Back in the old routine

Chris Bidmead knocks the cron daemon into shape for more efficient system backup.

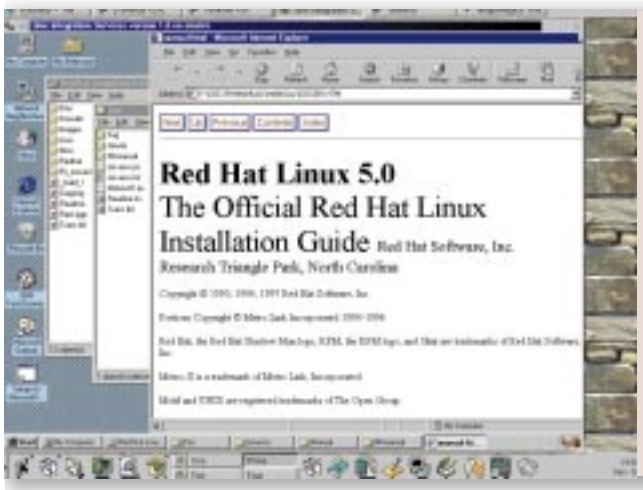
I don't write nearly enough about backup in this column. You can't get enough of the stuff. But having said that, I confess I haven't exactly been methodical about backing up my own system here. I do it when I feel like it — which is quite often. But it should really be a job for the cron daemon. It takes a little organising, but I'm beginning to get into shape.

Actually, the way I'm going about things isn't too silly. I strongly believe in running routines manually first, until you understand them and have checked they're doing what you intend. Then you can encapsulate them into scripts. Yes, you can write scripts straight away if you're really sure of what you're doing, but I'm not in that position. So I'm still currently doing manual primary backups of the whole system, or at least those parts of it that I can't recover from CDs. However, I've started automating the incremental backup — the files that are changed or added between each primary backup. To do this I've adapted a script that comes with the GNU tar distribution. Fig 1 is my version, called incr.

The tar part of this script is actually a single line — note the use of the backslash to lay it out in an easy-to-read way. You'll see how the script uses the $then and $now local variables to write a Volume label (the -V switch) to identify the session. My tapelist script [*see p243*] can

[FIG 1]

```
#!/bin/bash
 # (based on Dump thingie from tar.info)
 # CHB 21 Nov 98
 # assumes date.last.backup contains the date
 # of the last backup (but you guessed that).
 #
 # set up environmental variable TAPE
 # (which mt and tar take as their default)
 TAPE=/dev/nst0 ; export TAPE
 # Yes, you have to be specific about this if
 # cron is going to be running this, 'cos cron runs
 # in its own environment.
 # set up some local variables
 statefile=/var/state/date.last.backup
 now=`date`
 then=`cat $statefile`
 # find the end of data on the tape
 mt eod
 # create, incrementally, preserve permissions, →
    be verbose
 tar -c -G -p -v\
 -N "$then"\
 -V "Incr: $then to $now"\
 /home/bidmead\
 /mnt/james_ii/update\
 /mnt/james_ii/bidsown\
 /mnt/james_ii/to_be_in\
 /mnt/abbott/wells_d/accounts
 # update the state file
 mv $statefile $statefile.old
 echo $now > $statefile
```



Drop of the hat

A BIT OF A TEASE, THIS PICTURE. READER ANDREW SARGENT <ANDREWSARGENT@HOTMAIL.COM> WROTE TO SAY HE CAN'T READ THE RED HAT DOCUMENTATION FROM WINDOWS 95. SURE, SOME OF IT IS GZIPPED, BUT THERE ARE VERSIONS OF GZIP FOR WINDOWS. AND MANY OF THE DOCS ARE HTML, WHICH, AS YOU CAN SEE, MICROSOFT EXPLORER RUNNING ON WINDOWS NT ON MY SIEMENS-NIXDORF PRIMERGY SERVER READS JUST FINE. THE EAGLE-EYED AMONG YOU WILL HAVE SPOTTED THAT THE WINDOWS NT 4.0 DESKTOP SEEMS TO BE RUNNING UNDER KDE. IT IS. THE WORKSTATION HERE IS THE DELL POWEREDGE WITH ITS NEWLY INSTALLED S.U.S.E 5.3, AND THE WINDOWS NT DESKTOP APPEARS THERE COURTESY OF CITRIX METAFRAME.

read these labels and their associated block numbers, so I can subsequently use mt seek <blocknumber> to wind to any particular session. The GNU version of this script saves the date information in a file that's in the same directory as the script. Messing around with Caldera OpenLinux I discovered a world-writeable directory called /var/state in which this kind of information can be stored. This doesn't seem to be a standard — my Red Hat installation doesn't have it — but it seems to me like a good idea, and something like this is worth setting up in whatever version of Unix you may be using.

Before installing it in the cron system, I tested the script by running it manually. You'll see that its first move is to wind the tape to the end of data (mt eod). This enables me to store multiple backup sessions on a single tape — handy, because my Hewlett-Packard SureStore DAT24 can put 24Gb on a DDS-3 tape.

Once sure this was all working correctly, I was ready to move the script into the /etc/cron.d/Daily directory. Not all Unix systems have this facility set up, although it's easy enough to organise. If yours doesn't have something like this, you'll need to add an entry for incr into /etc/crontab, the system cron file.

My Caldera OpenLinux /etc/crontab has an entry that looks like this (read as one line):

```
05 5 * * * root
[ -x/usr/sbin/cronloop ]
&& /usr/sbin/cronloop Daily
```

This means that at five minutes past five every day (the first * in the line) cron will run /usr/sbin/cronloop with the parameter 'Daily'. (For Linux users, man 5 crontab will explain how crontabs work.) On my OpenLinux system /usr/sbin/cronloop is a script that looks in the directory (under /etc/cron.d) named in the parameter and runs any executable files it finds there. I've put my own note in this directory to remind me of what's going on. (I find it very useful to leave README's around the place to help me remember how I've set things up or changed them. Some of you may alternatively be using a central logging system for this: if this works for you, tell me how you do it.)

The executables in this directory are run daily due to an entry in /etc/crontab which runs /usr/sbin/cronloop, a script. Cron won't mess with this README file because /usr/sbin/cronloop only acts on executables. The basic idea seems to be to put only symlinks in this directory. They have funny names beginning with numbers, presumably so they sort into a predictable order, and are run in that order. I've added (so far) 10tapelist and 12incr, both links to scripts of mine in /usr/local/bin (/usr/local/sbin might have been more appropriate). This will

[FIG 2]

```
#! /bin/bash
# wind through the tape stopping at each
filemark
# to examine the next block, hopefully an
archive
# label or a meaningful initial tarball entry

TAPE=/dev/nst0 ; export TAPE
INPUT=$TAPE
EOD="<<eod>>"
BLOCK='mt tell | cut -f 3 -d " "'
DATA='dd if=$INPUT count=1 2> /dev/null'

mt rewind

printf "`eval $BLOCK`\t`eval $DATA`\n"
while mt fsf 2> /dev/null ; do
printf "`eval $BLOCK`\t`eval $DATA`\n"
done

printf "${EOD}\n"
mt rewind
```

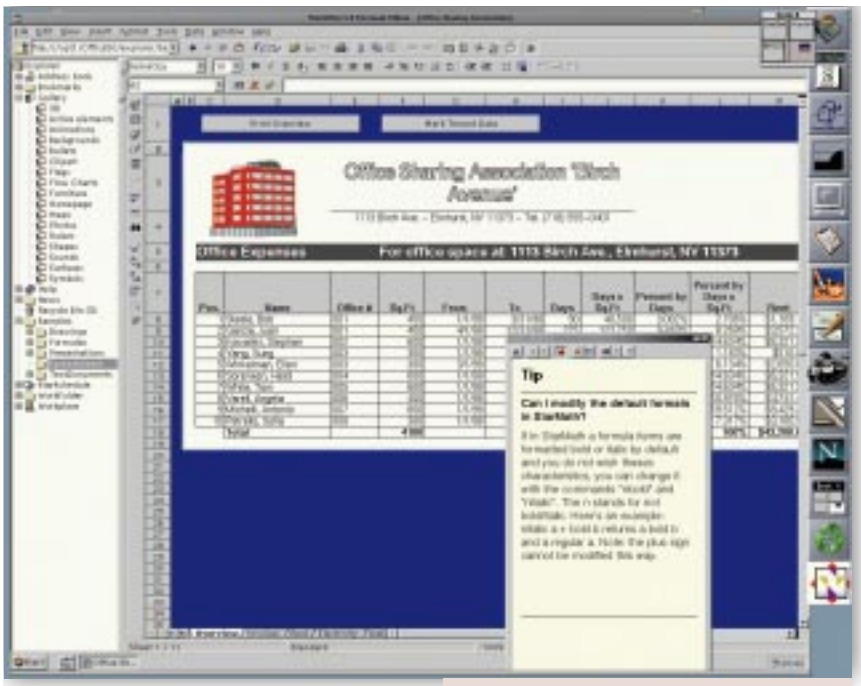
SIR CLIVE AS THE FATHER OF LINUX

Scratching around the internet archives, I found several references to the computer on which Linus Torvalds got his early programming experience. It was a Sinclair QL, the 'Quantum Leap' machine that Sir Clive Sinclair [*pictured, right*] introduced in 1984 — the same year the Mac arrived. Sir Clive is mostly into ear radios and folding bicycles these days, but I thought it would be fun to ring him up and tell him about his place in the history of Torvalds' plan for world domination, in case he didn't know.

"Linux?" Sir Clive said. "What's Linux?" With the swelling din in the media these days about 'the new Windows Killer', it's becoming increasingly difficult to find anybody to bore with my favourite operating system. So Sir Clive is a godsend.

I'm planning to do lunch with him next week and give him the full five yards — particularly as I know he's always been interested in getting back into the computer game at some stage.

But I shall have to be diplomatic about exactly why Torvalds got into low-level bit twiddling on the QL in the first place. He really just wanted to play games, but his keyboard packed up. He couldn't find another QL keyboard, so he had to write a driver for a different kind. Then he went on to do another driver to attach a PC-style disk drive: the QL came with those ingenious but dubious 'microdrive' tape thingies.



run an incremental backup each night (primary backups are still manual) followed by a list of what's on the tape, which cron will mail to me (well, actually to root, but that aliases to bidmead — see /etc/aliases).

Tapelist is the script I published here exactly a year ago. In case you missed it, Fig 2 shows the updated version.

You'll see a couple of notes about environmental variables. The warning is worth reiterating. By default, tar and mt both operate on a device defined in the environmental variable TAPE. I have this set by default, and it took me ages to work out why incr worked perfectly well when run interactively but mysteriously seemed to do nothing when activated by cron. Well, I thought it was doing nothing, but my /var partition kept filling up with junk of some kind. The 'junk' turned out to be the tar archive itself. Cron knows nothing about the user environment, and therefore didn't know I'd set TAPE=/dev/nst0. If TAPE isn't set, the output of tar goes to stdout. Cron helpfully emails this to the appropriate user, so the entire set of backed-up data was turning up in my /var/spool/mail directory! The solution is to set TAPE in the scripts themselves, as Fig 2.

incr worked perfectly when run interactively, but seemed to do nothing when activated by cron

What a star!

IT WAS RELATIVELY EASY GETTING HOLD OF STAROFFICE 5.0, THANKS TO MR BATCH FTP AT MY ISP, DEMON INTERNET FTP SERVER. I JUST MAILED FTP@DEMON.NET WITH THE URL OF THE FILE, AND DEMON'S FTP DAEMON COLLECTED ALL 65MB OF SO05_01.TAR WHILE I WAS OFF-LINE AND STORED IT AT ITS OWN FTP SITE FROM WHICH I COULD DOWNLOAD IT AT FULL SPEED. NEXT MONTH I'LL BE SAYING MORE ABOUT STAROFFICE 5.0, HOPEFULLY RUNNING IT ALONGSIDE THE BRAND-NEW LINUX OFFICE 99 WHICH MARTIN HOUSTON IS SENDING ME.

Ring out the old, ring in the new

I've been talking to Michael Waddicor, who heads up the Bradford-based computer services company, Aquila Vision. Michael is fanatical about operating systems, and supplies Microsoft- and Unix-based solutions to his customers. But he's been a long-time personal user of Linux, and is delighted to have watched it develop to the point where he can confidently recommend it to his commercial customers, along with full professional support. He's offering good prices on a wide range of Linux distributions, so check out his web page at http://www.aquila-vision.co.uk/. Michael tells me he's happy to give phone support to his Linux customers: ring him after 5pm if you want to be sure of getting through.

On a sad note, I'm very sorry to hear that Martin Houston at http://www.deluxe-tech has decided not to continue selling S.u.S.E Linux from his web site. He tells me it was something he used to be able to do in his spare time, but the market has now grown to the point where selling Linux is a full-time job and it's just got too much for him to handle. Martin has been a passionate and active supporter of all things Linux, and was the first person to draw my attention to the great features of the S.u.S.E distribution. Thanks for all you've done, Martin.

Installing a kernel from the boot floppy

During the YaST install of S.u.S.E. Linux 5.3, the kernel I needed to install was on the boot floppy; none of those offered on the CD-ROM would have worked (*see Linux Workshop, page 212*).

To keep the installation routine happy, I picked an arbitrary kernel from the CD; but the moment that was transferred to hard disk, I switched virtual consoles (Ctl-Alt-Fn, where n is between 1 and 6) to get to a command line prompt.

The floppy isn't mounted at all during the installation, so my first move is to create a mount point, just an empty directory called, arbitrarily, /fd, and mount the floppy there. The two commands are:

```
mkdir /fd
```

and

```
mount /dev/fd0 /fd
```

The YaST installation process has temporarily mounted the Linux root partition under the /mnt directory, so the next command I run from the virtual console:

```
cp /fd/linux /mnt
```

copies /fd/linux (the kernel on the floppy) to /mnt (which will be simply / when I go live).

I then delete vmlinuz (the kernel YaST had just installed) and rename linux to vmlinuz, as this is the kernel name that YaST is expecting. Then I switch back to the YaST installation, which now runs /sbin/lilo, setting up the low-level boot structure, and we're done.

PCW CONTACTS

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