Leonardo and the Gimp

Chris Bidmead finishes off an old master and gets back into the Unix flow with a little 'piping'.

Last month I mentioned how the Gimp (www.gimp.org) running on Linux helped me show the detail behind my mate Marcus' scanned-in Old Master. The 'X-ray' picture we arrived at using Image/Colors/Equalise falsified the colours. The ultimate goal was to bring out the detail, while keeping the general colour scheme intact.

The approach I chose was to create three layers: the original picture left as background; a layer comprising the central figures of the two embracing cherubs; and between these a 'sharpener' layer that I'll explain in a moment.

I wanted to keep the cherubs separate so that any changes to the rest of the picture needn't affect the delicate skin rendering (hopefully by Leonardo's own hand). To isolate the cherubs I used the lasso tool to create a loose selection outline around them. Copying this selection and pasting it back again creates a temporary 'floating' layer that will drop back onto the background when you click outside the selection.

The trick is not to click outside the selection: instead I used the right mouse button to pop up the Gimp's Layers and Channels information box. If you edit the name of the floating layer from 'Floating' to, say, 'Cherubs' (double-clicking on the name pops up an edit box), the new layer becomes permanent.

Of course, the loose lasso selection had left part of the background adhering to the outline of the cherubs. A straightforward approach to getting rid of this would have been to zoom in and carefully remove it by hand using the Gimp's eraser tool. As with other professional-quality, image-editing packages this tool works like a brush, and so can be given a variety of different shapes and sizes and a spectrum of different looks from soft to hard. Like everything you do in Gimp, erasing can be undone through multiple stages, so even if you're using a mouse rather than a proper digital drawing tool, you should still be able to achieve a perfect outline.

As a shortcut, though, I took advantage of the relatively uniform darkness of the background and used the Select By Color tool. You can choose the 'fuzziness threshold' to define the size of the colour range that will get selected. Setting a Fuzziness Threshold of 15 picked up most of the extraneous dark area, and I could collect up the rest by switching the tool to Add mode and clicking on the individual islands of unwanted colour. To see clearly what I was doing, I clicked on the appropriate eye icons in the Layers and Channels info box, to make the background and sharpener layers temporarily invisible.

Screenshot 1: You can use the Gimp's Lasso tool to create a rough outline around the central subject. The first time I came across this tool was on a Macintosh back in 1984, but image manipulation has come a long way since then

With the cherubs now isolated in their own layer it was time to develop the sharpener layer. Here I operated on the whole picture with a range of different tools and techniques to pick out the hidden detail. There's only room to give you the general flavour of the hours of fun I had with this. One obvious trick is to increase the brightness and contrast simultaneously so that the darkest parts are held back but the brighter details

emerge more clearly. You can also operate selectively within different colour domains – for example by splitting the picture into its red, green and blue components (using the Channels tab on the Layers & Channels tool). And finally there's a Sharpen filter under the Filters/Enhance menu that, used discreetly, can emphasise the drawing quality of outline details without making too much of a mess of the fill areas.

The net result was a kind of 'colour etching' that over-characterised the detail. It certainly didn't represent the final result I was looking for, but contained ingredients that I wanted to add in a controllable way. The key to this is that the transparency of each layer is individually adjustable. So if the 'sharpener' layer sits on top of the untouched original layer, by adjusting the upper layer's transparency I can arrive at the exact level of enhancement I'm looking for.

The cherubs are on the top layer, so if I make this completely opaque I can add them back into the picture in their original state. But if I dared to think the handiwork of the master could be strengthened by a little sharpening I could easily do this. By gently increasing the transparency of the Cherubs layer, the oversharpened rendition of the Cherubs in the middle layer would show through, until I arrived at the effect I was looking for.

Screenshot 2: The Gimp's Color by Selection tool is a useful shortcut for defining the outline of an image. It's possible to apply it to the whole background layer, but you run the risk of picking up extra arbitrary background detail across the whole picture that happens to come close to the flesh tones

explore the way simple standard tools can be pulled together to perform specific ad hoc tasks – a key Unix feature, for those not hypnotised by the glamour of the graphical user interface. 'Piping' is the technique of passing the output from one tool into the input of another and I used this to demonstrate the rather banal task of displaying the name of the newest file in a particular directory.

Standard tools can be pulled together to perform specific ad hoc tasks – a key Unix feature

OK, not a lot of basic Unix here. But this all started last month, if you remember, with a cry from a reader who had Linux up and running but didn't know how to do anything useful with it. The Gimp comes bundled with most Linux distributions and is available as source code – so it should run on virtually any flavour of Unix. There's even a version in development for OS/2. So the only other thing you need is an old master. Check out Marcus' collection at www.lostleonardo.com.

More power to the pipe

Talking of basic Unix, back in February I exposed you to some of my doodlings at the command line. The idea was to

One of the solutions I came up with, after much huffing and puffing, was:

```
ls -lt /etc | grep "\-r" | ↙
head -1 | rev | cut -f 1 -d ↙
" " | rev
```

(*Key: ↙ code string continues*)

... but I did suggest you might be able to offer something more elegant. I wasn't wrong. My own tentative experiments precipitated a gale of email from you, and I learnt a lot.

Alex Dicks (alexander.dicks@christ-church.oxford.ac.uk) suggested an elaborate solution using sed, which he describes himself as 'messy' (I agree, Alex). He goes on to say: 'There ought to be a command 'cols' which would be able to automatically extract a vertical strip from text. I cannot see one, at least on my machine...'

Some news about that in a moment, Alex. Meanwhile, thanks for the correction to my statement that the first summary line of ls -l 'gives a total of the number of files'. As Alex rightly points out, the summary counts the total size in kilobytes.

Steve Williamson (steve@chipping.demon.co.uk) picks up on what may have been another sleepy moment of mine: 'You said "in a line whose fields are broken by one or more spaces we have no way of counting to the ninth field." With Unix there is always a way. In this case, you can forget about counting and squash all continuous spaces into one space using 'tr'. I suggest something like this (see Fig 1).'

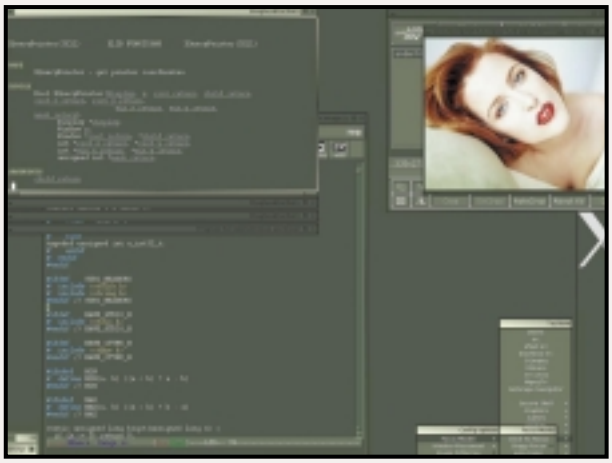
Excellent point, Steve. The tr -s [" "]

FIG 1

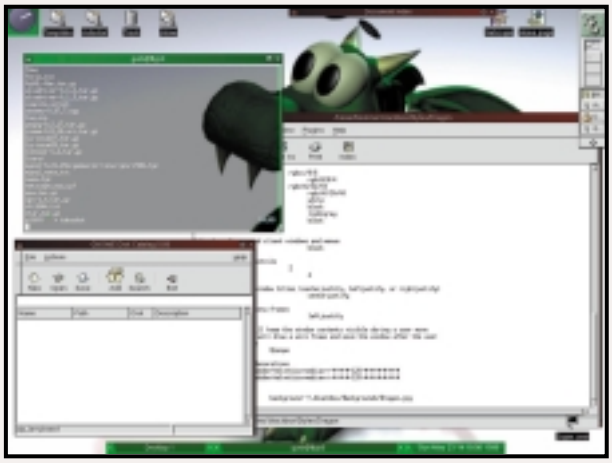
```
ls -lt | grep ^[-] | ↙
head -1 | tr -s [" "] | ↙
cut -d" " -f9-
#
#          ^            ^
#          clearer than ↙
squash repeating spaces
#          having to both↙
to 1 space so that cut
#          escape & quote↙
will work with a fixed
#             to select ↙
regular  column number1
```

(*Key: ↙ code string continues*)

(actually we don't need the square brackets here, do we?) ensures that all multiple spaces are reduced to single spaces, so we can count the fields accurately from the beginning of the



Above is the Blackbox window manager for Linux, written by Brad Hughes. Russell Howe (rhowe@wiss.co.uk) recommended it as 'a breath of fresh air'. It's designed to be fast and light, and Russell confirms that it 'runs like greased lightning' on his 486DX2/66. See the Blackbox site at http://blackbox.alug.org.



With a little tweaking (a more detailed explanation can be found at http://gnome.windsofstorm.net) you can substitute Blackbox as the window manager for Gnome – see the screenshot above. This combines faster performance with the luxury of a fully-featured desktop.

string. One possible gotcha is that we'll change the name of any (rare, presumably) file that has double or more embedded spaces. More fundamentally, though, using a field-divider character like a space isn't the best way of slicing up the string, as we'll see in a moment.

Martin Lester proposed using ls –colour and a pipe through sed (to say nothing of a few greps, cuts and revs) to filter out files from non-file directory entries. '...But I don't like it because it's ugly,' he says. I think so too, but Martin also offered the much better:

```
ls -tpQ | grep -v '[/@|=]$' ↙
| head -1 | cut -c2- | rev | ↙
cut -c2- | rev
```

This summarises a couple of nice ideas that other readers offered too: the use of the -Q and -p switches. -Q is there to persuade ls to wrap quote marks round file names (to get round the embedded spaces problem), and the -p puts a special identifying character at the end of non-file directory entries that can be used subsequently to filter them out with grep.

But I think the winner has to be Paul Leyland (pleyland@microsoft.com), who stepped in with the very tool Alex Dicks was looking for. 'I must say you made heavy weather of it. ;-) Here's my solution, without all that faffing around with tail, rev, cut ... and still getting the wrong answer!

```
ls -lt | egrep -v '^total|^d'↙
| colrm 1 55 | head -1
```

(Tested on RedHat Linux 6.0)

'Note that you can match more than one pattern in egrep, and that colrm is wonderful for ripping out junk from fixed-format output.'

Thanks, Paul. Like Alex, I didn't know about colrm, which comes as standard with the GNU textutils. Other readers have pointed out that you can do the same thing in this case with cut -c56-, because the terminal – means 'to the end of the line'. Alex (above) uses something similar. The fundamental point is that ls -l outputs lines that can be split in an entirely controlled way by column counting. Field counting, based on spaces or whatever, just isn't the right way to do it, as I now realise.

Paul goes on to say: 'I agree that your solution may be more pedagogically useful as it introduces more tools...' That was certainly part of the point, Paul. Thanks for the fig leaf to cover my embarrassment. '...but mine has no problem with spaces in filenames, symlinks and so on. If you really want to omit named pipes, device special files, and such exotica, the modifications should be obvious – just look for the leading "-" but leave out the "r" so you still pick up the unreadable files.'

I think probably the best way to do this, Paul, would be an OR-NOT in the egrep, like this:

```
ls -lt | egrep -v '^total|^↙
[^-]'  | colrm 1 55 | head -1
```

Readers new to regular expressions need to know that in the first two instances ^ means 'at the beginning of the line'. But inside the square brackets it means NOT, so here we're filtering out all lines that begin with the word 'total', as well as all lines that do not begin with '-' (the marker for regular files).

This should be bullet-proof against all non-file entries. Try it in your /dev directory, which is probably packed with every file-type beastie known to Unix.

Thanks to the many other readers who don't get a mention here. If you think I've overlooked a solution of yours that is better than Paul's, the court of appeal is open.

By the way, for those who spotted the m-word in Paul's email address, he adds: 'Lest you are wondering why someone at microsoft.com should be using Linux, I ought to point out that I'm at MS Research in Cambridge. Research people tend to have rather wide-ranging tastes.'

You can match more than one pattern in egrep and colrm is wonderful for ripping out junk

CONTACTS

Chris Bidmead welcomes your comments on the Unix column. Contact him via the *PCW* editorial office or email: **unix@pcw.co.uk**