

UNIX Basics

by Peter Collinson, Hillside Systems



DENISE ORTAKALES

A Century

Tempus fugit. This is my 100th article for *SunExpert*. Because this column has appeared in every issue, you can deduce that this is the 100th edition of the magazine too. I thought the milestone provided an opportunity for some introspection on the years that have zoomed past since I started to write that first article in 1989.

Back in 1989, I had just left my safe and secure employment as a lecturer in Computer Science at the University of Kent, Canterbury, UK. A lecturer translates into an associate professor in U.S. terms. I'd been there for 15 years, having considerable fun. I had become a UNIX guru; ran the first UNIX VAX11/780 in the United Kingdom, providing UNIX service to all the users at the university; and was deeply involved in the kernel work needed to glue that machine (and others once the code was done) onto our campuswide local-area network. The network was based on Cambridge Ring technology, at a time before there were commercially

available local-area networks.

I'd also fallen into the role of the first commercial network provider in the United Kingdom by enabling the UUCP network to reach our shores. I then found that I needed to fund the communications costs and started to sell the service to academic and commercial institutions. We called this network UKnet. I remained involved with UKnet and the company that it became until it was sold to PSInet some three years ago.

Anyway, in May 1989, I was suddenly a freelance computer consultant. I couldn't conceive of not having a UNIX machine to work with. Actually, at that time, having a UNIX machine at home was not as easy or cheap as it is today. There were no BSD clones, and Linux was some way down the line. You could look to the commercial market and buy Xenix, but that was like stepping back in time to UNIX Version 7. To be current with things that were happening, you needed a Sun. SunOS was

the "reference" operating system; it was ubiquitous in universities worldwide. Most, if not all, freely available software was created by users of SunOS for the SunOS platform.

The cheapest machine running SunOS that you could buy was a Sun 386i. This cost a five-figure sum in British pounds, actually pretty nearly the same sum in U.S. dollars, around 10,000. (The price differential is not caused by taxes and tariffs, which is a popular U.S. myth about European pricing; it's what the market is used to, companies charging what the market will bear.) Anyway, I dipped into our family savings to buy the machine. The Sun 386i seemed fast at the time. One factor was that I was the only one using it. I was used to shared computers at the university.

There were many problems with the Sun 386i; for example, its internal clock was massively inaccurate. The main problem was overheating. In the summer, my non-air-conditioned office

UNIX Basics

became hot, and the machine turned itself into a toaster. I remarked in an article at the time (“Environment Variables,” July 1990, Page 29):

“It’s early May, and we have been having a heat wave in the United Kingdom. My Sun 386i has been suffering a little from heat exhaustion and began to weep, wail and gnash its teeth mightily about servos not working properly. Allowing the machine to cool down meant that all was well, so I decided that I should take the opportunity to remove the filth that builds up inside the machine, clogging various grilles and fans. A clean grille improves the internal airflow and cools the disk down just below marshmallow-browning temperature. A user-serviceable air filter would be a good add-on for these machines. However, while working out which four screws were needed to remove the front grille, I removed the plastic skin on the back of the machine—to discover a lot of embossed signatures.” The names of all the people involved in the project had been placed permanently in the metal casing. A nice touch.

I was more than pleased when later Sun decided to get out of Intel-based machines and made me an offer of a cheap upgrade to a SPARCstation I. It had loads more MIPS and a more “standard” operating system.

From the start, I’d been on the world network, and working for UKnet meant that it was convenient for the company to allow me to continue to log into its machines using a leased line at slow speeds. At that time, I used a UUCP connection to ship email and could have had news if I’d wanted it. Anyway, I was well set up in terms of computing power and communications when Mark Seiden mailed me about this new magazine that he was working on, and would I like to write for it? Sure, why not?

The Articles

I had not actually considered that people might be prepared to pay me for writing. Up to that point, my literary output had consisted of my Ph.D thesis, some academic papers, zillions of UNIX man pages and a bunch of pieces for the newsletters of two UNIX user groups: The European group, then called the EUUG, and the U.S. one, USENIX. The notion of writing commercially had not occurred to me.

Some short time later, Mark Seiden mailed to ask me what I might like to write about? I plumped for UNIX Basics, because it was an open-ended topic and I could easily think of at least four possible subjects that would work as articles.

Writing a monthly column is somewhat strange. You are now reading an article that has around 3,500 words and covers a lot of space in the magazine. However, a chapter in a book is considerably longer. The length—which has crept up over the years—and my desire to have a complete “story” in each article, limits the topics that can be covered. Some topics are too small to be made into a complete article, so you have to find some strand that can be used to group things together; other topics are too large. For instance, I wrote about the `make` command back in 1992, as a two-part article (“Make: Parts I and II,” February and March 1992, Pages 34 and 26, respectively). I am not happy with making

two consecutive issues contain one topic. It can be quite hard to make each article of the pair into something that is also self-contained. Also, if the reader is not interested in that topic, or knows about it already, then he knows in advance that next month’s article won’t be of interest either.

I think that when writing, you have to have some idea of who your reader is. Broadly, I’ve tried to aim my articles at some mythical “interested” UNIX user. There are UNIX users who are interested in practical help, so a proportion of the articles are basically “here’s some tips and tricks.” There

*Broadly,
I’ve tried
to aim
my articles
at some
mythical
‘interested’
UNIX user.*



are UNIX users who are interested in expanding their knowledge of how to get their work done, so some of the articles are introductory material on UNIX tools and utilities. There are UNIX users who are interested in finding out how things work, so a proportion are about the underlying mechanisms with some free bits of computer science thrown in for good measure. Of course, articles don’t always split cleanly along these lines. There are usually elements of all three in every article.

A tiny proportion of the articles are what I call “self-indulgent,” encapsulating some topic that’s interesting me at the moment but perhaps does not quite fall under the “UNIX Basics” banner. I guess that you are now reading one of those. Most of the ideas for articles spring from something I am doing, so they are often topical for me when I write them. The time delay that publishing imposes sometimes means that my currency in that topic is long gone by the time I get email from someone querying something in the article.

I also find that there are some readers who consume the articles to discover whether I know what I am talking about (no one is infallible) or to find out whether they know all there is to know about some topic or other.

I have one other mental ground rule. I try to write about the programs and utilities that you will find on your Sun machine when the system was installed from the CD. I must confess that I don’t always stick to this rule, but I am sure it’s a reasonable guideline. There is a great deal of high-quality free software available on the Net, but I suspect many people are unable to install and use it. I don’t necessarily mean that they cannot pull the file with FTP, run the `Configure` script and type `make`. But they are prohibited from doing so, perhaps by company policy, perhaps by unimaginative management,

perhaps by lack of resources, perhaps by lack of privilege on the machine, perhaps by lack of confidence in their ability to install or test the result. After all, free software is always *caveat emptor*—let the buyer beware. Once installed it will require maintenance, and that maintenance costs time. Of course, it's gotten harder with Solaris, when there is no C compiler on the machine, and installing one is either a matter of money or some hard work.

The strange thing about being a writer who started as a techie is that I have no idea why people like the stuff I write. I know that at least some people do, because unlike many authors, email provides me with frequent feedback, people can ask questions, send in comments, complain and criticize. Most of the email I get seems to imply that people like the articles. I get very few bits of purely hate mail, except when I tickle someone's religious convictions about some editor, shell or system that I have criticized.

However, I have no idea how to improve the articles. *SunExpert* has a good policy of providing its authors with a proof of their article after copyediting, so I've learned quite a bit about improving my writing technique from seeing what these excellent people have changed in my text. Of course, they have a special problem with me: I write with English spelling, and they need to alter it into something that's palatable for U.S. readers. I also occasionally use an English idiom or make an English cultural reference. Usually, these are inadvertent. I try to check things that I am writing if I am suspicious that they won't port into U.S. culture. Often English idioms work, although I recall that I "came out" as a Brit after I had used the Victorian game of Snakes and Ladders to illustrate symbolic links. It seems that in some parts of the United States, this game has become known as "Chutes and Ladders," a practical game about building. Someone spotted this and emailed, "Ah ha, you're a Brit!"

The Systems

Well, what's happened to computing in these past eight years? As more and more machines have been sold, hardware prices have dropped dramatically, with disks and memory becoming cheap(ish). To replace my SPARCstation II today with some equivalent desktop machine would cost about half of what I paid eight years ago. It goes without saying that the system would run much faster. It seems to me, though, that much of the speed increases in CPUs have not been passed onto users, but eaten up in kernel code by the overhead in supporting objects in the language that the program was written in, or by the interpreted programs that are now in vogue.

Certainly, Sun has lost its preeminent position as the main UNIX software reference platform. To be fair, the developers at Sun have worked hard to make the Solaris system environment as "open" as possible. A C source program written for any UNIX system is likely to compile out of the box on your Solaris system. There are masses of backwards compatibility libraries and kernel hooks in a Solaris system to ensure that code will compile and run.

I resisted the move to Solaris for a considerable time. Waiting was the right thing to do, in hindsight. It took Sun

about two years to get Solaris to the point where it performs as well as SunOS does. The move to Solaris was fairly painless, I suppose. I must confess that I have never become used to the "new" arguments to the `ps` command, although I did try for some time. But my reflexes are too strong. I always type

```
ps ax
```

I now always use the "old" SunOS version that Sun thoughtfully still provides.

I was prompted to move to Solaris (see "Moving to Solaris," May 1996, Page 20) by the adoption of the Common Desktop Environment (CDE) running on a "proper" X11 Window system. I like the CDE virtual desktop, but I cannot say that I make much use of the drag-and-drop interface that CDE provides. I am still using my windowing system to present me with a large number of virtual text terminals into which I type my commands or edit text. You could say that I use an interpreted graphical program to deal with mail. Brent Welch's `exmh` program is written in Tcl/Tk and does all the things I need in a mail program, with the bonus that I can access mail via the command line-based `mh` suite should I want to, and I do, when I am far from home dealing with mail by using telnet over the Internet.

The Internet

I suppose we all concede that the big world change in the past eight years has been the growth of the Internet. From my perspective, wide-area networking has changed from store-and-forward UUCP connections to machines talking with IP, allowing me access out to the world, and for the world to have access to my machines. Sun has always been well placed to generate systems that allow you to create servers accessible from IP. There have been a great swath of articles in this column that examine how the Internet works and how to use it.

The Internet has changed my life considerably. I've always telecommuted these articles from Canterbury in the bottom right-hand corner of England to the *SunExpert* offices in the top right-hand corner of the United States. I'm still originating articles in `troff`, largely because I am used to it, but also because actual page layout is not an issue to article writers. Using `troff` means I can make font selections and set up subheads and paragraphs. Also, and this is a big plus, I end up with a text file with which I can use the standard UNIX tools, notably `grep`. However, the folks in the *SunExpert* editorial office use different systems, so I convert my `troff` input into Microsoft Word format using some Word macros (don't get too excited; the macros are very limited and only understand the subset of `troff` that I use), and use Word to generate the Rich Text Format file that is dispatched by email to the editorial team.

The Web has been a big factor in my life in recent times, and again this has been reflected in my articles. I now telecommute 6,000 miles from Canterbury to Berkeley, CA, to Webmaster the USENIX Web site. It's viable not only to use telnet over that distance, but also to use it within an encrypted `ssh` data stream.

UNIX Basics

The Growth of Microsoft

Of course, another huge change in the world of computing has been the aggressive rise of Mr. Gates' organization. I suppose we can thank him for contributing toward making computers into relatively cheap items. The number of machines I possess has grown from a single Sun to include a PC running BSD/OS and several machines each running one of Microsoft's systems; I run Windows NT 4.0, Windows 95 and Windows 3.1. All of these machines are on my LAN, and all can access files using some form of networked file system. I've used NFS for the Windows 3.1 system for some time, and always heap huge quantities of praise on Samba, which provides remote file access for the remaining Windows releases. Andrew Tridgell and the rest of the team have generated a UNIX product that makes the Microsoft systems usable. I keep no file of value on any of the Microsoft systems, so I don't need to back them up.

What I desperately need is a virtual file system implementation for my Sun that uses the Samba library to access files on the various Windows systems from my Sun. There's something that does this for Linux, but it isn't particularly portable into the Sun environment (on a cursory glance anyway).



I run Microsoft systems because they support desktop applications that I cannot obtain for UNIX systems. I have a need to be able to deal with Microsoft Word format documents. (Whatever that may be. Microsoft is great at changing formats with every release, so that documents quickly become incompatible.) I have a need to edit graphical images and, basically, there are simply no UNIX tools with the appropriate power at a reasonable cost. To be fair, I haven't looked for such a product, because I assume that if it exists, then it will be more expensive than the equivalent Windows product and will not offer the same functionality. This assumption could be completely wrong, I suppose, but the market has conditioned me to think this way.

I am not alone. As was pointed out to me in some recent email, and rather triumphantly I thought, Microsoft has won the desktop battle. The reasons that UNIX has "lost" are complex. I think that it was basically a failure to deliver point-and-click technology fast enough, a failure to deliver cheap applications at the right time, a failure of UNIX companies to cooperate to battle a growing monopoly, a failure of application developer companies to price software reasonably for the UNIX market, and a failure to provide a single technology base that could be picked up

UNIX Basics

by application developers and used to create a large application market. A failure.

Microsoft has generated a machine culture where every user is on their own; where every single machine is set up differently, making support a nightmare; where there is no adequate support for software; where work is lost; where systems crash; where programs that work fine today will fail tomorrow because of some apparently unrelated change to the system; where programs are created to be just adequate; and where simple tasks are simple, but complex tasks are accomplished by people doing simple tasks again and again because there is little or no automation easily available to users.

And the people purchasing systems have bought it, because it all looks so easy. They were sold the idea that you could get some cheap software that would apparently do some simple job, but they were not told that it would cost them significant sums in human time to get around the problems that the software would induce.

So, the UNIX world's failure is not only a failure to deliver applications, but a failure to communicate to the person buying the system that reliability, predictability, automation, good quality support and robust hardware are worth more than just a little. It's also possible to argue that the companies involved didn't provide any of these qualities, but I am not so sure.

The world is now locked into Mr. Gates' products, and his monopoly is demonstrably a bad thing. It's bad that his

company doesn't need to react to user pressure to improve things, to make reliable products that don't crash, to enable systems to be predictable and to make the system work for the user rather than the user working for the system.

Finally

Well my ramblings seem to have reached the appointed size, and I should stop. I hope that you will join with me in wishing all those involved with this magazine, contributors and staff, a Happy Hundredth Birthday. I'd also like to take the opportunity to thank everyone who has emailed me a comment, good or bad, or sent in some questions, dumb or hard. Getting a response from readers matters, and questions help me develop article topics.

For your information, you can find `exmh` on <http://sunscript.sun.com/exmh/>, and `Samba` is at <http://samba.canberra.edu.au/pub/samba>. The USENIX Web site is <http://www.usenix.org>. Finally, the latest version of `ssh` is available for testing, see <http://www.cs.hut.fi/ssh>. ✍

Peter Collinson runs his own UNIX consultancy, dedicated to earning enough money to allow him to pursue his own interests: doing whatever, whenever, wherever... He writes, teaches, consults and programs using Solaris running on a SPARCstation 2. Email: pc@cpg.com.
