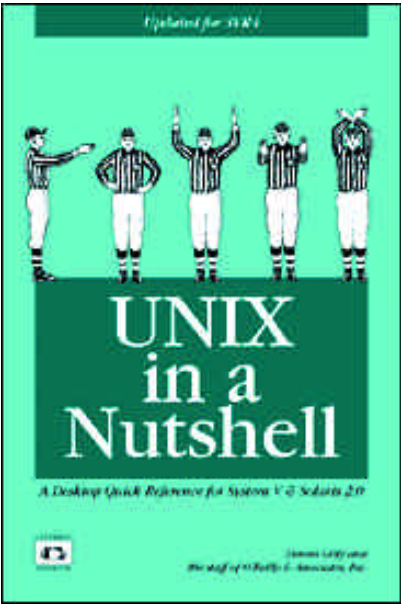
All that has completely changed now. Solid, informative books about Linux abound. When readers ask me, what books should I buy to get started, I still respond that the best way in is probably to hold off from buying books and get stuck in to the docs on the disks. But once you're through that, or if you really don't like reading on-screen, the book scene waiting for you is a toothsome banquet. Which ones to choose?

The starting point is *The Unix Philosophy* by Mike Gancarz, from the Digital Press, which asks — and answers — the rarely faced question "Why UNIX?". When it comes to "How UNIX?" my favourite was always *Running Linux* from O'Reilly, and there's a new edition out now. Supplement this with *Linux Network Administrator's Guide* if you're going to get hairy with networks. This too is published by O'Reilly, but it's also part of the Linux Documentation Project so you can download it from the web or consult it on-line as necessary.

But the must-have book for me has always been *UNIX in a Nutshell* (yes, that's from O'Reilly again!). It covers all the basic commands but manages to be more than just a command dictionary, finding plenty of room for worked examples and illuminating commentary.

There are several versions of *UNIX in a Nutshell* for different flavours of UNIX. I favoured the System V version but it had surprising (to me) omissions. When I was first struggling with the mount command I was alarmed to find no mention of it in the book. A seasoned UNIX jock patiently explained to me that this was because it was a user's manual, and mere users had no business messing with system commands like mount.

Well, there's now a (rather fatter) version of the Nutshell book specially for Linux, and, yes, mount is in there. This new version recognises that most Linux users are also going to be their own system administrators, so there's now a complete section on System and Network Administration at the back of the book. There's also a rather breathy introduction with headings like "The Excitement of Linux" which, while not inaccurate (sample: "Linux revives the grand creativity and the community of sharing that UNIX was long known for...") may help to obscure the point that, largely thanks to the weighty endorsement of books like this, Linux is clearly ready for prime time.



chat with colleagues and some recent experiments of my own, is that you just don't get to fry a modern LCD display with mere software.

Although X wasn't working on the Scenic, at least Linux was all in place. If the character-based consoles are as far as you're ever able to get with a particular UNIX installation, please don't despair. For the first four months after I introduced Linux to this column a couple of years ago, I couldn't get X working. That Linux installation wasn't pretty, but it was still powerful. I got a lot of things done with it and, thanks to all the loose documentation, the manual pages and the built-in Info hypertext system, there was plenty to keep me busy and keep me learning until I worked out how to get X going.

This time I didn't have to wait that long. X is now up and running on the Scenic (I'm writing this using XEmacs on the machine), and I'll tell you how I did it. Originally I anticipated filling the next 50 paragraphs with a detailed technical description containing a lot of example data like

640x480 @ 60 Hz, 31.5 kHz hsync					
Modeline	"640x480"		25.175		
640	664	760	800	480	491
493	525				

with some heavy discussions about dot clocks and horizontal and vertical sync frequencies. This column isn't afraid to venture into tough territory, but Bidmead's Law of Hard Work states: don't do it if you don't have to. Some of you may regard this Chronicle of How I Got X Going as something of a cheat. So be it. This is how it goes with Linux in real life.

Step one was to ignore any possible difficulties and just go for it. Caldera Open Linux comes with a pair of alternative X systems, the freely distributable XFree86 version and a commercial package called Metro-X. I'm dedicated to the cause of free software, but I also like an easy life, so given the choice I started with Metro-X. Alas, it turned out not to include any support for the Scenic's CT65550 graphics chip. Happily the XFree86 version does; as is often the case, the free software is ahead of the commercial equivalent, and this implementation had the CT65550 covered.

But there's more to a video subsystem than the graphics chip. Other key factors are the RAMDAC and the display. In the absence of any specifics on these I ignored the possible problems and sailed into the XFree86 graphical setup routine I

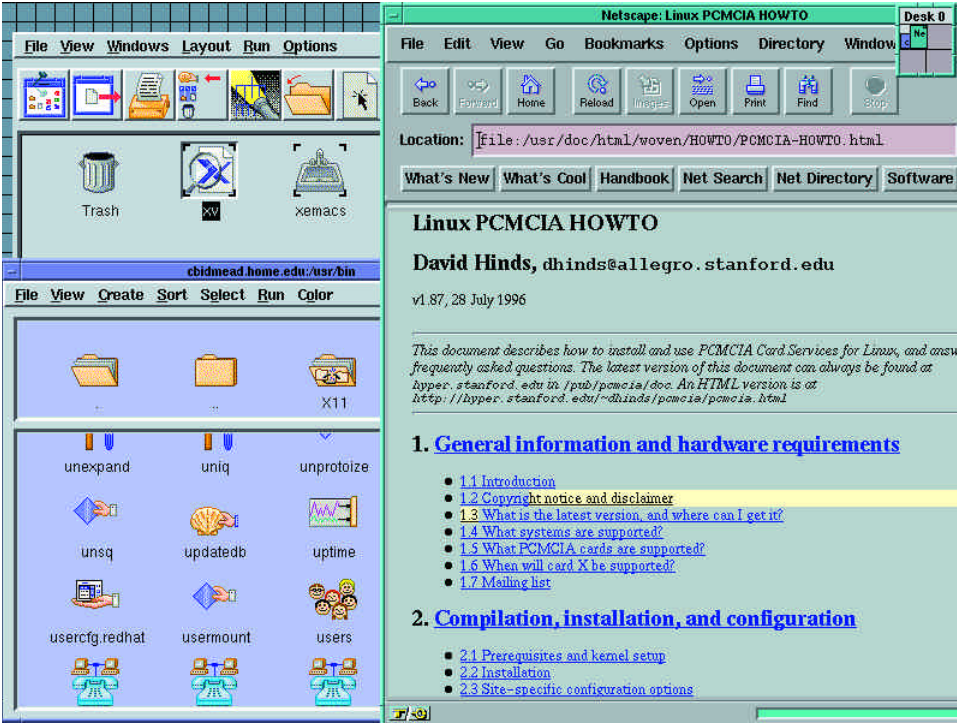
mentioned last month. It's called XF86Setup and it tries to establish an elementary X display straightaway, and then invites you to fiddle with the parameters through a dialog box with buttons and pull-down menus.

The X screen it came up with was about two thirds the size of the Scenic's LCD, which I could live with temporarily. But it was defaulting to the wrong mouse, which meant I couldn't pick my options by point and click. Happily, the keyboard navigation that XF86Setup offers as an alternative worked, somewhat awkwardly, and once I'd fixed the mouse (the touch-sensitive pad the Scenic uses is, conveniently, PS/2-compatible), it was a lot simpler.

With the majority of desktop machines, XF86Setup would probably get you all the way home. But notebook computers tend to be weird. The Linux Laptop Home Page at www.cs.utexas.edu/users/kharker/linux-laptop (or the RedHat mirror at www.redhat.com/linux-info/laptop) is run by Kenneth E Harker, who seems to be working very hard to keep it up to date. It covers a number of the popular machines, but the Scenic Mobile was too new to be on it. I was on my own, and, as it turned out after several hours with XF86Setup, on my own with a display that determinedly remained two-thirds of the size and fizzed a lot every time I wiggled the mouse.

The XF86Setup utility is really just a pretty front-end to a configuration file called XF86Config that sits (usually) in the /etc directory, which is the canonical place for these kind of files. So my next step on the road to The Joy of X was get out of XF86Setup and start mulling over the config file directly. Like all good UNIX config files, this one is in plain ASCII, editable by any text editor provided you're a user with read/write access to /etc, which on most systems implies you're root. Root is always presumed to know what s/he's doing, so the fact that you can dramatically mess up the entire system by tweaking these /etc config files isn't supposed to be a problem. My consolation was that messing up XF86Config could at worst only deprive me of my fizzing, shrivelled X display and leave me at the command line. From there I could at least restore a backup of XF86Config.

Past experience on other Linux systems



Here's another increasingly common use for web browsers — as readers for internal documentation.

The Caldera OpenLinux Base I've installed on the Scenic Mobile comes complete with a large set of HOW-TOs and other documentation set up for easy-on-the-eye reading through the bundled Netscape browser, so I can read up on Linux wherever I am

has taught me that the supplied autoconfig utilities like XF86Setup tend to be a lot smarter than I am. So if they couldn't produce a decent display, I had a lot of experimentation and twiddling ahead of me. I accordingly armed myself with "The Hitchhiker's Guide to X386/XFree86 Video Timing (or, Tweaking your Monitor for Fun and Profit)" by Eric Raymond et al, and those of you who are disappointed that I'm not going to parade the guts of video tweaking in this month's column had better hasten onto the net to procure it, if it's not already in your X11R6/lib/doc/ directory. Speaking of the net, the other (and really smart) thing I did was to put out a Mayday call. Not on the newsgroups, although I did first scour comp.os.linux.x and comp.os.linux.setup to see if this ground had been covered. (An excellent way to do this is to run a search from www.dejanews.com.) Instead I went to the Siemens Nixdorf web site at www.sni.de and found a discussion group set up there for problem logging. The net is full of so-called problem reports that just say something like "I'm having terrible trouble getting X to work on my laptop. Please can anybody help?", so I took the trouble to describe the problem, specifying the model number of the machine and the version of

Linux and XFree86 I was trying to set up.

The response from the Siemens Nixdorf engineers the next day wasn't wildly helpful: it just suggested I contact the UK help desk. From there I learnt that Siemens Nixdorf doesn't support Linux, but at least the help-desk guy gave me some pointers to existing help on the web. Following these up resolved down to "The Hitchhiker's Guide to...etc" so things seemed to be going round in circles.

I was settling down with the Guide when another bit of mail dropped into my mailbox. It came from Heiko Boch, a German computer science student at the Technische Hochschule in Darmstadt, Germany. Heiko had seen my *cri de coeur* in the Siemens Nixdorf discussion group, and was happy to step in and help. His mailing included a ready-to-go copy of XF86Config he'd hand tailored for the Scenic Mobile.

Thanks, Heiko. That's real-life Linux — people turn up and help. Now if I can just get this network card working...

PCW Contact

Chris Bidmead is a consultant and commentator on advanced technology. Email him at unix@pcw.co.uk.



Top hat time

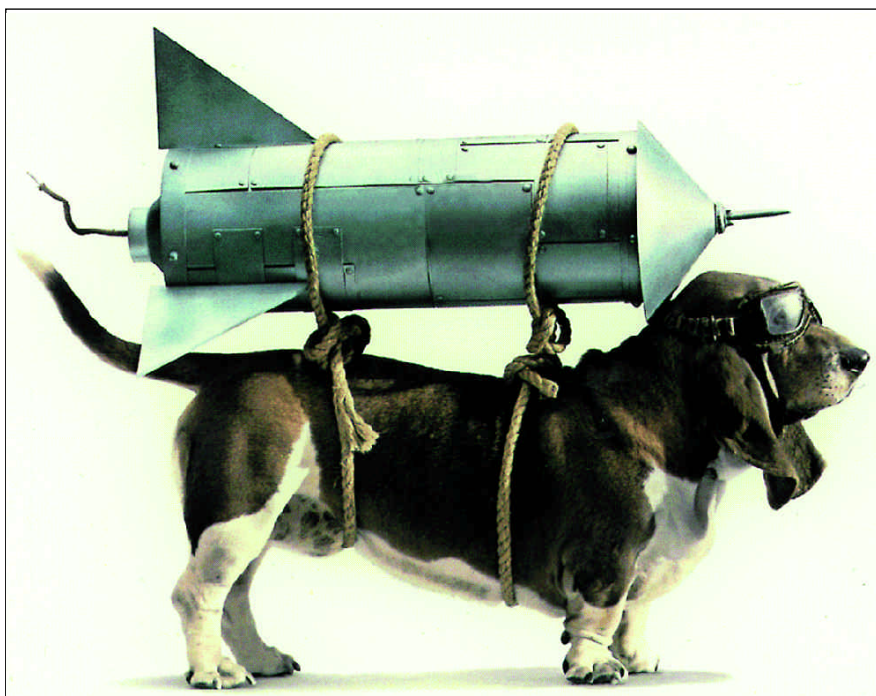
Chris Bidmead looks at RedHat and OpenLinux Base, and installation on a notebook computer using XFree86 3.2. There's the latest RedHat Linux for you on our CD-ROM, too.

One of the frequent themes of your emails to me is the request for Linux on the "front cover". This month you're in luck, because the very latest version of RedHat Linux is there, on our cover-mounted CD-ROM, ready for you to install.

Meanwhile, devotees of other UNIX-like operating systems want to know why I am forever harping on about Linux, at the expense of, say, FreeBSD? No doubt some of you will regard the inclusion of RedHat on our CD-ROM as not so much a positive blow for Free Software or the UNIX life-style, as a kick in the teeth for FreeBSD or NetBSD.

Operating-system wars are not very interesting. My network sports a number of different operating systems: Linux, NeXTStep, AIX, SCO and (as a link into the Microsoft world) Windows NT. And that happens to be just about as far as I can stretch it for the moment. I've nothing against the BSDs, or other UNIXes, and it's always good to hear from readers who have positive and interesting things to say about them. This column is certainly not written on the assumption that you're all running Linux, or that Linux is all you want to read about. Linux, for me, is just one instantiation of a UNIX-like operating system, chosen because it does make a pretty good reference point: it's probably the easiest for anyone to get hold of and it runs on the widest range of software.

To get a broad picture of the free UNIX products available, you might visit www.public.iastate.edu/~free-unix/homepage.html or www.ici.net/faq/unix-for-pc.html. And the free version of SCO on www.sco.com is proving popular, too.



Perhaps not a wildly informative screenshot, but I fell for this picture when UNIX workstation vendor, Silicon Graphics, used it to explain why its own new Octane graphics workstations (starting price \$25,000) are in a different league from anything else you could put together around a PC. This is supposed to represent a PC with a whizzy graphics card installed

There are also many readers who turn to this column to expand their horizons without necessarily wanting to get involved in installing any of this stuff. That, too, is fine by me. If you're among this group, please don't junk the CD-ROM: there is a lot of really useful documentation on it and Martin Houston, the Linux guru who helped put it all together, has fixed it so that you can read most of it from the comfort of your Windows installation, using a web browser.

Probably the best thing on the CD is Matt Welsh's *Linux Installation and Getting Started* book which you'll find on the disc, complete and in HTML format under the

/instguid/ directory. Load the file called `gs.htm` into your browser and you're on your way. This is the way to read it from DOS or Windows, because this copy has been doctored to work with DOS-foreshortened filenames. If you're already running Linux or another UNIX, use the copy at [/doc/HTML/ldp/install-guide-2.2.2.html/gs.html](http://doc/HTML/ldp/install-guide-2.2.2.html/gs.html).

Be aware that Matt's book is a general guide to Linux. For specific details of how to install the Linux from the CD, turn to the instructions that Martin Houston has provided in the folder called "linux.txt". The instructions assume that you'll probably

want to preserve your existing DOS or Windows installation and will show you how to create an additional partition using only existing DOS software and the `fips` DOS utility provided on the CD-ROM. As Martin says, please do read the documentation thoroughly before you proceed, and back up your data first.

RedHat and Caldera

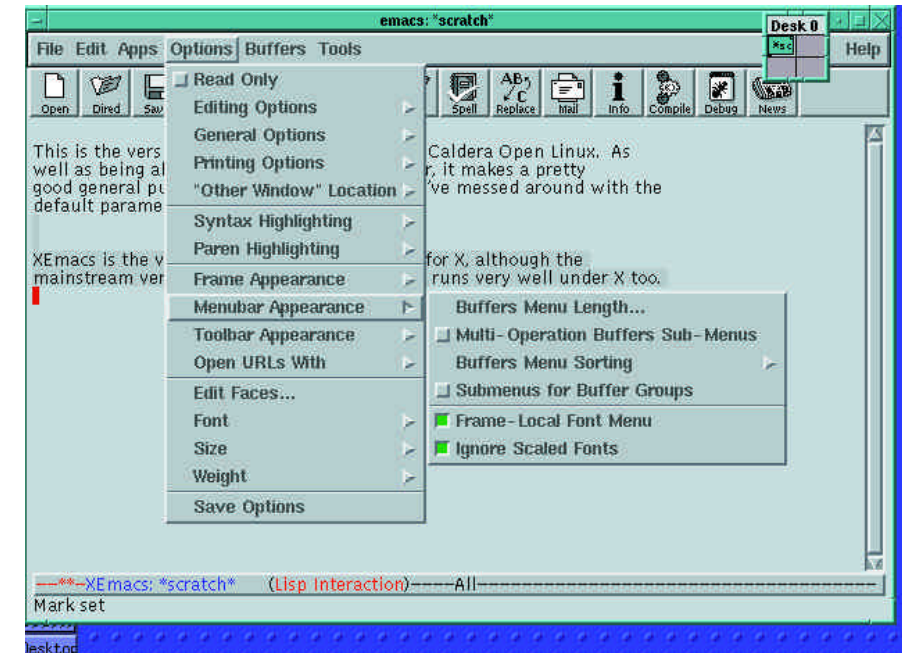
Until recently, RedHat provided the core operating system on which Caldera built the Caldera Network Desktop release. That has now changed, and the new Caldera product is called Caldera OpenLinux Base and it's built around the company's own implementation of Linux. Essentially, it's the same kernel as the one on our CD-ROM but with some additional Caldera features, including a proprietary implementation of PPP (the code used to dial up an internet connection), a commercial X-server and a fancy desktop called Visix LookingGlass. Also included is a licensed copy of Netscape Navigator.

Caldera plans a range of products under the name Caldera OpenLinux (COL) with Base as the entry level. In a month or so it will add OpenLinux Workstation which will include NetWare client software and a secure web server. Later in the year, the range will be expanded to include the Caldera OpenLinux Server; a multiprocessor version with a built-in SQL database intended as a high-powered web site, or as an enterprise intranet server.

I don't suppose many of you "home-brew" readers will be rushing out at the end of the year to spend over \$1,000 on this last item, but equally, I don't doubt that a lot of serious business customers will be flocking to Caldera to snap this up. As a full-blown commercial server it's not at all expensive for what it offers, and it will come with full technical support.

The interesting thing from the point of view of anyone using Caldera products is that they'll all be based on the same industrial-strength version of Linux. And if you're thinking "hang on a minute, you just said the high-end server version will be multiprocessor..." Yes, right. The truth of the matter is that the core of all the Caldera products will be multiprocessor. Bryan Sparks, CEO of Caldera, tells me that if you load any of Caldera's products onto a Quad Pentium box, it will find all four processors and know what to do with them.

You may also be thinking "Yes, but



The new Caldera OpenLinux Base comes with the latest update of XEmacs, the version of the Emacs text editor specially developed for X. I use variants of Emacs for all my writing, and as a totally cross-platform writing tool it cannot be beaten at any price (even though it's free!)

whatever happened to the spirit of free software among all this commercialism? Isn't Caldera just exploiting the labours of some dedicated, unpaid software engineers for its own commercial ends?" Well, frankly, I'm still in two minds about this. Caldera is certainly treading a delicate path in this case and we'll have to see where the company goes with it. In the firm's favour, though, this is probably the place to mention that it will be making available a freely downloadable version of OpenLinux from its web site at www.caldera.com. I understand that this will be stripped of any proprietary hinderances (so you will not get the LookingGlass desktop accompanying it, for instance) and will mostly be covered by the GNU licence, so the source code should be available.

The only exception, I am told, is that the installation routine will be copyright Caldera. This is not to deter you from passing a copy on to your friends once you've downloaded it, or making multiple copies across your own network: the intention is to prevent the code being exploited by certain CD manufacturers who have been bundling multiple Linux distributions into cheap CD sets and cornering the market. There's nothing in the GNU licence to prevent this. Indeed, the GNU licence is expressly designed to encourage distribution of all kinds, but the creators of easy-to-install Linux distributions have worked hard at writing installation routines and testing

them, and they feel they should be in charge of dishing them out.

Caldera OpenLinux Base is sold and supported in this country through a company called Avalan. It costs around £55 and the company's Ben Partridge can tell you where to get hold of it (see "PCW Contacts", p274).

Installing Caldera OpenLinux Base

Several readers have been asking me about installation on notebook computers. In the early days, the problem was getting the X-server to understand the LCD screen, but this has become much easier with last October's release of XFree86 3.2, which contains drivers for the main notebook video chips and includes an easy-to-use graphical setup. To help me gauge just how much easier, Seimens-Nixdorf kindly delivered one of its top-of-the-line notebooks, the Scenic Mobile 700, with a built in CD-ROM and stereo sound system.

It came with Windows 95 installed and my natural instinct was just to format the hard disk. However, I do know that many of you like to run machines that dual boot between the two operating systems and I get a lot of letters asking how to do this. Windows 95 was occupying the whole 1Gb of the Scenic's hard disk, so I needed to shrink this down to about half that size and create a second partition for Linux. Well, two new partitions, actually: one for the Linux root file system, and a second

partition dedicated to swapping. (If you are puzzled by this, look up "Linux partition requirements" in the Matt Welsh book, mentioned earlier.)

Firstly, I used Windows 95's own disk defragger to squeeze out the spare space and pack everything down into the lower 500Mb of the drive. To create the new partitions I could have used the fips utility, but this seemed a perfect opportunity to put Partition Manager 3.0 to the test. It runs from Windows 95 as an old DOS application, closing down the whole of Windows before loads and bringing Windows back again when it has finished. It shows a diagrammatic representation of the partitions on your drives and allows you to resize them visually by mouse dragging, and this gives you a much better picture of what's happening than having to calculate in megabytes or, as in the early days of fdisk, in cylinders.

In fact, I ended up with four partitions in all: one for Win95, a pair for Linux and Linux swap, and a fourth to house OS/2's Boot Manager... "Eh? How did OS/2 get in here?" you may wonder. It so happens that version 3.0 of Partition Magic comes with OS/2's Boot Manager as an optional way of handling multiple boots once you've got your partitions set up.

It's rather unfortunate that Boot Manager needs a whole partition to itself even though this is only as small as 1Mb, because the PC architecture only allows you a total of four primary partitions. But in this case it fitted in nicely. Installing the Boot Manager is a simple matter of clicking a pull-down menu from inside Partition Magic.

When I eventually quit Partition Magic I had just three partitions: in between Windows 95 and the Boot Manager was an unformatted wasteland onto which I was about to drop Caldera OpenLinux (establishing and formatting the Linux partitions is usually something you do during Linux installation).

Before you can do that, you first need to create one or more boot diskettes from the CD-ROM: at least, that is what I have always had to do so far; but not this time. Between them, Caldera and the Scenic Mobile had a neat trick up their sleeves — something I'd never seen or done before on an ordinary PC. The Caldera CD is set up to act as a boot disk on hardware that knows how to boot off a CD, and the Scenics BIOS setup has an option to do just that.

A chat with Caldera's Bryan Sparks

Bryan Sparks, CEO of Caldera, was in the UK earlier in the year setting up distribution and support, so I thought I'd corner him and get the Caldera story direct from the horse's mouth, so to speak. He has a nice, soft-spoken manner and what he says sounds OK; not at all like marketing-speak.

Caldera started inside Novell as something called "The Corsair Project". Bryan Sparks was one of 15 researchers whose brief it was to look for new ways to

develop quick time-to-market system software products. I wondered why he settled on Linux, rather than any of the other free UNIX offerings?

"We looked at NetBSD, FreeBSD and all the others," Sparks told me. "The truth is, FreeBSD is a good product and its networking has always been very, very strong. Linux's networking wasn't, at the time, but that was fixed when Linux 2.0 came out." (By the way, the Linux 2.0 kernel is the one used in our cover CD-ROM RedHat version.) Sparks calls picking Linux "a kind of gut move on our part". It was the originator of the kernel himself, Linus Torvalds, who finally decided the Novell team. "We flew him in to Utah to get to know him better because this was going to be a big risk for Novell. In fact, we got to know several of the key Linux people and we thought their personalities were perfect."

The idea for a commercial platform based on Linux was originally Sparks' own. He'd worked on several projects inside Novell, including the NetWare for UNIX that was called "Portable NetWare", and he was in on the early days of Novell's relationship with USL, the company that owned UNIX and which Novell eventually, and disastrously, bought. "It was a total mess," says Sparks. "I was working with USL for six months, and there were such irreconcilable philosophical differences that I just couldn't stand it any more." Tragically, Sparks could see the opportunities that UNIX represented, if only the politics hadn't been getting in the way. "I went back to Novell and said, 'Boy, you have the opportunity of a thousand lifetimes here'. Windows NT was totally unproven. UNIX System V Release 4 was decades ahead — well, I'm exaggerating, but it was a very good technology. But the personalities at USL just killed it."

However, the experience prepared Sparks for Linux. In fact, he liked Linux so much that he wanted to quit Novell and set up his own company around it with two other Novell employees. "But I owed a lot to Ray [Noorda], so before we quit we took the business plan to him." Sparks had put together a working prototype. "I had good friends at Visix giving me their user interface, and a bunch of other things I'd done myself. I showed it to Ray and he said 'Boy, we need to be doing that...'"

Ray Noorda OK'd the Corsair Project in early 1993, but in the following year, at the age of 70, he announced his retirement. "We ended up with Bob Frankenberg coming in," said Sparks. "I don't have any qualms about him: he listened to what we were doing (we were kind of a skunk-works, off-site). He was a very bright man and he said 'This is really good... but we're not doing it.' But Ray Noorda still believed in it and financed what became Caldera with his own money."



I felt I was cheating somewhat. This was just too simple: insert the CD, boot the machine, and up comes the skeleton version of Linux that Caldera uses to run its installation routine.

Anyway, I'll have more to tell you about Caldera on the Scenic Mobile, next month.

PCW Contacts

Chris Bidmead is a consultant and commentator on advanced technology. Email him at unix@pcw.vnu.co.uk
Caldera OpenLinux Base is available through Ben Partridge at Avalan on 01923 208449; email Ben.Partridge@PSC.co.uk



The common good

... and bad, unfortunately. But Chris Bidmead is in combative mood as he attempts to install the Common Desktop Environment on his network, and defends a remark he made earlier.

A couple of months ago I mentioned that CDE, the Common Desktop Environment, is now available for Linux, and I showed you a screenshot of it running on LinuxPro, the version of RedHat Linux supplied by WGS (Workgroup Solutions) whose web page (unsurprisingly) is www.wgs.com.

I'd borrowed the screenshot from the WGS web site. But this is supposed to be the Hands On section, so I made a vow to myself that I'd follow through by getting hold of the software and installing it on my network. There's some bad news about this, some good news, some more bad news, but ultimately, I'm glad to say, some good news.

Before we get into that, I'd like to deal with some heated email from a reader who objected to my remark in the column that the CDE screenshot made LinuxPro look "uncannily like the AIX desktop — which of course is the point of CDE". When the major UNIX manufacturers got together in 1993 around an initiative called COSE (the Common Open Software Environment), of which CDE was to be the first component, the intention was to simplify things for system administrators and users by offering a "look and feel" that would be similar across all the UNIX platforms. One of the first manufacturers to implement this was IBM on AIX, and as I happen to run AIX here, it seemed worth mentioning the resemblance.

Andrew Lehane seemed to think I was suggesting IBM had invented CDE singlehandedly. If my very brief mention in the screenshot caption gave that impression, I'm sorry. Long-term readers of this column may remember that back in January 1994, speaking of IBM's

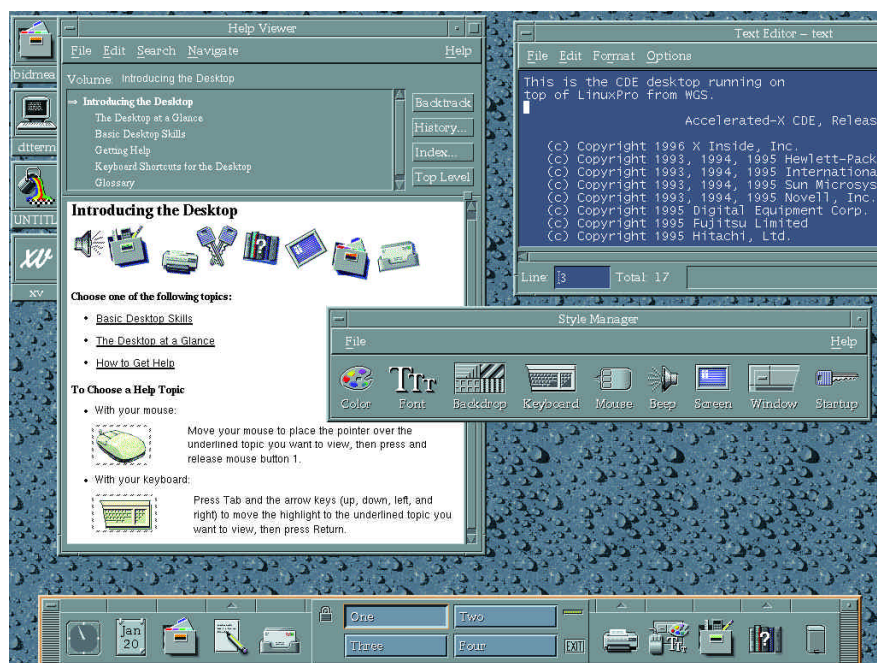


Fig 1 This is the Common Desktop Environment. It happens to be running on top of LinuxPro, but in theory it could be any UNIX and you wouldn't know the difference. It's not just a pretty face — there are internal subtleties like drag-and-drop which I'll investigate in later columns

introduction of the LaunchPad on OS/2, I said: "And the first of the promised COSE interface features also arrives with version 2.2 — a dashboard control based on Hewlett-Packard's VUE (Visual User Environment), a point-and-click program launcher and screen manager."

Yes, the panel at the bottom of Fig 1 derives heavily from VUE, but Andrew's complaint ("Why make these statements when, as I am sure you realise, CDE is in fact a decedent of HP's VUE Windowing System, an enhancement to OSF/Motif developed in post-Apollo days and released prior to 1991 for HP-UX 8.0?") seems to suggest that this is the whole story. In fact, all the COSE members

pooled their technologies to create CDE, as you can see from the copyright notice in the picture. As COSE was an initiative rather than an organisation in its own right — or as Sun's CEO, Scott McNealy, famously put it, "COSE is a verb and not a noun" — the product needed to be nurtured through the existing alliance of UNIX manufacturers, the Open Software Foundation (OSF) which has since been rolled into The Open Group. Andrew's loyalty to Hewlett-Packard (it turns out that he's working in its Telecom Systems Division) is commendable, and I'm delighted that he reads and likes the column, but I hope I've managed to deflect his accusation of "inaccuracy".

Good and bad

Now for that good news/bad news stuff. As soon as I discovered there was a Linux version of CDE, I dropped an email to Mark Bolzem, the Linux guru who runs WGS. The first lot of bad news was that he was very nervous about sending me review software because, as it turns out, it costs him an arm and a leg in royalty payments each time he ships the product. I find it ironic that someone whose core business is distributing royalty-free software should be held to ransom by the so-called Open Software Foundation, which charges him \$100 for the CDE and Motif components in each CDE shipment.

It's a tribute to Mark that he managed to get a package out to me, comprising the six-CD set that makes up LinuxPro, along with the CD and manuals for CDE. This was the good news. The implementation of CDE he uses comes from an outfit called Accelerated X, mainly known for its souped-up commercial version of the X server for UNIX on PCs. Its CDE requires the Accelerated X server, which is what Mark ships with his LinuxPro.

I installed LinuxPro and was all ready to go on to the CDE when the shattering news came through about Apple's acquisition of NeXT. I dropped everything and spent the next week or so on the internet, news gathering and chatting with gurus by email for some articles I was writing. After the dust had settled and it became clear that Gilbert Amelio may turn out to be the best thing that ever happened to Apple and NeXT (although it's a rocky road ahead) I finally caught my breath and went back to the relative tranquillity of LinuxPro and CDE. Or rather, just LinuxPro. Because — the next bit of bad news — I simply couldn't get my system to read the CDE CD.

I'd installed LinuxPro on the old Apricot Xen LS-II. You may remember the trouble I had with SCO OpenServer not being able to recognise the Xen's Sony CDU31a CD-ROM drive because it uses a proprietary non-SCSI, non-IDE interface. There's no problem with Linux though — a driver for the CDU31a is part of the standard distribution. One of the Linux system developers, Mark Evans, has even written a driver for the rather obscure on-board busmastering ethernet port on the Xen.

The Apricot Xen LS-II had read the LinuxPro installation CDs perfectly, and as far as I can remember had never shown any sign of trouble with other CDs before. But

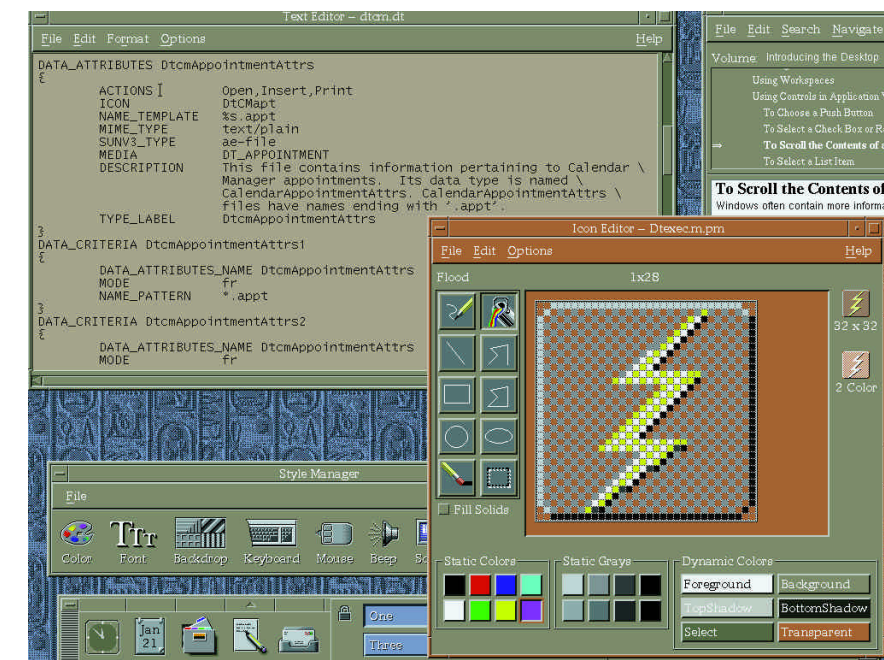
with this particular CD it was spinning its wheels, just at a time when I was desperate to install CDE and get a screenshot to you. Attempting to mount the CDE disk produced the error message (after a long hang) "can't read superblock". Even before I tried mounting the drive, its flickering LED indicated that it was labouring during the spin-up following inserting the CD. Oddly, the NeXT machine and the AIX PowerPC had no difficulty reading the CDE CD. It was just this combination of the Apricot drive and the CDE CD that was coming unstuck. After some experimentation — and I'm not going to tell you how many hours I spent messing around with this — I was forced to the conclusion that some drives, particularly older drives, just don't like some CDs.

This was distinctly bad news, and I found myself wondering how long it would be before I could free up another machine, reinstall LinuxPro on it, and then proceed with CDE. I was about to pack away the Apricot and sit down to write my excuses to you when I suddenly realised that this was the old, narrow, standalone-PC mentality that I'd supposedly escaped from three years ago when I started this column. The AIX machine, which had no trouble reading the CDE CD, is networked to the Apricot. Thanks to NFS, the Network File System that comes with every UNIX under the sun (or Sun, I should perhaps say), it's a trivial matter to hook up the AIX CD-ROM

drive straight into the Apricot and treat it as though it were a local drive. NFS is a client-server connection. Here, the AIX box is the server and needs to expressly "export" the drive in question. How you do this varies somewhat depending on the version of NFS. Under AIX, I used Smit, IBM's System Management Interface Tool, to mount the CDE CD in the drive and then export the drive — or rather, the directory I'd mounted it on — over the network. The directory doesn't actually go anywhere at this stage; it just gets added to a list of directories that the NFS daemon advertises as being available for network connection.

On the Apricot I then needed to mount the exported directory onto a local, empty directory. I created a /mnt/NFS directory and mounted the AIX CD-ROM there. If you read the manual pages for mount (you need to do this with the command "man 8 mount", 8 being the manual chapter that deals with system admin) you may come away reeling from the surfeit of options available. But the Linux mount command is pretty smart these days: if you just ask it to mount aixbox:/mnt/cdrom /mnt/NFS (mount this alien filesystem on this local directory), it should be able to figure out for itself that you're talking about an NFS mount and that it needs to be read-only because you're dealing with a CD-ROM.

Having done this, I had the full contents of my CDE CD available from the Apricot's



The CDE desktop reconfigured with a different background and colour scheme. Here you can see the icon editor together with a sample of the code CDE used to tie icons to executables and ascribe particular behaviours to them

/mnt/NFS drive. I switched to this directory, ran the dtinstall script I found there, and CDE loaded itself on top of my LinuxPro, with the handsome results you see in Fig 1.

Networking, modules and more

LinuxPro is based on the RedHat distribution, with some refinements added by WGS. Traditionally, Linux installation kernels come loaded with drivers for a huge number of devices and rely on autoprobing software to find out which ones to activate. This can leave you with an overlarge kernel, which is sometimes a tight fit where memory is limited. Once the installation is finished, you're normally recommended to recompile a slimmed-down version of the kernel tailored to your specific machine. For beginners, I should point out that this turns out to be a lot simpler than it sounds.

My Apricot Xen LS-II has 16Mb of RAM, which is plenty of room for Linux. Nevertheless, I prefer to start with the smallest possible kernel and add features as and when I need them. In the early days of Linux you couldn't do this, but newer Linuxes let you load "modules" which can be drivers for physical devices, file systems, or translation code like the iBCS module that lets you run applications written for other versions of UNIX. Mark Evans has modularised the driver for the Apricot Xen's on-board i82596 ethernet controller, so I installed a kernel that had no networking enabled, made sure this was working, and added the networking later. If you haven't yet installed Linux I should point out that most Linux distribution CDs come with a large variety of pre-compiled kernels in the form of disk images. You choose the appropriate image, transfer it to a floppy (where it appears as a bootable kernel and a bunch of vital support files) and use that as the initial boot disk. During the installation process you get the opportunity to install that same boot kernel onto your hard drive.

Typically, the installation process will mollycoddle you through procedures like setting up the network. The upside of this is that you get a working system with the minimum of effort. The downside is that you don't get to understand the fundamentals. Because I was starting without a network driver, the rest of the networking naturally refused to configure itself. Doing this manually was instructive and, luckily, not too arduous.

I found myself telling reader Michael Butler <m.butler@ic.ac.uk> about this at some length, and what follows below is a

Fig 2 Results of ifconfig

```
eth0      Link encap:10Mbps Ethernet  HWaddr 00:00:49:20:26:2F
          inet addr:192.168.1.25  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1176 errors:0 dropped:0 overruns:0
          TX packets:555 errors:0 dropped:0 overruns:0
```

Fig 3 Kernel routing table

Destination Iface	Gateway	Genmask	Flags	MSS	Window	Use
192.168.1.0	*	255.255.255.0	U	1436	0	569
eth0						
loopback	*	255.0.0.0	U	1936	0	48 lo

Fig 4 Resorting to ping

```
PING 192.168.1.3 (192.168.1.3): 56 data bytes
64 bytes from 192.168.1.3: icmp_seq=0 ttl=255 time=1.7 ms
64 bytes from 192.168.1.3: icmp_seq=1 ttl=255 time=1.4 ms
64 bytes from 192.168.1.3: icmp_seq=2 ttl=255 time=1.4 ms
64 bytes from 192.168.1.3: icmp_seq=3 ttl=255 time=1.3 ms
64 bytes from 192.168.1.3: icmp_seq=4 ttl=255 time=1.5 ms
64 bytes from 192.168.1.3: icmp_seq=5 ttl=255 time=1.5 ms
```

condensed version of that correspondence. Michael had written to me about the trouble he was having with his network card, and he'd opened his mailing with: "I have been converted to Linux through reading your column in PCW. It's a great column and worth the cost of the magazine alone!". This is a great way of getting my attention.

Once the install of the minimal kernel is complete, I login as root and then check that there's a set of modules, including the apricot.o module, somewhere among the library files. In my present version of Linux this turns up under /lib/modules/1.2.13/net.

Linux comes with a number of tools for handling modules, and you can get the full list with the command apropos modules. Among these is the insmod (insert module) command. This knows where to find the modules, and knows that modules are .o files, so the command line to add my network driver is just

```
insmod apricot
```

We've now installed the ethernet driver for the device known as eth0, but it's not yet doing anything very useful. You connect it to the network in two stages: first we need to give our eth0 a TCP/IP address, known as the "dotted quad" address. We use the ifconfig networking utility to set this up:

```
ifconfig eth0 192.168.1.25
```

With my own small network setup this

means I've given this particular machine the identifying number 25 on a network whose name is 192.168.1.0. (The full story is much more complicated, but let's not worry about that now.) The next step is to tell eth0 where to find the network. For this we need the route utility. Again there are all sorts of complexities you can get into with route, as you'll see if you consult man route, but for now let's just go with

```
route add 192.168.1.0 eth0
```

And that's basically it. If you now run ifconfig and route without command line parameters you'll get a report on the state of play. ifconfig gives something like Fig 2 (there'll also be an entry for lo, the loopback interface). The output from route should look like Fig 3. To check that the network really is connected I can use the old standby, ping, to send test packets to another machine and have them echoed back at me. For this I need the dotted quad address of the second machine

```
ping 192.168.1.3
```

which should give something like Fig 4, confirming the network is now working fine.

PCW Contact

Chris Bidmead is a consultant and commentator on advanced technology. Email him at unix@pcw.vnu.co.uk