



## Fully booked

Where do you think Chris Bidmead gets all the Linux information he imparts each month? From a book, of course. Here he reveals some of his sources, and revels in more Partition Magic.

### Partition Magic revisited

Last month I mentioned the miraculous partition moving and resizing software for OS/2 that One Stop Software of Bristol (tel 0117 985 3370) sent me for review. I said that Partition Magic seemed like a great idea, but I was chary of testing it out. Moving partitions around can be disastrous, particularly as the manual warns of "bugs in OS/2's Boot Manager" which mean you may not be able to find the moved partition when you come to reboot.

The authors of the manual are right to be cautious, but the Boot Manager problem turns out to be easily fixable. All you have to remember is to run FDISK again, and get it to re-save its information by changing the name tag on the partition you've just doctored.

Partition Magic seemed to be just what I needed when the time came to install Caldera. My Intel-built 486 box is fitted with a very handy 1Gb IDE drive from Micropolis that gets around the usual large-drive problems by fooling the BIOS into thinking it's a pair of drives. On Drive 0 I have a DOS partition, an HPFS partition, a partition that runs Windows NT, and the Boot manager. Drive 1 is divided into a pair of partitions, Linux FT root and Linux FT swap. So where do I put Caldera?

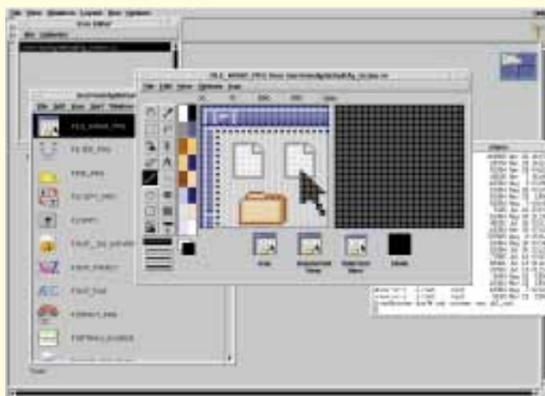
The Linux swap partition doesn't really need to take up nearly half a pseudo-drive, and OS/2 is only on this machine to manage the Boot Manager. So I thought I'd use Partition Magic to shrink the OS/2 partition from around 160Mb to 60Mb, put a new Linux swap partition in the space recovered (which the Boot Manager needn't know about), thus freeing up the

previous Linux swap partition on Drive 1 for Caldera. Caldera and Linux-FT could then both share the same swap partition on Drive 0.

Partition Magic was the key to getting all this moving, but a catch was immediately apparent. You can't use Partition Magic to resize or move partitions that it or its host operating system is running on. Happily, the authors have foreseen exactly the sort of dilemma I was in, and bundle in a version that will run under DOS. DOS can't see partitions beyond the 1024 cylinder limit, but that problem doesn't arise with the cunning Micropolis drive. So I copied Partition Magic to a network drive, rebooted under DOS, reconnected to the network and ran the program. It shrank my OS/2 partition in a couple of minutes. Fast, but not hasty, it proceeds cautiously through each stage, with a visual progress indicator showing it checking current integrity, clearing the truncated partition, adjusting the structures, and doing a post-integrity check.

This puts a new, empty partition between OS/2 and Windows NT. Linux sees hard-disk partitions as raw devices

*A taste of Caldera. The icons are gorgeous and there's a very good icon editor to modify them or create your own*



### Reading matter

Last month I gave a brief plug to Mike Gancarz's book *The Unix Philosophy*, which I'd say is essential reading for anybody interested in the whys and wherefores of operating systems. It got me thinking that maybe I should tell you about the other books I'm relying on (you didn't think I just know this stuff, did you?). If you're just getting started with Linux you'll find a ton of documentation on the Net (Websurfers might start at

<http://sunsite.unc.edu/pub/Linux/Welcome.html>

but see box on Sunsite mirrors, page 268). But the problem for beginners is that there's just so much of it.

That's why I'm so keen on Mike's book, which you can get from Digital Press. All the other documentation tells you how — this book tells you why. Put it beside the O'Reilly *Unix in a NutShell*, a snappy catalogue of what you can do and how, and you have the makings of a complete general Unix library.

If you're running Linux you might want to make room for, well, *Running Linux*, again from O'Reilly. It's very thorough on subjects ranging from "What's so good about Linux anyway?" to "How do I configure my TCP/IP addresses?", and is written in an easy-going style that covers the technicalities without getting bogged down.

There's a complementary book from O'Reilly, *Linux: Network Administrator's Guide*, if you want to delve deeper into TCP/IP and set up email and news systems, and this includes a very good technical and historical background on networking in general. This book is also complimentary, with an "i", in the sense that it's part of the Linux Documentation Project and therefore downloadable from the Internet if you don't feel like lashing out for the O'Reilly hard copy.

Cashing in on the Linux boom comes another of those silverised, airport-novel style tomes from Que called *Special Edition: Using Linux*. This has a Slackware 2.0.1 CD-ROM

*Caldera's Looking Glass desktop is full of neat tricks. Here's a command-line widget as an alternative to the plain xterm terminal*

labelled

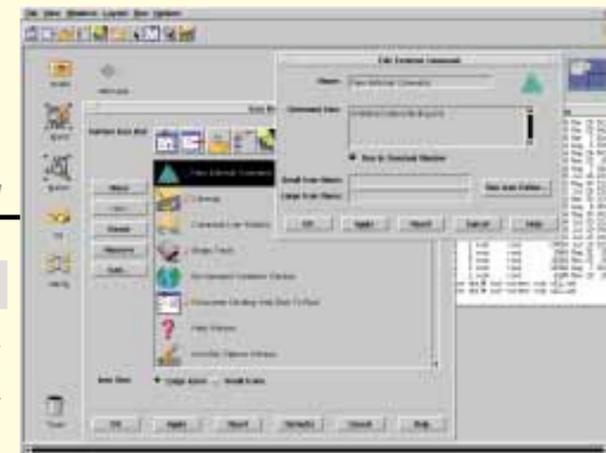
`/dev/hdax` or

`/dev/hdbx`,

where a or b indicate the first or second drive, and the x is the number of the partition on that drive. According to my calculations the new partition would be `/dev/hda3` to Linux, all ready and waiting to be turned into a swap device.

You have to be very careful when messing about with partitions from a number of different operating systems — it's all too easy to format the wrong one. Because of this I've got into the habit of creating partitions whose sizes differ by 25 or so megabytes, so their sizes give a clue to which partition you're dealing with. You need around 25Mb as the granularity because different operating systems measure size in different ways, and what looks to OS/2 like a 100Mb partition can look like 98Mb to Unix.

But when I powered up Linux and ran `fdisk`, the partition wasn't there! The newer disk utility, `cdisk`, revealed what had happened. It's an iron rule of PC disk



management that you can only have four partitions on a disk. You can cheat around this: one of the partitions can be an "extended" partition under DOS, OS/2 and Windows NT, and then subdivided into "logical" partitions which become lettered drives. If you have only one drive on the system, these subdivisions will be D:, E:, F: and so on, complementing whichever of the primary partitions is currently drive C:.(There's a complication here: primary partitions on a second drive get allocated drive letters preferentially, but we won't go into that right now.)

Logical partitions are really intended to store data, although some operating systems like OS/2 can be run from logical drives. The copy of OS/2 that I had so carefully shrunk with Partition Magic was on a primary partition; but the next partition

stuck inside the back cover, on the strength of which the publishers blurb it as "Your one-stop resource to this increasingly popular operating system." Like *Running Linux* it discusses what you might want to do with the software, as well as how to do it, so there are chapters on getting onto the Internet, and even on how to behave when you get there. It's about the size of the two O'Reilly *Running* and *Net Admin* books combined, and covers the whole ground.

If I had to pick just one book out of these three I think it would have to be *Running*. The *Special Edition* book with its bundled CD-ROM comes a close second, but, I'm sorry, I have a marked aversion for writers who, like the author of the Que book, call the X Window System "X Windows" passim.

While we're talking about Linux documentation, another "must have" if you're serious about tracking this fast-developing operating system is the *Linux Journal*. It's a slim monthly (a year's stack of them takes up the shelf

space of one PCW!) but it's packed with Linux-specific tips, news and reviews, and I think I'd be lost without it.

### Where to get them

*Linux Journal* is published in Seattle. The subscription-meister Amy Simpson (subs@ssc.com) charges \$19 to US residents for an annual subscription and adds a not unreasonable \$10 to wing the thing beyond the North American continent. Alternatively, if you bought Linux on a CD, check with your dealer.

You can shop for the O'Reilly books on the Internet at

<http://www.ora.com>

or get them, and the other books, through the UK bookseller, Computer Manuals. Their phone number is 0102 706 6000 and they have a Web presence at

<http://www.demon.co.uk/compman>

and an online bookstore at

<http://www.easynet.co.uk/compman.htm>

along, set aside for Windows NT, was extended. The space I had created became a new logical partition in the extended partition.

The whole extended partition gets a Unix raw device name, but the subdivisions aren't recognised. The Linux fdisk just shows the raw device as an extended partition. Though the eyes of cfdisk Linux can see the subdivisions, there's no way it can make use of the new space as a swap partition.

### Swapping with Linux

So, I temporarily abandoned the idea of a swap partition until I could get around to a total backup and rebuild of the hard disk. But now, neither of my Linux partitions had any swap space, and some of the X stuff was refusing to run. Happily, there are two ways of organising swapping under Linux — swap partitions or swap files. Swap partitions are preferred because they're out of the way and run no risk of being fragmented. But swap files can do a perfectly good job — that's the only way OS/2 and Windows NT can do it.

My Linux FT partition (/dev/hdb1) was larger than the new Caldera partition, so I decided that it should host the swap file. First I needed to be able to see the Linux FT file system from my Caldera partition (which would be convenient anyway for sharing data files and applications). Under Unix, you do this by creating a mount point (an empty directory) and connecting it to the raw device on which the other file system is sitting. So from within Caldera I ran the following commands:

```
# mkdir /linux_ft
# mount -t ext2 -o defaults /dev/hdb1
/linux_ft
```

The -t parameter defines the filesystem type I'm connecting to. The -o defaults option is shorthand for making it read/write and setting some other standard parameters. With these two commands, I've just connected the whole of my Linux FT file system seamlessly into my Caldera file system. Now to create the swap file.

First I need to create a file of the right

### Sunsite mirrors

Sunsite is a very busy US server and they suggest that if you're calling from Europe, you use one of the many local mirrors. UK readers have a choice of at least four:

- **Coventry**  
ftp.maths.warwick.ac.uk:/mirrors/linux/sunsite.unc-mirror/
- **Greenwich**  
ftp.idiscover.co.uk:/pub/Linux/sunsite.unc-mirror/
- **London**  
src.doc.ic.ac.uk:/packages/linux/sunsite.unc-mirror/
- **Mildenhall**  
ftp.dungeon.com:/pub/linux/sunsite-mirror/

size. There are several ways of doing this, but the standard procedure is to use the dd command. This modestly-named utility is a real powerhouse low-level file converter, but all we're going to use it for here is to fetch a bunch of zeros out of the zero bucket and pour them straight into the new file. The Unix bucket you fetch zeros from is the pseudo raw device /dev/zero (there's a corresponding bucket to pour away junk you don't want, and that's called /dev/null). Here's what the completed command line looks like:

```
# dd if /dev/zero of /linux_ft/swap
bs=1024 count = 16284
```

This is the kind of Unix command line that's famous for frightening away tourists, but it's perfectly readable when you understand that if stands for "input file", of is "output file", bs is block size and count is the number of blocks. (Anyone a little more clued up than a tourist might wonder why the parameters are bereft of their usual introductory hyphens (-if). The answer is that they just are.)

Now we have the 16Mb file and we're going to format it to be a swap file. As this is a low-level operation, and some of the 16Mb of zeros may still be floating around loose in the the disk buffers, we run:

```
# synch
```

to make sure it's all written physically to disk. Now the formatting. We only need to

do this once, with the command

```
# mkswap -c /linux_ft/swap 16394
```

The -c parameter (there's that hyphen) tells mkswap to check for bad blocks. Why you need to tell mkswap that the file is 16394 blocks in size I'm not sure, and what happens if by mistake you make this number too big I dread to think and am not inclined to test. This is low-level stuff — tread carefully.

Another sync to flush the buffers, and we're ready to tell the system to go ahead and use the new file for swapping. The command is just

```
swapon /linux_ft/swap
```

You can now check that the swap is working by running the free command, which reports how much real and swap memory you have available. Everything's ready to run, and you can test the swapping by loading a lot of apps and seeing them all run properly. There's a nice little piece of X screen furniture called xsysinfo that lets me watch the swap file filling up as I work, and it quickly becomes clear that I'm probably going to need more than 16Mb of swap space.

Swapping to a file turns out to be rather slower than I'd like, so I'm going to have to find some way soon of creating that dedicated swap partition. That will be a process similar to the one we've just gone through, skipping the dd part and substituting the raw device name of the partition for the filename. And I don't have to have either a swap file or a swap partition — the design of Linux lets me mix and match a number of different swap devices.

We're not quite done yet with this temporary swap arrangement. Next time I power up Caldera I don't want to have to type the swapon command explicitly. Actually, a swapon command kicks in automatically at power-up time: by default it's already there in one of the session initialisation files. In the form "swapon -a" it will begin swapping on every swap device it knows about. It's very like the mount command, and like mount it "knows about" devices because it goes to look them up in a system file called fstab (file system table) which on Linux systems is found in the /etc directory.

Next month I'll talk about how to fix up /etc/fstab, and we'll have a proper crack at Caldera.

### Raw device

"Raw device" is one of those obscure terms that Unix people take for granted. One of the key design goals of Unix is to make everything look and behave like a file, so a chunk of hard disk or a serial port is given a file-like name and stored in what looks like an ordinary directory (/dev/hda1 or /dev/tty0).

The code necessary to join the "filename" to the function of the device is hardwired into the

kernel. (Newer Unixes are using dynamically loadable device drivers, and Linux is moving in this direction.) Unix comes with the /dev directory stuffed with device names. They are not arbitrary, and you can't just stick your own names in there, although you can create a new name like /dev/mouse and link it to an existing name like /dev/aux1.

### PCW Contacts

**Chris Bidmead** is a consultant and commentator on advanced technology. He can be contacted at [bidmead@cix.compulink.co.uk](mailto:bidmead@cix.compulink.co.uk)