



# Read between the lines

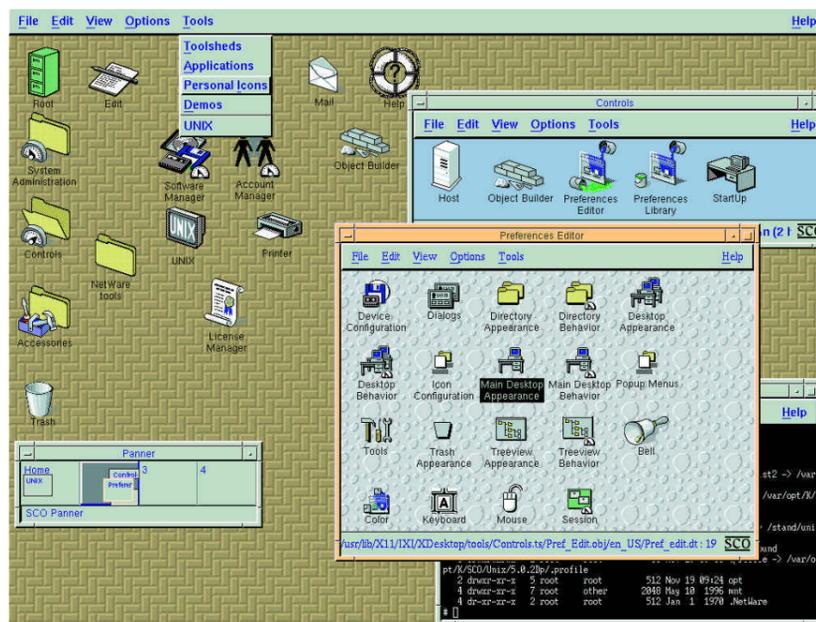
Chris Bidmead passes on a salutary lesson, gained by trying to install SCO OpenServer without taking enough notice of the documentation.

I had to delay telling you about the new, free, SCO OpenServer ([www.sco.com](http://www.sco.com)) because of installation problems. Yes, I know I promised a long time ago not to go on about installation problems in this column, but I do know a lot of you spend time struggling in this unfruitful area — as I do, too. I learnt some salutary lessons from (eventually) installing SCO, so I thought it would be useful to provide you with a quick rundown.

Initially, I completely failed to install SCO OpenServer on my trusty old Apricot Xen II 486, which has been a reliable home in the past for various different versions of OS/2 and Linux. SCO OpenServer is snotty about low-end hardware and in particular eschewed the Sony CDU31a proprietary interface CD-ROM drive. It thinks exclusively in terms of SCSI.

Apricot (or perhaps I should now call the company Mitsubishi Electric) stepped swiftly into the breach and sent me one of its LS500 range of desktop machines. This one is a 100MHz Pentium, equipped with a SoundBlaster 16 sub-system and a Cirrus Logic GD543x video chip on the motherboard. No SCSI, but the Triton chipset includes a dual ATAPI interface which drives the 1Gb hard disk and the CD-ROM. ATAPI, as I suppose everyone must know by now, is a recently standardised, simple, parallel interface that vies for performance with the lower end of the more traditional, more costly, SCSI connector.

What I like about the LS550 is its straight up-and-down compatibility. Buy a "bleeding edge" machine and it may turn out to be hard, or even impossible, to find drivers outside the world of Windows for the exotic Wide SCSI adaptor, or the gee-whizz 3-D



**Fig 1** SCO OpenServer doesn't have the classy interface of NeXTStep, but the neat icons give a clean, efficient look to the desktop, and they can be cleverly animated

video system. All this stuff on the LS500 is widely supported, and is going to work very nicely with any of the operating systems I'm likely to throw at it.

However, SCO Openserver still refused to come quietly. The problem centred around that requirement for a SCSI CD-ROM. It seems there's a way around this if you read the installation notes buried somewhere on the CD-ROM. (Tip: if you can read the CD-ROM of a new operating system on your old OS, and you usually can, then do so. Scour it for information before you launch into a new installation).

Dave Gurr, market development manager at SCO UK, tells me that SCO realised the growing importance of the

ATAPI interface only after the architecture of OpenServer had been committed to SCSI. SCO's engineers will have thoroughly fixed the problem in the new release which is due out about the time you are reading this. On the version I was trying to install, there is an interim solution; you have to use a kludged loadable driver called wd that talks ATAPI to your system, yet kids the OpenServer that it's dealing with SCSI.

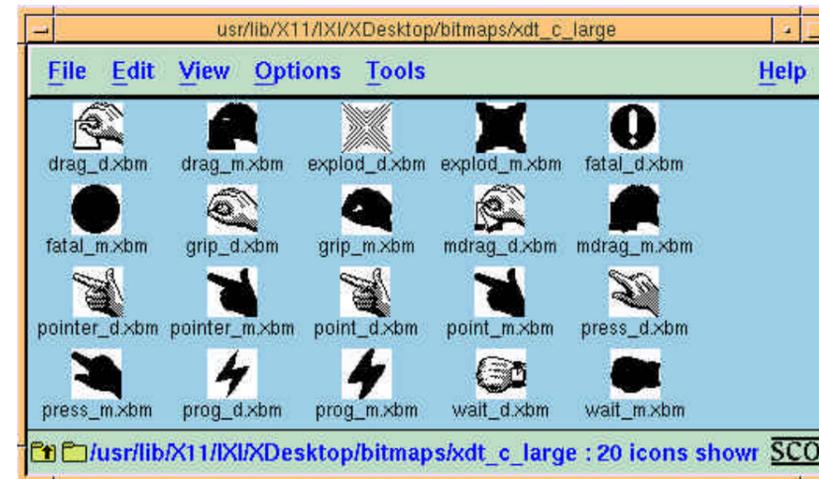
This works fine but confused me no end. The SCO boot disk detects the need for the wd driver, loads it and then invites you to fill in the various "SCSI parameters". You have to go along with the deceit by telling the poor deluded system what your Host Bus Adaptor number is, along with the SCSI

address and logical unit number. ATAPI interfaces don't have these numbers, so you have to follow SCO's arbitrary scheme for translating between ATAPI and SCSI parameters.

An ATAPI system that hangs an IDE hard disk and a CD-ROM off the same interface will run the hard disk as master and the CD-ROM as slave. So I thought I was putting in the right numbers: HBA = 0 for primary ATAPI interface, SCSI address = 1 for CD-ROM as ATAPI slave. But the installation kept coming back to me with: "Can't Find SCSI interface". Fair enough, I thought. I can't lie to you. There isn't one.

Following several attempts I re-read SCO's installation notes and the hardware spec for the LS550. The answer was staring me in the face, but you know what happens in these situations: you don't read it properly because you think you're covering ground you know already. If you ever find yourself getting as stuck as I did with this one, the only thing to do is to throw away everything you think you know and re-learn things from scratch.

I had picked up the idea from the SCO installation notes that the IDE drive and the CD-ROM were hanging off the ATAPI interface as master and slave. But when I had read the Apricot LS550 documentation carefully, I noted that the Triton chipset provides two ATAPI interfaces. The CD-ROM wasn't a slave, it was the master on its own secondary controller. The correct



**Fig 2** SCO: The pointing finger icon, which is the mouse cursor, pincers its finger and thumb to pick up objects, and rotates its wristwatch into view while waiting for a job to finish. Here are the icons that SCO's X.Desktop uses to do this

parameters for the wd driver were HBA = 1, SCSI address = 0. After that, the installation found the CD-ROM drive sweet as a nut, and the rest was plain sailing: a totally automated, go-away-and-have-supper, one-hour-and-a-bit data transfer from the CD-ROM to the hard drive.

## Sterling work, Carruthers

One of the minor joys of Linux-FT is that when you install it to load the UK keypad and hit the pound sterling sign on top of the number 3 key, you actually get the £ sign. Try this with RedHat or any of the other US Linux distributions and the chances are you'll hear a beep and nothing will turn up on the screen. This is the console screen I am talking about — so let us leave X out of it for now...

One of my early excursions into the bowels of Linux a couple of years ago was an attempt to track down this pesky little problem. It turns out there's no need to do what I did, and start unpicking the source code. But on the other hand, reading chunks of the source during this venture was a valuable part of my Linux education. If you haven't explored the source on your system, why not cd to /usr/src/linux and have a mooch around. This is the one big advantage that the Linux folks have over devotees of, say, SCO. Even though SCO OpenServer is now "free", you won't find any source knocking about.

You'll be asked during the installation of most Linuxes which keyboard you want to use, and thereafter the appropriate keymap will be loaded automatically each time you boot. You can see how this works in

Linuxes such as RedHat that use the System V initialisation scheme (most do these days) by finding a directory called rc.d under /etc. This is where all the initialisation scripts are run, like a kind of glorified archipelago of AUTOEXEC.BATs, every time the machine comes up or whenever you shift from one run level to another. On my RedHat system there's a sub-directory under rc.d called init.d which contains a script called keytable. Find your own keytable script and the related scripts to see how the appropriate keymap gets loaded.

You might think that loading the correct keymap would be enough to set you up with the £ sterling sign. But alas, not so. The keyboard mapping is only part of the story. You will also have to make sure that an appropriate screen font is loaded. This is achieved by a utility called, reasonably enough, Setfont. Setfont should be loading an ISO Latin font such as lat1-16.psf. You can do this manually while testing the system by running Setfont from the command line. See the man pages for Setfont for full details.

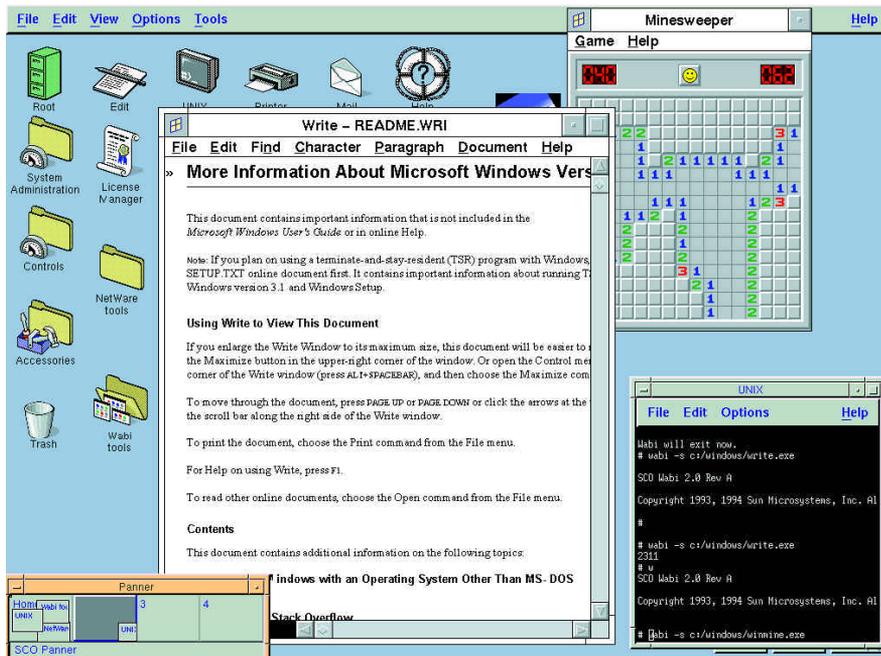
So you've set your screen font and your keyboard. What happens if, as I found, your keyboard still fails to deliver a £ sign? Well, frankly, I gave up. I got used to writing "100 UKP". It didn't help that my favourite machine came with a US keyboard that doesn't even have a £ sign, and I've never bothered to change it. I completely forgot about the problem until I ran into a Unix veteran with the splendidly colonial name of Shaw Carruthers in one of the Linux electronic discussion groups running on the CIX conferencing system.

## In a lather with Java

I came across a nice little anecdote in the comp.os.sys.be newsgroup, which I reproduce here with the permission of its author, Kurt Glaesemann who is at [kurtg@iastate.edu](mailto:kurtg@iastate.edu). The discussion was about some of the more unreasonable things people expect of computers and Kurt mentioned a web site he'd visited where nothing seemed to work. He inspected the source code for the page, where he found a section that went something like this:

```
if (JavaScriptExists) then
  ... cool stuff ...
else WriteJavaMessage("sorry
this page needs JavaScript")
end if
```

I leave it as an exercise to the reader to work out why this might cause problems with a browser that doesn't support JavaScript. (Hint for beginners: the script above is written in JavaScript).



**Fig 3** SCO OpenServer comes with Wabi 2.0 (as a cost extra) which I'm using here, somewhat unadventurously, to run Microsoft Write and Minesweeper. You can open Wabi as a Windows environment, or you can run individual Windows apps in their own windows, rather like "seamless" WinOS2 under IBM's Warp

Shaw told me that the two other elements you may have to fix are the ability of the application you're running to handle ASCII characters with values over 128, and the behaviour of the console you're using. Don't forget that even if you're only on the command line, you're still running an "application" — the shell.

With current versions of Linux, this shell is likely to be bash (Bourne Again Shell) by default. (You can usually check this by typing "echo \$SHELL" at the command line and seeing whether it reports "/bin/bash". If you don't get any output you probably forgot to put \$SHELL in capitals). When bash loads it consults (among other settings) files such as .profile a personal keybinding file called .inputrc, which sits in your home directory. Take a look at man bash for the full details. Shaw suggests that you include the following lines in your .inputrc file to make sure that bash can cope with European characters:

```
set meta-flag On
set convert-meta Off
set output-meta On
```

Now let's look at the console behaviour. What you know to be a character-based, memory-mapped screen is, as far as UNIX in this mode is concerned, a dumb terminal sitting at the end of a serial connection. If you enter the command "stty —all" (or "stty all" on older systems), UNIX will tell you all it

knows about the terminal, including its alleged "baud rate". You can also use stty to change the features of the virtual terminal, including its handling of ASCII chars above a value of 127. Shaw suggests that you add the following lines to your .profile script to make sure the console is 8-bit clean:

```
tty -s
if [ $? = 0 ]; then
stty cs8 -istrip -parenb >&0
fi
```

The tty command silently (-s) checks to see if it really is dealing with a terminal and returns 0, tested in the following line, if this is the case. Stty then sets the character size to eight bits (cs8), negates any tendency to clear the high bit on input characters (-istrip) and tells the terminal not to look for parity bits (-parenb). You'll notice the minus sign isn't used here to flag a parameter, as it is in many UNIX commands (like the tty command in the first line), but to toggle a parameter on or off — just one of the delightful inconsistencies which gives UNIX its charm.

#### Preparation for ISDN

Following our discussion about ppp and dial-up to Internet Service Providers last month, I've taken a bold step forward and installed a base rate ISDN line.

In theory this gives me higher

## Linux gets Wabi

As I write, Caldera has just announced that it is shipping version 2.2 of the Linux port of Sun's Wabi, the Windows environment for UNIX. If you really must run Microsoft software such as Excel and Word on your UNIX machine, this is certainly one way of doing it.

I'm hoping to take a closer look at Caldera's offering in the near future. Meanwhile, I have the SCO OpenServer version running here (Fig 3), and it certainly does the job, provided you define the job as only running a limited subset of 16-bit Windows applications.

What worries me about all this is the price. Version 2 of Wabi requires 16-Bit Windows, which (on my SCO system at least) you have to install yourself. So add the cost of a Windows licence, to the Caldera price of nearly \$200, and you've got yourself an expensive way of running old Windows applications.

bandwidth to ISPs (provided they support ISDN, of course) although the speed of the link in practice is dependent on factors like the load on the ISP and the remote servers you are visiting.

There's an immediately discernable benefit in the speed with which you establish the connection, because ISDN dispenses with the dialling tone and the tuneful duet of line speed and communication standards negotiation that can take up to 30 seconds with a ordinary analogue phone line.

As well as the ISDN line you need a terminal adaptor (TA) through which you make the computer connection. It helps if this behaves as closely as possible to a modem, so that your software doesn't have to be completely rethought. Because base rate ISDN is effectively a pair of lines, it's also useful if your TA has an analogue input that you can treat like an ordinary phone line, either for a modem connected to a second computer, or for use as a voice line. P&L Systems, which is run by Paul Lynch whom regular readers will remember as my NeXT guru, has lent me one of its ISDN TAs. It's the ZyXel Elite 2864i "ISDN modem" and it has all these features, and more. Next month I'll tell you more about how I got this going with my UNIX network here.

#### •PCW Contacts

Chris Bidmead is a consultant and commentator on advanced technology. He can be contacted at [unix@pcw.vnu.co.uk](mailto:unix@pcw.vnu.co.uk)