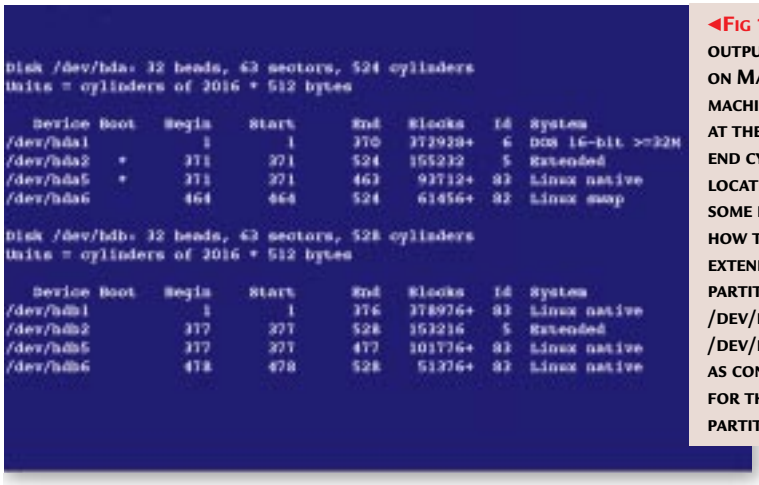
If the cap fips

Chris Bidmead covers partitioning with fips for more memory, and completes the installation.

Last month, my mate Marcus and I were installing Redhat Linux 5.1 onto the extra half-gigabyte drive he'd added to his system for the purpose. But a disk-space shortage stopped us dead, so our best option seemed to be to shrink the existing Windows partition. Last month, in the *Hands On Unix* column, we looked at Partition Magic, but there's a free, traditional-software way to resize DOS partitions using the fips utility created by Arno Schaefer <schaefer@rbg.informatik.th-darmstadt.de> which is bundled with most Linux distributions. We found it on the RedHat 5.1 CD-ROM under the /dosutils/ fips15c directory. (The older fips.exe directly under /dosutils is probably best avoided.) Beneath this is another directory, fipsdocs, which you should peruse carefully before proceeding. Fips has been in use for several years now and is probably as stable and reliable as any software working at this low level can be. But any utility that changes your partition table is potentially dangerous.



◀FIG 1 THE OUTPUT OF FDISK ON MARCUS'S MACHINE. LOOK AT THE START AND END CYLINDER LOCATION TO GET SOME IDEA OF HOW THE EXTENDED PARTITIONS, /DEV/HDA2 AND /DEV/HDB2, ACT AS CONTAINERS FOR THE LOGICAL PARTITIONS

[FIG 2]

```
/dev/hda1 (Windows) 364Mb /dev/hda5 ( /) 91Mb /devhda6 ( swap ) 60Mb

/dev/hdb1 ( /usr ) 370Mb /dev/hdb5 (/var ) 99Mb /dev/hdb6 ( /home) 50Mb
```

Long division

1 Fips reduces the size of your DOS partition by changing some values in the partition table and boot sector. Before using it, you need to run a defragger, like the Disk Defragmenter which comes with Windows, to move all the spare space to the end of the partition. It's worth doing the defrag whether or not the Microsoft utility recommends it. The defrag exercise left us with 316Mb free space.

2 Fips runs under DOS, so we created a DOS system floppy with the fips files on it, following the instructions in the fips.doc file, and rebooted the machine. In a single-drive machine, fips pauses to let you read the warnings about not running it under a multitasking operating system, and then jumps into showing you your partition table. With a two-drive machine like ours, fips pauses again before this to ask which drive we want to operate on. We chose the first drive and the appropriate partition table appeared — a useful check that we were looking at the right drive.

3 As fips is about to change your Master Boot Record (MBR), it first asks if you want to make a backup of this part of the hard disk. We said yes, and fips confirmed "created file a: rootboot.000".

4 Fips then suggests a size for the new partition. To leave some slack for Windows, we adjusted this downwards. The adjustment is made with the cursor keys, the screen showing you the increasing size of the Windows partition and then changing the cylinder position to start the new partition, as well as the size the new partition will be.

That's virtually all there is to it. You receive the usual warnings before committing the new partition table to disk, but even then you can back out because you've saved the old partition table, and the restorrb.exe utility is there to put it back for you.

Once that's done, you're ready to exit from fips, remove the floppy and start your Linux installation — or in our case, reinstallation. But first we rebooted to confirm that Windows was working correctly. A check with Windows fdisk [Fig 1] indicated that the partitioning had been carried out exactly as fips said, giving us an additional 150Mb which we decided to use for swap and /, the root partition.

A cut above the rest

The new partition fips had created was marked as DOS 16-bit. When the reinstall brought us back to Disk Druid, we deleted and used the space to create two new partitions. We made /dev/hda5 60Mb and allocated it to /. This left 40Mb as /dev/hda6, which we reserved for swap. (In case you're wondering about the numbering, these are logical partitions which, under Linux, start with 5.) Using the Disk Druid "growable" partition property I mentioned last month, we ended up looking like Fig 2. We also used Disk Druid at this point to mount the existing Windows partition, /dev/hda1, on /mnt/win95. You can call this mount point (/mnt/win95) anything you like.

We followed the convention of putting the mount point under the /mnt directory and giving it a self-explanatory name.

Smooth installation

After this, the installation proceeded as before. This time we didn't run out of space and we proceeded smoothly to the installation of the X Window System. The dialogue box asks you for the name of your video card and installs the relevant X server from the CD. The "X server" is the application which makes itself responsible for handling all your graphic primitives; it's nothing to do with a hardware server.

Monitoring the issue

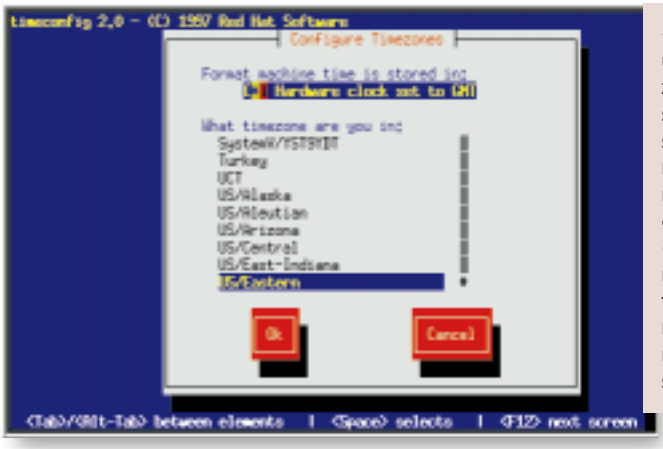
The next point, your monitor type, can be confusing if your particular monitor is not on the list. The X server takes its configuration for your system from a text file called XF86Config, in the /etc or /etc/X11 directory. The installation tried to write this file for us. It essentially needs to know the horizontal and vertical frequency ranges within which it can safely work without frying the monitor.

Monitors are extensively badged, so the name of yours may not be on any list known to Linux. We were using a badged Apricot 15in and Marcus suggested we tell the installation it was an NEC Multisync 3D. The Multisync model is six years old so the frequency ranges written to XF86Config are guaranteed to be fairly conservative.

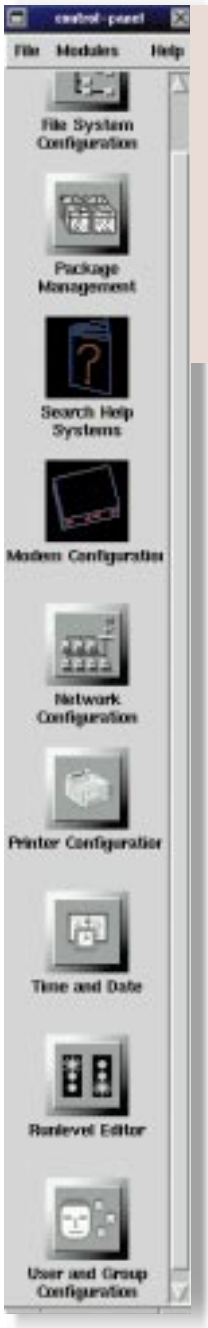
Once you have a working X configuration and know a bit more about it, you can always edit the XF86Config file directly or re-run either of the X configuration programs, Xconfigurator or XF86Config.

Time will tell

- We skipped the next step (installing the network) as Marcus wanted a standalone machine to start with, and moved on to setting the clock, or, more correctly, the relationship between the hardware clock and the Linux system [Fig 3].
- Then a few more questions: what services should be automatically started? This is a complicated subject, but for now you can just accept the defaults — the installation is intelligent about this and you can always change it afterwards [Fig 4]. If you want to query individual services, move the highlight to that service and hit F1.
- Configure printer? Impatient to get finished, we skipped this for now and moved swiftly on to the next screen, which invites you to set the root password [Fig 5]. Only a couple more steps to go now. The installation asks if you want to create a custom boot disk. A disk like this is handy for rescuing a fouled-up system, so yes, you want to create this disk, believe me!
- However, before this actually happens, you're moved on to the next question: "Where do you want to install the bootloader?" What you're setting up now is the actual boot sequence for the completed installation.



◄Fig 3 Linux understands time zones and summertime shifts. Tell the installation you have a hardware clock set to GMT (most PCs let you do this through the BIOS) and pick a time zone from the list supplied



◄Fig 4 Once your X interface is up and running, you can use the control panel to configure elements of the system you skipped during installation — networking, for example

▲Fig 5 Setting the root password. Whatever you do, don't mess this up unless you enjoy reinstalling!

Hello, lilo

OK, take a deep breath. This is where a lot of installations go wrong. As I mentioned last week, the bootloader is lilo, a low-level, operating-system-independent utility that uses primitive BIOS calls to install the Linux kernel and kick the system into life. Lilo also knows how to boot Windows and other operating systems.

To boot Windows, it needs to know where the Windows partition is (in our case, on /dev/hda1). To boot Linux, it needs to know the device name of the root partition (/dev/hda5) and the location of the kernel. Unless these factors are right, your operating systems won't boot.

In the past, I've always recommended

putting lilo on the boot sector of the Linux root partition, not the MBR. This is because Windows 95, if installed after another operating system, will helpfully "repair" the MBR to DOS standard, damaging lilo if it's there, with baffling consequences for the beginner. However, Red Hat's Disk Druid poses another problem with its penchant for creating logical partitions. The DOS MBR is a solid old standard, but it can't do smart things like jump to a logical partition. So, if you tell the installation that you'd like to put lilo on the boot sector of the Linux root partition (in our case /dev/hda5), the DOS MBR will never find it. But you won't discover that until you reboot!

I've argued with Red Hat's techie, Donnie Barnes, about this but I can't get him to see it as a problem. He thinks people should just *know* that the DOS

▲ **Fig 6 Where to put lilo? If you've used Disk Druid, the MBR is probably your safest bet**

MBR can't cut it, and use another boot loader or put lilo on the MBR [Fig 6], being careful not to let another operating system mess it up.

Anyway, the bottom line is that it's probably simplest to tell the installation to put lilo on the MBR. The installation also allows you to include other operating systems you'd like to boot, so this was the point to mention, that we had Windows 95 on /dev/hda1.

What the installation is actually doing here is creating a configuration file called /etc/lilo.conf that tells lilo how to behave when it's triggered by the boot sequence. When you know Linux and lilo better (you'll find plenty of documentation on lilo under /usr/doc) you'll be editing this file directly. The very, very important thing to remember is that lilo doesn't use this text information directly. It first needs to translate the info into a form it can understand at the primeval, pre-file system, pre-operating system stage of the boot process. To do this, you must run lilo at the operating-system level every time you change lilo.conf. Thankfully the installation process understands this, and at this point runs lilo for you automatically.

And that's it... we're done. The installation creates the tailored boot disk, and then you can remove the floppy and the CD and reboot your system. By default, the reboot takes us into Linux because that's how we set it up, but if we want Windows we just have to type "Win95" at the lilo boot prompt.

Installing Linux on a standard production PC

As Marcus drove off home with his dual-boot Linux/Windows 95 home-built machine, I moved on to my next project. The intention was to show how easily Linux can be installed on a standard current production model from a well-known manufacturer – the kind of machine I know many of you out there are using. Dell was happy to help out with a loan machine, and, by way of a change, I thought this time I'd use the S.u.S.E 5.3 Linux distribution.

Steve Jackson, Dell UK's business manager for Servers, wanted to be absolutely sure his company's product wouldn't let me down, so he picked out a PowerEdge 2300 with a 350MHz Pentium II and 6Gb of hard disk; not quite the "standard model" for which I was hoping, but Steve was keen to supply a machine that had been "certified for Unix". Dell machines of all sorts are happily running Linux all over the world, but Steve was taking no chances.

If you knew S.u.S.E.

S.u.S.E. is a German-based company whose very solid Linux distribution is available on the same terms as Red Hat: you can get it free down the wire; very cheaply on CD; or reasonably cheaply, in a box with a manual. The boxed set comprises five CDs in a jewel case, a boot floppy, and an installation manual of almost 400 pages. Despite its somewhat quaint English, this manual makes a useful reference book for all flavours of Linux. With the official version of S.u.S.E you also get 60 days installation support. As with Red Hat, the UK supplier I'd recommend is John Winters, who runs the Linux Emporium at www.polo.demon.co.uk/emporium.html.

"There are two ways of installing S.u.S.E Linux," says John, on his web site, "the one-page, 30-minute method [I tried it and it worked] or the rest-of-the-manual method which will give you as much detail as you could want on the intricacies of Linux installation. The actual installation process is handled by S.u.S.E.'s YaST [Yet Another Setup Tool] program which allows you to do both the original installation and later configuration."

Heaven and Dell

If Dell's Steve Jackson hadn't insisted on the PowerEdge, I don't suppose there'd be much to write about at this stage. The ironic twist is that Steve managed to pick the one machine in Dell's range that, as it turned out, won't run current Linux versions right out of the box. Next month, I'll explain the problem and the solution, and we'll move on to installing networking.

PCW CONTACTS

Chris Bidmead *can be contacted via the PCW editorial office (address, p10) or email* unix@pcw.co.uk.