Loading up Linux

Chris Bidmead hears the call of all those readers longing to use the Linux operating system and, in the first of this three-part series, shows you how to make a start.

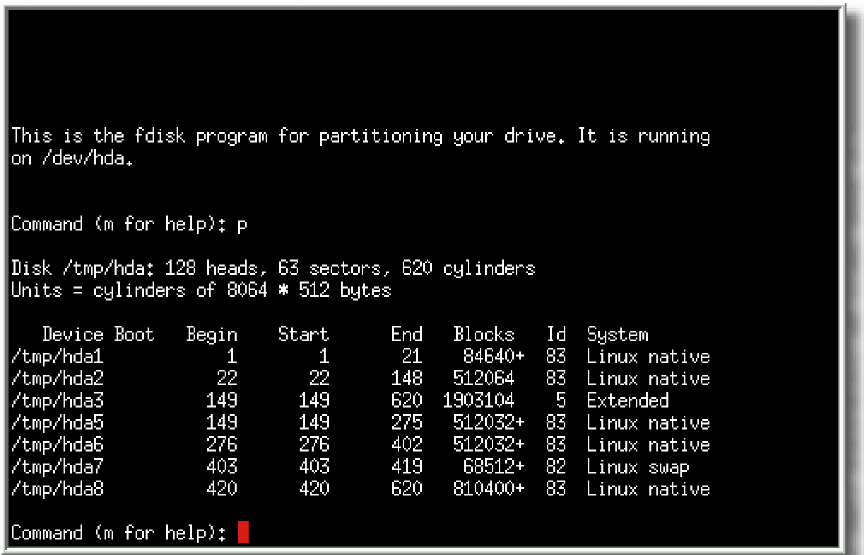
The email received file for my *Hands On Unix* column is bulging with queries from readers who would like to install the popular, free, Unix-like operating system called Linux, but, in approximately this order of priority: they're fearful that they'll mess up their existing (usually Windows) installation; or, they shy away from an operating system that seems to demand a lot of technical knowledge; or, they haven't a clue how to get started.

Fact of life

This three-part workshop is aimed to help Windows users who want to get started with Linux. Let me make it clear from the outset — I'm not a guru. I'm just this guy who happens to have been kicking around Linux, and other Unix flavours, for several years. You don't *need* to be a guru to run Linux, but if you want guru-type help, there's plenty of it about. The good news for you (and me) is that there's a lot of documentation out there. The bad news is that much of it is out of date — well, not really bad news, just a fact of life. When technology moves as fast as Linux, everything on which you lay your hands is automatically out of date to some degree. For Linux documentation, as for web pages in general, you need to learn to read like a good history researcher, critical of the content and conscious of the context.

You don't need to be a guru to run Linux

If you turn to *Hands On Unix* (on page 269) you'll find some suggestions about where and how to start. Here, I am tackling the kind of real-life problems likely to arise with your own installation, but I cannot cover all eventualities. Contact me if you hit snags that I have not mentioned in these pages and we'll try to hack it out together [*"PCW Contacts", p244*].



```
This is the fdisk program for partitioning your drive. It is running
on /dev/hda.

Command (m for help): p

Disk /tmp/hda: 128 heads, 63 sectors, 620 cylinders
Units = cylinders of 8064 * 512 bytes

   Device Boot    Begin     Start      End    Blocks   Id  System
/tmp/hda1             1         1       21     84640+  83  Linux native
/tmp/hda2            22        22      148    512064   83  Linux native
/tmp/hda3           149       149      620   1903104    5  Extended
/tmp/hda5           149       149      275    512032+  83  Linux native
/tmp/hda6           276       276      402    512032+  83  Linux native
/tmp/hda7           403       403      419     68512+  82  Linux swap
/tmp/hda8           420       420      620    810400+  83  Linux native

Command (m for help): 
```

▲ Fig 1 The traditional way to do this is with fdisk. It's more flexible but, as you can see, less user-friendly

My mate Marcus runs a Windows machine which he put together himself six months ago for less than £500. He picked up the motherboard with 32Mb of RAM and a Cyrix 200 processor from a UK company called Big Red at **www.bigred.co.uk** for £125. He added an old Mitsumi four-speed CD-ROM and a 515Mb Western Digital hard drive filleted from a defunct PC, plugged in an NE 2000-compatible Genius 2000 network card and an antique Western Digital/Paradise 90C31-based video card, then connected the whole thing up to a 14in monitor he picked up for £100.

I've picked Marcus' machine for a variety of reasons. I like its oddness: installing Linux on a standard machine can be too easy to be interesting. This one has a non-Intel processor clocked at 150MHz: hardly state-of-the-art, and therefore a chance for Linux to demonstrate its spriteliness at the cheaper end of the market. (Several readers of the *Hands On Unix* column have reported going up to the attic to retrieve an old 386 machine and loading up Linux to bring it back to useful life.)

The other thing that appealed to me about Marcus's machine was that I could get Marcus to do most of the work. The first question was: where were we going to put Linux? The Western Digital IDE hard drive was dedicated to Windows 95. In theory we could shrink the Windows partition to make room, but this would be a squeeze for both Windows and Linux. We decided to put in a second hard drive.

Marcus cannibalised a four-year-old 520Mb Seagate ST3660A from another machine. "The hard disk install was very easy," he says. "There was a spare connector on the IDE cable and I established the new drive as a slave on the primary IDE interface, setting the switches according to the details printed on the drive cover itself." Many older drives may not have this key information handy, in which case it's worth quizzing the relevant web page — www.quantum.com or www.westerndigital.com, for example. If you can't find your drive there, check out the marvellous hard drive database maintained at www.mm.mtu.edu/drives/ by Chris Hooper.

The Linux distribution we chose was the Official version of RedHat 5.1. "Official" in this context means that you get some extra proprietary software, a nice manual and 30 days' free installation support. You can buy it from many places, including ordering it direct from **www.redhat.com**. Another option is the Linux Emporium here in the UK at **www.polo.demon.co.**

uk/emporium.html, where you can also pick up the GPL version (no manual, no support) for a couple of quid.

The Official RedHat distribution comes with a boot floppy (with the GPL version you need to manufacture your own from the CD) but we didn't have to use it because, like many recent PCs, the AmiBIOS v2.4 in Marcus' machine knows how to boot from a CD. This means you just have to insert the RedHat CD in the drive, switch on the machine, and after a moment or two you're looking at a character-based screen with a prompt that just reads "boot:".

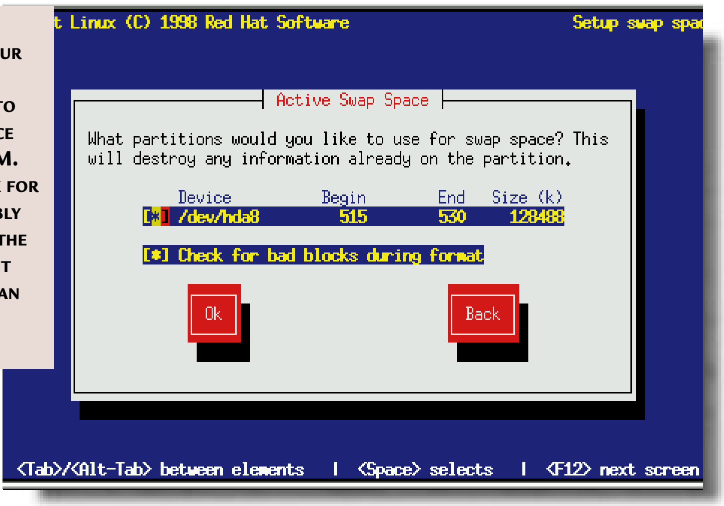
Enter lilo

What do you do next? Well, actually, nothing. This boot prompt is being offered by lilo, the Linux boot loader program. The prompt is there so you can take advantage of the flexibility of lilo, but at this point all you need to do is wait a few seconds (or hit carriage return) and lilo will begin to load a copy of the Linux kernel (the core of the operating system) into the machine's memory from the CD and then automatically start running the installation program.

> ***Installing Linux on a standard machine can be too easy to be interesting***

During this stage you'll see a number of messages sprinting up the screen as various parts of your hardware are detected. If they don't mean anything to you, just ignore them, although they're useful for diagnostics if things go wrong.

☛ **The opening screen** puts up a welcome message and then trots you through some simple questions, like which language you want to use and what your keyboard type is. You're also given a choice of installation methods: you can even install over the network from a shared Windows drive. We settled for "Local CD-ROM".



▸FIG 2 SETTING UP YOUR SWAP PARTITION. THE RECOMMENDATION IS TO MAKE IT AT LEAST TWICE THE SIZE OF YOUR RAM. THE OPTION TO CHECK FOR BAD BLOCKS IS PROBABLY NOT NECESSARY WITH THE NEWER HARD DISKS BUT WE'RE INSTALLING ON AN OLDER DISK

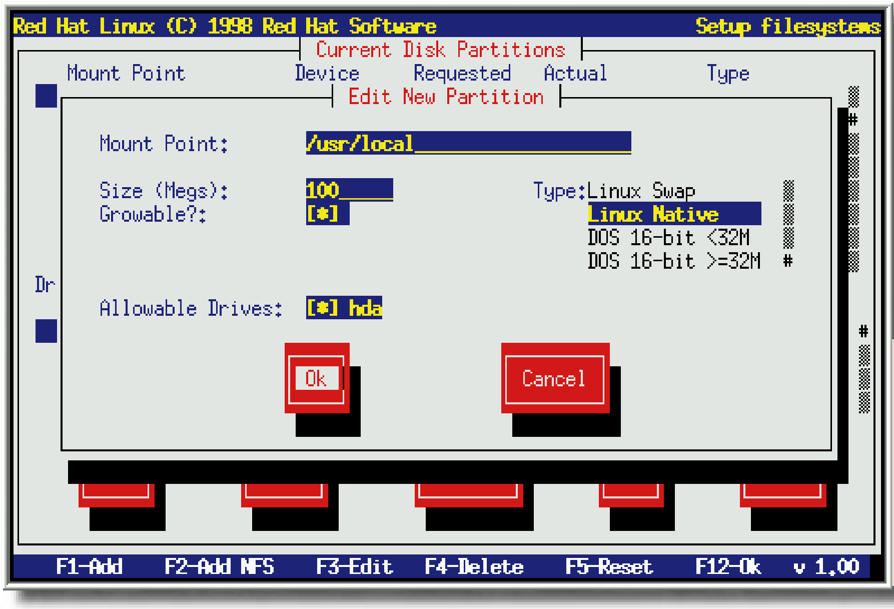
```
t Linux (C) 1998 Red Hat Software                    Setup swap spa

              ┤ Active Swap Space ├

  What partitions would you like to use for swap space? This
  will destroy any information already on the partition.

         Device          Begin       End    Size (k)
     [*] /dev/hda8        515        530     128488

     [*] Check for bad blocks during format

          Ok                                   Back

<Tab>/<Alt-Tab> between elements  |  <Space> selects  |  <F12> next screen
```

☛ **The next screen** asks if you want to install a new Linux system or upgrade an old one. We chose "Install" and were then presented with a choice of using Disk Druid or fdisk [Fig 1] to create our partitions. Both do the same job of allocating areas of the disk and preparing them for the new operating system, but fdisk is the techie tool and Disk Druid the (not much) simpler partitioner for beginners. Whichever you use, by default the installation procedure won't overwrite your existing Windows partition (or partitions). We were extra safe because the Windows stuff was on a different drive. However, partitioning software is potentially dangerous, so tread carefully. (But of course, you did back everything up first, didn't you?)

Disk Druid doesn't offer much in the way of help screens. If you're new to partitioning I strongly recommend spending £35 to get the documented Official RedHat. Even so, Marcus and I poured over the manual for about five minutes before finding what we were looking for: suggestions on how best to use our relatively meagre 500Mb for Linux.

Swap shop

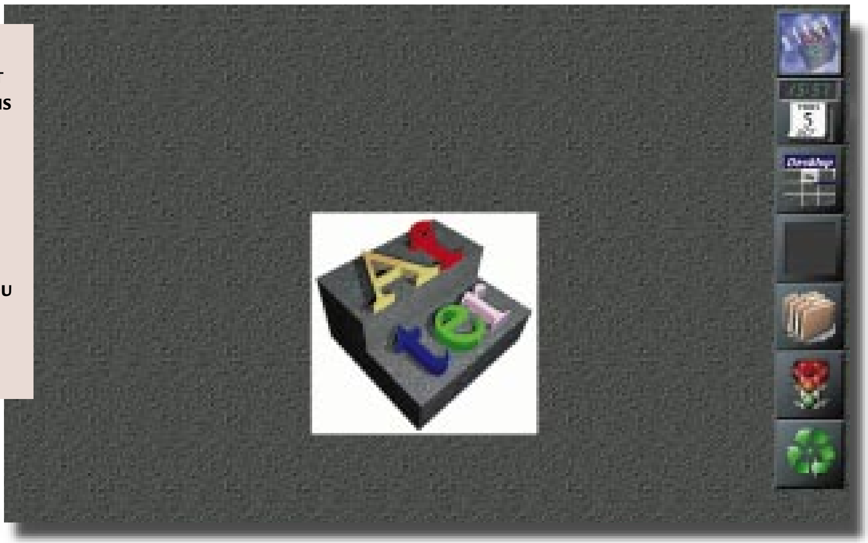
Guided by suggestions on page 26 of the RedHat 5.1 manual we settled for four partitions: a 40Mb swap partition, a 100Mb (growable) root partition, a /home partition of 50Mb, and a 100Mb partition (growable) for /usr. The swap partition [Fig 2] is used for virtual memory (equivalent to Windows' swap file but more efficient), while /home is the directory under which individual users' home directories are stored. So, 50Mb should be fine on a



◂FIG 3 DISK DRUID IS REDHAT'S EASY (WELL, EASIER) WAY OF ALLOCATING PARTITIONS

```
Red Hat Linux (C) 1998 Red Hat Software                 Setup filesystems
                      ┤ Current Disk Partitions ├
   Mount Point            Device    Requested   Actual         Type
                          ┤ Edit New Partition ├

      Mount Point:        /usr/local

      Size (Megs):        100                  Type:Linux Swap
      Growable?:          [*]                       Linux Native
                                                    DOS 16-bit <32M
                                                    DOS 16-bit >=32M
 Dr
      Allowable Drives:   [*] hda

                           Ok              Cancel

   F1-Add    F2-Add NFS    F3-Edit    F4-Delete    F5-Reset    F12-Ok    v 1.00
```

single-user machine — /usr is, broadly speaking, the location for applications, utilities and libraries associated with the local machine, and the root partition is for general system files and the rest. The thinking behind this way of dividing the disk takes us into intermediate-to-advanced territory which I will not cover here. Until recently I had always just created a root partition and a swap partition, as you might prefer to do. Actually, you don't even need a swap partition: Linux will happily swap to a file in the same way as Windows. But you'll get better performance if you dedicate a partition to swapping. By the way, the "growable" option that Disk Druid offers you for partitions [Fig 3, p243] is not as dynamic as it sounds. If you know IBM's AIX implementation of Unix you might think this means a partition which can get bigger on-the-fly. In the RedHat context it just means that Disk Druid will allocate larger space than you designate if more turns out to be available by the time you've finished your allocation, and before you commit your partition table to disk. We ended up with root and /usr as 214Mb each.



▸Fig 4 The default AfterStep screen is what we're aiming for with the installation on Marcus' machine. It's austere and simple but totally configurable if you want more bells and whistles

Partitioning under Disk Druid may strike the seasoned techie as a little weird because it creates logical partitions by default, even if (as now) you only want to put no more than the allowed four partitions on a drive — a situation where I would recommend sticking with primary partitions. (If you don't understand what I'm talking about, ignore this bit — or, better still, read the Linux Partition HowTo at **sunsite.unc.edu/mdw/HOWTO/mini/Partition.html**.)

You are then moved on to the selection of a choice of packages — or rather, groups of packages — to install. You will need to transfer the Linux kernel from the CD-ROM to the hard disk, of course, so this and other essentials like the X Window System, which handles the graphical front-end, are already marked for installation. Amongst others, we chose to tick additional boxes for printer support, DOS/Windows connectivity, graphics manipulation, X games, X multimedia support, Print Server, NFS Server, SMB connectivity, anonymous FTP, a web server and more.

Are you still with me? From my email in-tray I know that many readers get confused at this stage about what to install or not to install. It really doesn't matter too much — everything you need to make the system work at the basic level will be installed anyway, and the package installation can easily be changed later. For example, if you're not a programmer you may shy away from the option to install the C and C++ development system and libraries. That's OK, as it's easy to add what you need later. When you get to know Linux better, you may want them on your system, because a lot of applications and utilities come as source code and you really don't need to be a programmer to compile them.

Everything you need to make the system work at the basic level will be installed anyway...

The high life

At this point, the partitions are created and turned into usable filesystems with a high-level format utility that Unix calls mkfs (don't worry — the installation procedure automates all this for you). With Marcus' machine this took about five minutes. The install procedure then told us that 276Mb-worth of packages needed installing and began the job, displaying the estimated time to install everything as soon as the system got familiar with the speed at which things were moving (7mins 45secs). So, pleased with our work so far, we decided to reward ourselves with a coffee break. But then — uh-oh...

The screen read: "RPM install of tix failed: unpacking of archive failed on file /usr/lib/tix4.1/ListNBk.tcl". Several other failure messages followed. The installation was becoming horribly unravelled. RPM is the RedHat package manager, a Linux standard for application installation and removal, and the RPM packages were failing to unpack on the /usr partition. We both spotted the reason at the same time: at 214Mb, we'd made the /usr partition too small and had run out of space.

Fancy that

We were going to have to start the installation all over again. We had two choices: either be less ambitious about the number of packages we installed, or find some more disk space. We decided to give Linux more room, and this meant downsizing the existing Windows partition on the first hard disk. There are some fancy packages for doing this (see the Partition Manager review in *Hands On Unix*, p272). But there's also good-old fips, the free software partitioner, so fips is what we'll start with next month.

PCW CONTACTS

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