news@UK

The Newsletter of UKUUG, the UK's Unix and Open Systems Users Group
Published electronically at `http://www.ukuug.org/newsletter/`

Volume 16, Number 1 **ISSN 0965-9412** **March 2007**

Contents

News from the Secretariat

Jane Morrison

Thank you to everyone who has kindly sent in their subscription payments so promptly. We have received a good number of early payments. Those remaining outstanding will be followed up this month. Those who have not paid by the end of April will not receive the next issue (June) Newsletter.

The UKUUG Spring conference took place in Manchester between the 19th and 21st March. The event was very popular with the tutorial and conference fully subscribed.

We are very grateful to the event sponsors, Transitive, Sun Microsystems, Novell and mod3. Google very generously sponsored the Conference Dinner.

The Linux 2007 Conference will be held in Cambridge this year at the University Arms Hotel from 2nd to 4th September.

A web form for potential speakers' submissions is available at
`http://www.ukuug.org/events/linux07/`

As in previous years we are hoping to be able to achieve sponsorship for the event which will allow us to keep delegate fees at a minimum. If you know of any company who may be interested in sponsorship please let me know.

The next Newsletter will be the June issue and the copy date is Friday 18th May.

You can contact us with comments about UKUUG's activities, or about the contents of this Newsletter at:
`newsletter@ukuug.org`

Chairman's Report

Alain Williams

I write this shortly after the successful Manchester conference. This had 'virtualisation' as its major theme and was over-subscribed, as was the tutorial on Kerberos. Many thanks go to Sam Smith for the hard work that he put in to it.

Our Linux conference is our next major event and will be in Cambridge, 2 to 4 September, and will be held just before the Linux Kernel Conference – so we may get a few extras dropping in on us.

I have been considering the idea of arranging more conferences in the form of focussed one day events designed to help our members solve real live issues. Initial topics may be: ZFS; making web servers run faster; backup in the data center. For this to be successful we will need to know what your problems are today, please let me know by writing to
`ukuug@ukuug.org`

UKUUG Liaison report

Sunil Das

Yesterday was Christmas and tomorrow is Easter – how the time has flown by. January started with an initiative concerning membership growth. The UKUUG Chair, Liaison Officer and Secretariat met with a Consultant to explore ways to promote UKUUG, the resulting report being presented to Council in February.

Round-table meetings have been arranged, in particular, with Novell at their Bracknell site. Ways in which UKUUG can give greater visibility to Novell's Gold Sponsoring Membership were explored. Consequently, literature distributions at each other's events took place in March with UKUUG materials being put out at the seminars promoting SUSE Linux Enterprise 10, the next-generation Linux platform from Novell. Also, Novell participated in the exhibition at UKUUG's Spring Conference in Manchester.

Other actions were put in the pipeline for the coming months.

A meeting with Oracle whilst visiting Dublin resulted in a promise of a newsletter article and technical submissions to the UKUUG September conference in Cambridge. Items for this newsletter have been actively sought, resulting in the articles about shell programming and IT4Communities, and there is another "letter from Toronto" by Peter Salus.

Exploring reciprocal relationships with other groups resulted in exchanging email with Ellie Young, the Executive Director of Usenix.

So don't forget to help in the liaison activity with suggestions and ideas by email or telephone. Initial contact can be made using

`sunil.das@ukuug.org`

OpenSolaris Starter Kits

Pete Dennis

The OpenSolaris Starter kit contains the Solaris Express (Community Edition), LiveCD images for Nexenta, Belenix and Schillix. The source for OpenSolaris. It is possible to order a DVD set via the OpenSolaris web page and hit the link "Request a DVD".

If you are not a registered user then you will need to complete the simple registration page which takes less than a minute; once the form is saved an email will be sent to your supplied email address to complete the registration.

It is possible to download various sources from the website as well as the SunStudio compiler to perform builds.

`http://www.opensolaris.org/`

Ruby on the Rails: Oxford, 26th April

Jonathan Conway will speak about Ruby on Rails, the well-known Open Source web framework.

This is a combined BCS Open Source Specialist Group (OSSG) and BCS Oxfordshire event.

The talk will take place at the Computer Lab, Wolfson Building, Parks Road, Oxford OX1 3QG on Thursday 26th April from 18:30 to 21:00.

For further details, please see:

`http://ossg.bcs.org/2006/09/28/ruby-on-the-rails/`

`http://www.oxon.bcs.org/program2006-7.htm#Apr`

NLUUG Virtualization event

We have received an announcement of a virtualisation conference being held by NLLUG on May 10th: further details are at:
`http://www.nluug.nl/events/vj07/`

LISA '07: Call for Papers

We have received a Call for Papers for LISA '07, the 21st Large Installation System Administration Conference which will be held between November 11-16, 2007 in Dallas, Texas, USA.

The deadline for extended abstract and paper submissions is May 14, 2007.

The Call for Papers and other details of the conference can be found at:
`http://www.usenix.org/lisa07/`

UKUUG Spring Conference: Manchester

Paul Waring and Roger Whittaker

This year's Spring Conference took place at Manchester Conference Centre on the 20th and 21st March, preceded by a tutorial on the afternoon of the 19th.

Both the tutorial (on Kerberos, delivered by Simon Wilkinson of the University of Edinburgh) and the conference itself were full to capacity, with an audience of more than 50 for the tutorial and more than 120 for the conference.

The Kerberos tutorial assumed no particular prior knowledge of Kerberos, but Simon went through the basic concepts fairly quickly at the start. However, for those who had little prior knowledge, it provided a good background understanding, while for those with more experience of Kerboeros, the tutorial was both a refresher and an opportunity to gain a deeper understanding.

Simon delivered the session in a very clear and confident manner: he provided printed notes which went well beyond the material actually delivered, and most of those present seemed very impressed by the session.

The conference proper was mostly divided into two streams: one sticking closely to the main theme of virtualisation, and the other covering some more general topics in system administration. Before spitting into two streams, there were plenary sessions on Tuesday morning.

The first talk was by James Youngman of Google. His title was *"That couldn't happen to us ... could it?"* and he looked at some of the unintended consequences of the policies and procedures for massive automation of system administration that are necessary to deal with such a vast infrastructure as that at Google. The session was both amusing and thought-provoking, and gave some insight into Google's methods (though not too much: James refused to go into detail in response to more than one question from the floor at the end of the session).

Next, Ian Pratt of Xensource gave a general overview of the Xen hypervisor and its history, mode of operation and current status. He compared Xen with competing forms of virtualisation and offered a very upbeat assessment of its future.

The next talk was by Alex Bennee and Mark Curtis of Transitive, discussing and demonstrating Transitive's on-the-fly binary translation tools. Transitive are best known for having provided the PPC to x86_64 binary translation tool used by Apple under the name Rosetta. However, this talk and demonstration was largely concerned with Transitive's Quick Transit software, which allows the running of Solaris Sparc applications on 64-bit Linux on commodity hardware.

The ability to do this depends on the use of the openSolaris libraries and userland: the `runsol` command takes you into a Solaris chroot environment where all binaries run through the binary translation software.

After lunch upstairs, the conference split into two branches. The larger of these was the virtualisation track. Owen le Blanc of Manchester Computing discussed the use which his department has made of Xen virtualisation, particularly for replacing web servers running on single machines with new hardware running multiple Xen instances on Debian.

Ian Pratt of Xensource then followed up his morning session with a deeper and more detailed look at Xen, with some discussion of internals, use cases and the road map for future development.

Bret Giddings of the University of Essex then discussed his experiences with VMWare as a virtualisation platform, with a fairly large installation of VMWare ESX Server combined with a Dell/EMC SAN storage solution.

The day's sessions ended with a presentataion on FreeBSD jails by Poul Henning-Kamp. In the FreeBSD jail concept, a number of separate chroot environments exist, each with its own superuser and its own set of installed software and libraries. These jails are effectively (almost) isolated from each other. Poul described the work that had been done to adapt the chroot call to create the jail concept: the number of lines of code that needed to be rewritten or added was remarkably small.

In the other stream, some novel methods of administrating small and large systems were introduced. Julian Simpson of Thoughtworks offered a talk on Agile Sytem Administration, in which the Agile programming concept is applied to administration. Paul Anderson of Edinburgh University gave a talk on Mass System Configuration. Of particular interest in the afternoon was the talk on Perl 6 by Jonathan Worthington. Some of the new language features were discussed, some of which (e.g. grammars) promise to make life easier in certain projects. As it is we will all have to wait at least another year before getting to deploy these new features, but by the sounds of this talk we have some major improvements to look forward to.

At the end of the day's presentations, there was a short PGP key signing session before we moved off to Old Trafford for the conference dinner (generously sponsored by Google). Despite one of the coaches deciding to go a different way on two occasions, we managed to get everyone there and spent the next two hours or so socialising and enjoying a four course dinner in one of the conference suites. A competition to find the most amusing possible use of virtualisation and a prize draw to win a Google laptop bag rounded off the proceedings although I believe several people carried on the socialising in a pub afterwards.

The next morning brought us two talks on Solaris and a presentation on VMware servers, followed up by clustering management and filesystems and yet more Perl 6 (this time focusing on the implications for system administrators). As with most conferences, things quietened down a little bit on the last day, but there was still a good atmosphere of discussion and debate which carried through to the next session of talks on BSD and secure filesystems. The conference was rounded off by a session of 'lightning talks', in which several delegates gave presentations ranging from thirty seconds to ten minutes on a variety of topics, all of which were informative and in some cases highly amusing as well. Top marks go to Jonathan Worthington for being brave enough to extol a Windows 2003 product (PowerShell) to a room of Unix users.

Overall, the one thing that was noticable above anything else in the course of the conference was that no one seemed to be sitting on their own for any length of time. Given that IT and computing people have a bit of a reputation for being quiet and unsociable (at least that's the impression one always gets when telling people what industry one works in), it was good to see this stereotype didn't hold good at the conference. Everyone seemed to engage and network with each other and at the end of the day, in my opinion at least, that is what conferences are really about.

Conference slides are now available on the conference web site:
`http://www.ukuug.org/events/spring2007/programme/`

Shell scripting: the Cinderella language?

Clive Dark

One of the most neglected areas of programming is shell scripting. Much has been written about the technique of software development, but the lessons learnt in this area are sometimes ignored when it come to shell scripting. Shell scripts are often written by people without a formal programming background, and looked on with derision by 'real' programmers. Yet shell scripts can be mission critical, and, when badly written, have a severe impact on system performance.

The name "shell" was given because the program was meant to be no more than a wrapper around the real application. The early Bourne and C-shells were not capable of much more than that, but the Korn shell, particularly ksh93, and more recently Bash, has steadily added more features and functionality so that the language can carry out its chief function, and many more, with greater efficiency and elegance. It is to be regretted that many script authors never advanced their knowledge beyond the level of PC-DOS `.bat` files.

The shell has substantially been the tool of System Administrators, Testers, Integrators, and so on – those whose main function traditionally did not include programming. Today's System Administrators cannot get away without doing some coding, even with the growth of GUI driven aids. Yet they are rarely trained in good scripting techniques, or even programming basics. They learn by copying others' code, which is a bit like learning to drive by imitating how everyone else does it. You might be lucky and pick a role model who is genuinely skilled, but more likely you won't. "Who wrote this?" is often answered: "our script guru". Nothing in computing better deserves the phrase "In the land of the blind, the one-eyed man is king".

Performance issues are commonly caused by the unnecessary use of external programs. There are a number of reasons for this:

- Not realising that many "UNIX commands" – a misleading phrase if ever there was one – are not built-ins (a built-in is where functionality is within the shell program itself)
- Lack of shell syntax knowledge to avoid external programs
- Failure to understand the overhead involved in process creation

That final point crops up repeatedly. The argument is often given that UNIX utility programs, such as `ls`(1), `cat`(1), `sed`(1) are mature, highly efficient, and optimised such that they perform their actions very efficiently. This is undeniably true, so much so that they often complete their task in a shorter time than it took to create the process address space that they run in. The argument against this stance is simple – run an `strace`(1) or `truss`(1) timing summary to include child processes and demonstrate that the script spent more time executing forks than anything else.

The shell authors are aware of the problem, and both Bash and ksh93 have a large number of built-ins. Recent versions of Korn shell 93 in particular use a neat trick to build-in utilities such as `wc`(1) and `cat`(1) when loaded from `/bin`. Ksh93 also allows user-written built-ins, and has a "compilation" feature (shcomp) where scripts are dumped to a file in an intermediate compiled state for later execution.

Shell languages have matured to such a degree that it could be argued that `sed`(1) and `awk`(1) have become obsolete. For example, both Bash 3 and ksh93 support Extended Regular Expressions (ksh93 supports Basic REs as well), and shell pattern matching has been around for some time. Both these shells support textual substitution, yet still we see variables being echoed into `sed`(1) for that purpose. Bash and Korn shell can split fields (using IFS) and read a file record direct into an array, yet still `awk`(1) is routinely used to extract a single field from a record.

Restrictions on the size of arrays, the number of open file descriptors and so on, have largely been removed in recent versions.

The latest incarnation of ksh93, version ‘s’, has the capability of a file seek to not only a byte offset, but even a pattern (expressed as a shell pattern, an ERE, or a BRE). It has associative arrays, and variables not unlike a C struct (called “compound variables”). It also has a feature, reminiscent of the Perl ‘tie’ command, which enables functions to be called automagically when variables are read, changed, or removed. In fact the Korn shell now sees itself as a rival to Perl. That might be stretching things a little (OK, a lot), but it does give some indication of how far these languages have come.

Yet these huge improvements in shells have gone largely unnoticed by the UNIX community, which is in stark contrast to similar changes in other programming regimes.

Korn shell 93 is now open source, if you have not looked at it lately then visit the website at **`http://www.kornshell.com/`**.

It can be downloaded in a variety of binary formats, as well as source. Beware that there are incompatibilities with ksh88.

So, if you are a programmer used to other languages: don’t dumb-down just because these are shell scripts – it can be more challenging to write a good program in Korn shell that in a language with OO, templates, class libraries, and a nice IDE. See scripting as a challenge to your ability to write elegant code.

If you write shell scripts and programming is not your primary function, then consider that maybe you are a programmer now, and it is time to hone those skills.

Whoever you are, if you write or maintain shell scripts, learn the language, and please treat it with respect: it is capable of a lot more than you think.

Open Source in the UK Voluntary Sector

John Davies

I run iT4Communities, a service which introduces IT professionals wanting to volunteer their skills to charities and other voluntary organisations needing IT help. I have been interested in open source since Unix was a toddler but have never become an expert. I was even at the EUUG meeting where Rob Pike announced the Bell Labs Blit (remember that?).

iT4Communities is a charitable programme of the Information Technologists Company, a City of London Livery Company. The Company has always encouraged its own members to contribute their own skills pro bono and iT4C aims to extend this to the whole IT profession.

`http://www.wcit.org.uk/`

In 2004, a survey of IT in the UK Voluntary Sector showed only 14% of respondents had heard of open source and fewer than 2% were using it.

`http://www.icthub.org.uk/export/sites/icthub/research/Baseline_research.pdf`

The same researchers, Paul Ticher and Andrea Eaves have recently found that the sector has woken up to open source. 58% have now heard of open source and 11% report using it.

The availability of information about using open source has grown with the support of the ICT Hub. For example, the OpenITup initiative, driven by Matthew Edmondson at NCC has built a virtual learning environment, a creative space for the development and delivery training in free and open source software (FOSS) for sector technical people. The LASA / ICT Hub Knowledgebase for the sector shows 84 papers referencing open source and 40 referencing Linux. There has also been much discussion of the ethics of the sector and how these align with those of open source – or not – notably on the UK Circuit Riders list. However, there is already evidence – whatever the ethics – that this important sector of the UK economy is beginning to move towards open source.

iT4Communities uses a volunteer matching system based on Java and Tomcat with a MySQL database. This replaced a .ASP system in May 2006. iT4Communities retains IPR in the developed system and is aiming for a fully open source system which is developer-neutral. We are now well advanced with a major upgrade of the system using a second developer company. The lessons from this development model will, we hope, help other small organisations in and out of the voluntary sector to develop open source solutions without either the deadly embrace of a single supplier or the necessity to have their own, in house, developers.

iT4Communities volunteers already include a high proportion of open source and Linux expertise but the sector need for these skills is growing. If you have some time available and you would like to help a UK voluntary sector organisation then get in touch with me or one of my colleagues – or simply register as a volunteer through the web site. The open source community is especially welcome as a critical friend to iT4C – if you like what we do, tell your friends, even if you don't tell us!

For further details see:
`http://www.it4communities.org.uk/`

References:

`http://www.icthub.org.uk/`

`http://www.openitup.org/`

`http://www.icthubknowledgebase.org.uk/`

`http://lists.lasa.org.uk/lists/info/ukriders/`

Contacts:
`john@wcit.org.uk`
`info@it4communities.org.uk`
020 7796 2144

The Microsoft Perspective

Kel Vanderlip

"Nick McGrath, Director of Platform Strategy, Microsoft Ltd will speak about the Microsoft perspective on Open Source for the BCS Open Source Specialist Group (OSSG) on Monday 5th March 2007."

It pays to visit the UKUUG web site from time to time. For instance, last month I noticed that the British Computing Society had invited UKUUG members to a talk by Nick McGrath, Microsoft's Director of Platform Strategy. Some of you might remember Nick from last years Linux World event at Olympia. He sat on a panel of otherwise open-source advocates and defended Microsoft's corporate philosophy on the benefits of closed-source to developers and users. It was a bravo performance, but he didn't win the crowd.

About 20 people attended the BCS event, including at least 2 from the UKUUG. Nick was in great form, and he launched into a review of the Microsoft coding process, its benefits to everyone, and how good a place to be if you are a developer. Apparently the turnover is low (94% of the staff are long term) and the pay is high. He covered the product life cycle (no surprises) and emphasised the importance of security in new code.

Nick talked about Microsoft's open-source lab, which runs dozens of distros, and its staff which include open-source developers. He pointed us to a Wiki for the lab staff at
`http://port25.technet.com/`

Things were a little less clear on the Novell/Microsoft alliance, and its possible dangers to the open source community. Apparently we'll see a lot of new code aimed at easing the integration of Linux and Microsoft in the data centre. Nick announced a new Novell (Microsoft) OpenDocument to Open Doc converter at the meeting. I forgot to ask whether this code will be available or will help open-source developers.

At question time I mentioned to Nick that I found myself unable to support Microsoft products because I could not understand and fix bugs without access to sources. His answer was straightforward: it will never be Microsoft's policy to release source code. They make a fortune doing things the way they do, so they'll not change. The high profitability of Microsoft tells them that they are doing things right. So, if I want code-level support I can sign on to a support scheme, etc. etc. I was not encouraged.

The finale of the talk was great. It was Nick's 15th anniversary at Microsoft, and to celebrate he gave us each a copy ot Vista Ultimate (a real one) and a 6 month timeout release of Office 2007. I can safely state that without this encouragement I might not have had the Vista experience for some time. Nick made us all promise not to sell the shrink-wrap packages on eBay.

I ran home and stayed up all night having the Vista experience. I am now a recovering Vista user.

Office Document Format Standardisation: an update

Mike Banahan

Office Open XML – sounds like something to do with OpenOffice.org, doesn't it? In fact it's not, it's the name given to some 6,000 pages of draft standard being proposed by ECMA (the European Computer Manufacturers Association), though in reality most people seem to think that it's Microsoft's document with an ECMA rubber-stamp on it. It's supposedly the format that will or can be used by later versions of the Microsoft office suite and represents quite a change away from the closed binary formats of earlier times.

ECMA is entitled to submit documents to ISO for approval as full international standards via a 'fast-track' process which removes some of the extremely laborious work that tends to be done on non fast-track submissions. Having been involved in the C and C++ standards work I can vouch for the extremely fine scrutiny that is typically done on those submissions and the years of work before a final draft standard is submitted to the ballot process.

ODF is already a standard in this area. It was itself was fast-tracked and is generally, though incorrectly, considered to be the OpenOffice.org document format. It may not be perfect, but it is gaining implementations and is, of course, the default format for OpenOffice.org.

The fact that the ECMA document is seen as essentially Microsoft's own standard albeit in an ECMA suit is extremely alarming to people who are predisposed to distrust Microsoft. Unsurprisingly there is quite a lot of resistance to the idea of ISO permitting this to become a standard and there are various arguments put forward to support that view. Principal amongst them are, I think it's fair to boil them down to:

- we don't need two standards in this area. ODF is fine, thanks

- the proposed standard is rubbish
- the commercial and IPR terms associated with the proposed standard can't be trusted
- lots more besides

There's no point in me repeating the fine detail of all of the arguments which are extremely well documented at Grokdoc and at the time of writing the Wikipedia page seems to give an overview of the situation without being blatantly one-sided. It's rare to find anyone interested in this situation who isn't already highly polarised.

The fast-track process at ISO allows 90 days for National Bodies such as the BSI to digest the basics of the fast-track proposal. They then vote on whether or not a 'contradiction' exists. Following that, there is a contradiction resolution procedure followed by a five-month ballot period. If at the end the ballot says 'yes' the proposal gets blessed as an international standard.

This matters a LOT. Increasing numbers of public bodies are making noises about switching to, or only approving of the use of, international standards for document interchange. There's a strong argument that the proprietary Microsoft '`.doc/.xls/.ppt`' formats are a huge barrier to breaking the semi-monopoly of desktop software that that company evidently enjoys. A free, open and high-quality standard would be extremely important for anyone wishing to compete head-on.

Equally, it could only be seen as a huge win for vested interests if the market can be 'split' in some way with two competing standards being approved.

On behalf of the UKUUG I have joined the BSI's committee looking into XML document formats, the committee that's scrutinising the ECMA proposal. Based on that committee's advice, the BSI submitted a response to ISO which, though confidential, has been more or less published elsewhere and counted as a 'contradiction'. Overall, the majority of the countries that responded did so negatively.

It came as some surprise to learn that despite the reponses received, the ISO management have decided to let the fast-track process continue and as I write, the five-month balloting process is apparently under way. Nobody can predict the outcome of this.

Representing the UKUUG on the BSI committee, it's my intention to continue to argue that it would be a very bad thing for there to be two standards in this area. All that that will do is lead to confusion and cost – the industry simply doesn't need it. If other UKUUG members wish to get involved in the process, the BSI committee welcomes others to join it and I would be glad to see you there!

Licensing, Copyright and Patents

Peter H Salus

Over the past two decades, issues concerning software licensing/patenting have become increasingly important. As the entire SCO matter (theoretically, at least) revolves about the "ownership" and use of code, I will put that aside for now to take up a bit of history where copyright and patents are concerned.

Copyright is a form of license – a permission to do something. The first copyright law was the Statute of Anne, usually cited as 1709. (As with almost everything, the date is ambiguous: the Statute was introduced in 1709, but "passed into law" on 10 April 1710. The actual publication of the Act [BL 8 Anne c. 19, 261] (British Library) is just headed "Anno Octavo". A facsimile can be seen at:
`http://www.copyrighthistory.com/anne.html`

But, as the eighth year of Anne's reign terminated on 17 March 1710, contemporaries must have thought of the statute as dating from 1709. On the other hand, Adam Budd [*TLS* 15 July 2005, p. 24] calls it "The Copyright Act of 1710.")

At any rate, the Statute required booksellers to record their titles in the Stationer's Register to gain protection for their "literary property." All registrations stem from this. [Anyone interested in the history should read Ronan Deazley's *On the Origin of the Right to Copy* (Oxford, 2004).]

Patents are specified by Article 1, Section 8 of the US Constitution (which took effect 4 March 1789): "Congress shall have Power ... To promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries." The main body of the Law concerning Patents is Title 35 of the US Code:

`http://www.access.gpo.gov/uscode/title35/title35.html`

"Limited Times" has been interpreted in a variety of ways. As early as 1729, when John Gay sought an injunction against three unauthorized versions of *Polly*, his 1728 opera, the Lord Chancellor Charles Talbot granted Gay's injunction "in perpetuity." More recently, US Congress has extended Disney's rights in Mickey Mouse to 2019 [*PL* 105-298; 27 October 1998] This is the so-called Sonny Bono Act:

`http://thomas.loc.gov/cgi-bin/bdquery/z?d105:s.0505:`

Extending terms has gone hand-in-hand with extension of what is covered. "Literary property" was extended to include music and other creative material. Patents, which for many decades required physical instantiation, were extended to concepts, methods and software. (The US Patent Office accepted an application for a software patent for the first time in 1981.)

Sunbeam Products, for example, sued the West Bend Company over the configuration of the "stand mixer". In September 1997, the Fifth Circuit agreed that the configuration might lead to "consumer confusion" and held for Sunbeam. In Fall 1997, Amazon.com filed for a patent on one-click purchasing, which was granted on 28 September 1998.

It was a decision by the Court of Appeals for the Federal Circuit which significantly broadened the scope of patentable material. The decision was issued in July 1998 and the Supreme Court declined to review it the next year, making it a national precedent. The case is known as the State Street Bank decision. A concise summary is at:

`http://library.findlaw.com/1999/Mar/1/128488.html`

Prior to this, it was understood that business methods couldn't be patented. Software had been patentable, but it wasn't an easy process. State Street Bank and Trust Co. v Signature Financial Group removed the barrier to patents for business models as well as those hampering patents for software. Amazon's patent was a result of the case, a year before the appeal. The 2003 eSpeed patent-infringement suit filed against BrokerTec, alleging patent infringement on electronic-trading systems and methods, is another instance. (As late as 14 December 2004, Mr. H. Jones, Deputy Director acting for the Comptroller-General of Patents [UK], rejected two patent applications that were "only for business methods".)

However, there is one important point to be drawn from this: all the case law concerns business methods. Thus far, it is copyright, not patent law which protects operating systems, editors, data bases, etc.

Trade Secrets

A third form of intellectual property protection is the trade secret. However, there is no agency that certifies something as a trade secret and it is difficult to identify one after it has been compromised. The prime determinant appears to be the extent to which the information is known outside the company.

The language by which a trade secret is defined varies, but there are three factors that are common to all such definitions: a trade secret is some sort of information that (a) is not generally known to the relevant portion of the public, (b) confers some sort of economic benefit on its holder, and (c) is the subject of reasonable efforts to maintain its secrecy.

UNIX

Effectively, all contemporary operating systems grew out of Corbato's CTSS (Compatible Time-sharing System - 1961). UNIX is an operating system that grew out of the Multics project, a direct descendant of CTSS. See:

`http://www.multicians.org/history.html`

`http://www.multicians.org/multics.html`

From 1969 till 1973, UNIX was confined to AT&T/Bell Labs internal use. After October 1973, academic and research sites were given copies on disk or on tape. (For a full history of this, see Salus, *A Quarter Century of UNIX* [1994].)

To all effects and purposes, software licensing began when UNIX was ported to the Interdata 7 (in Australia) and the Interdata 8 (at Bell Telephone Laboratories (BTL)).

Prior to that, an OS ran only on the machine with which it was sold or, in the case of UNIX, on the Digital Equipment Corporation (DEC) PDP-11. Though the Australian port was at an academic institution, Bell Labs now began receiving requests from commercial firms. Furthermore, at various academic institutions, UNIX code was developed and altered and new applications were written. Most of these made their way (still primarily via tape) to New Jersey, where they were incorporated into subsequent versions of UNIX. (See: Marshall Kirk McKusick's *Twenty Years of Berkeley Unix* [1999] in "*Open Sources: Voices from the Open Source Revolution*").

Until 1979, nearly all versions of UNIX were within the public domain: BSD, 2BSD, 3BSD; First Edition through Seventh Edition; System III; 32V; etc. After 1979/1980, AT&T's versions of UNIX required paid-up licenses. (Andy Tanenbaum has cited this as the impetus for his writing Minix.) Beginning in the mid-1980s, there was an active attempt at making Berkeley UNIX free of AT&T code.

In 1974, Robert Greenblatt at the MIT AI Lab began the Lisp machine project. His first machine was called CONS (1975). This was improved into a version called CADR (1977). CADR was the direct ancestor of both Lisp Machines, Inc., of which Greenblatt was the founder and president, and of Symbolics, Inc. And Symbolics, in several ways, forced Richard Stallman to form the Free Software Foundation and the GNU Project.

Richard M. Stallman, though a freshman at Harvard, began working for Russ Noftsker at the MIT Artificial Intelligence Lab in 1971. He said:

I became part of a software-sharing community that had existed for many years. Sharing of software was not limited to our particular community; it is as old as computers, just as sharing of recipes is as old as cooking. But we did it more than most. The AI Lab used a time-sharing operating system called ITS (the Incompatible Timesharing System) that the Lab's staff hackers had designed and written in assembler language for the Digital PDP 10 ... As a member of this community, an AI Lab staff system hacker, my job was to improve this system. We did not call our software "free software", because that term did not yet exist, but that is what it was. Whenever people from another university or a company wanted to port and use a program, we gladly let them. If you saw someone using an unfamiliar and interesting program, you could always ask to see the source code, so that you could read it, change it, or cannibalize parts of it to make a new program.

Less than a decade later, everything changed for the worse. rms told me:

It was Symbolics that destroyed the community of the AI Lab. Those guys no longer came to the Lab. In 1980 I spent three or four months at Stanford and when I got back [to Tech Square], the guys were gone. The place was dead.

(Sam Williams says that Symbolics hired 14 AI Lab staff as part-time *consultants.* Richard was truly the last of the hackers.) Symbolics, further, din't recycle its code back to the community.

rms continued:

In January '82 they [Symbolics] came out with a first edition. They didn't share. So I implemented a quite different set of features and rewrote about half of the code. That was in February. In March, on my birthday [March 16], war broke out. Everyone at MIT chose a side: use Symbolics' stuff, but not return source for development. I was really unhappy. The community had been destroyed. Now the whole attitude was changing.

At this point, there were many sources for the "methods and concepts" employed by the UNIX authors. There was the *Commentary on 6th Edition UNIX* by John Lions(1977); two volumes of the *Bell System Technical Journal* (July-August 1978 and October 1984); Steve Bourne's *The UNIX System* (1983); AT&T Pacific's *UNIX System Software Readings* (1986); works on programming by Kernighan and Plauger, Kernighan and Pike, and Jon Bentley; works on the AT&T System by Bach and on the BSD system by McKusick et al.; Vahalia's *UNIX internals*; and Mike Gancarz' *The UNIX Philosophy* (1994).

There might be unrevealed code (under copyright), but it is difficult to imagine any prior to AT&T V7. And it is equally hard to imagine unrevealed ”methods and concepts.”

The GPL has grown up into a worldwide community. The basic notions that drove Stallman have migrated to the BSD communities and have been part of Linux from its very beginning.

Licensing exclusive, proprietary code makes about as much sense as licensing the screwdriver.

Information Architecture for the World Wide Web

Louis Rosenfeld and Peter Morville

O'Reilly Media
ISBN 0-596-52734-9
504pp.
£ 28.50
Published: 5th December 2006

reviewed by Lindsay Marshall

I know that I have a habit of writing pithy (OK, short) reviews, but I think this one may turn out to be very pithy indeed.

How do you approach reviewing what is the standard text for a subject? It really doesn't matter what any particular individual thinks about it, the informed reader still has to know about the material it covers, warts and all. And, if the authors can keep up, the book need never go out of date. Thus the third edition of this book, taking in all the new trends in social tagging and such things, and there wasn't anything available that was as comprehensive as the second edition, so the authors are well ahead of the game. (Which since they pretty well invented the whole area as a discipline is fair enough.)

In fact, I'm not even going to attempt to describe the relevant examples, the excellent rules of thumb, and all the other goodness that is in the book. If you are involved in any way with building information delivery systems then you have to be familiar with this book. It would make an excellent textbook for a module in Information Architecture too – its structure is, as you would expect, exactly right. Just go out and buy it, OK?

Learning JavaScript

Shelley Powers

O'Reilly Media
ISBN 0-596-52746-2
335pp.
£ 20.99
Published: 3rd November 2006

reviewed by Lindsay Marshall

JavaScript is, of course, the new black and everyone and their cat are rushing to bring out new or updated texts, all kitted out with, at the very least, shiny Ajax and DOM sections. And I have to say, up front, that I really think that the only JavaScript book you really need is the O'Reilly JavaScript (The Definitive Guide) – if you write JS or are going to write JS, you must own it. Introductory books are all well and good, but most of them just take up shelf space after you have been through them, because they are, by their very nature elementary. This is especially true for experienced programmers who ought to be able to pick up a new language from the resources available on the net without much effort. That being said, I found the book under review to be a rather good example of the genre – I even found a brief mention of a language feature that I don't recall meeting before.

The introductory material is coherent and has good coverage, though it is at times a little dry. Beginners might find that the level of difficulty ramps up pretty quickly though, getting on to advanced ideas such as using Flash for persistent storage. The last few chapters of the book look at DHTML, Ajax, and the use of libraries like Prototype and Script.aculo.us. This all useful stuff and well presented, though the material on the packages will age quite quickly as new releases and features seem to come at frequent intervals. There is also a short discussion of the future of JavaScript and what is being talked about for the 2.0 version, which, without going into details, gives a good sense of the direction in which things are moving. One omission does seem to be any mention of the unobtrusive JavaScript movement, but that is only a minor grumble.

If you are setting out to learn JavaScript, have some programming experience and actually want a good introductory text, then I would definitely recommend this one. Lots of examples and good, clear explanations of features. I'd still advise you to buy the reference book though.

MySQL Cookbook

Paul Dubois

O'Reilly Media
ISBN 0-596-52708-X
948pp.
£ 35.50
Published: 5th December 2006

reviewed by Lindsay Marshall

Forget Delia or one of the Grigsons, this is Mrs Beeton. It's huge – Harry Potter size (though well written and with a much better story), in fact it weighs 1.565Kg (55.2oz) according to my kitchen scales. There is some "egg boiling" material in here such as the material on connecting to databases, which is covered for each of Perl, Ruby, Python, PHP and Java, but this is not really the reason for the size. The real reason is that each "recipe" is detailed and illustrated by examples of queries and outputs, which just take up a lot of space – all that verbose SQL and tabular output.

The recipes cover everything from string and date handling, through JOINs up to using stored procedures. And even the ones that seem to be quite basic often contain a useful tip or some wrinkle that you might not have known about. I found the section on finding and eliminating duplicates most useful as that is something I always struggle with when I have to do it. I was going to say that I found the web specifc material in the final chapters not particularly useful, but as I looked through them I got sucked in by some very neat ideas – this does tend to happen throughout: your eye gets caught by something and you're off.

I suspect that even SQL experts will get something from this book. People like me who are only just starting to be happy with joining a couple of tables will learn a lot, or at least will have somewhere to look for the answers to their problems. The book is quite pricey, and it is likely that equivalent material is available for free scattered throughout the net, but it is probably worth the expense just to have it all in a concentrated form for quick and easy reference. And it will certainly stop stuff from blowing around on your desk when someone opens the window.

Backup and Recovery

W Curtis Preston
O'Reilly Media
ISBN 0-596-10246-1
729pp.
£ 35.50
Published: 16th January 2007

reviewed by Mike Smith

Performing backups has come a long way – these days it's not just about those nightly incrementals, GFS (Grandfather, Father, Son) schemes and tape rotation.

It's all about meeting RTOs (Recovery Time Objectives); RPOs (Recovery Point Objectives) and addressing the wide range of problem scenarios like file deletion, data corruption, object recovery (emails for instance, or database tables), hardware failures ... all the way to total disaster recovery.

Backups now have to be performed whilst services are online; they have to be compatible with a wide range of applications; data recovery has to be rapid. So I think there's a lot more to it now. In fact the latest backup technologies now boast "continuous data protection" (CDP) – securing data literally all of the time, doing this without any application downtime, and enabling recovery to ANY point in time. So on the one hand (and certainly in the past) backups have been quite mundane and boring, but it's also turned into quite a complex and dare I say, exciting, area.

I was interested in reviewing this book precisely because this complexity. Has this latest O'Reilly title (December 2006) got to grips with all of these issues?

Reading the introduction, it turns out that this is a rewrite of an earlier book that had a different title – "Unix Backup and Recovery". The picture on the front is the same though, some sort of crocodile thing called an Indian Gavial. Slightly confusingly, in a move from the original Unix theme, the subtitle in this edition now says "Inexpensive Backup Solutions for Open Systems"... but SQL Server gets a chapter, as does MS Exchange and there are other bits and bobs on what I wouldn't call Open Systems. So I'm not sure my vision of modern backup techniques is going to be covered. Nevertheless, there are certainly still some topics that are of interest to me today – namely VMware.

Right at the beginning, on page 3, the author states that he is in the process of writing another book, and that gives me the answer I'm looking for. This book (the one I am reviewing) is about small budget backup schemes. The one he is working on is more focussed on the enterprise environment – the sorts of things I had in my introduction, plus de-duplication, replication and similar concepts. It will also cover the commercial backup products. There is a chapter on commercial tools in this book, but it's more about concepts and techniques used by commercial products; not the products themselves. In fact I didn't spot a single product name in there.

The first technical chapter covers some historic tools (that no doubt are still in use all over the place today!) - things like `tar`, `cpio`, `dump`, `dd` etc. Three chapters are dedicated to open source backup solutions: Amanda, BackupPC and Bacula. There's also one on performing CDP like functions and snapshots with rsync and other tools. There's a part (the book is divided into six parts) on Bare-Metal recovery on various platforms (Solaris, Linux, Windows, HP-UX, AIX and OS/X); and a part on database backups (Oracle, Sybase, SQL Server, Exchange, PostgreSQL and MySQL).

The final part has a chapter on performing backups in a VMware environment. Although this is useful, providing information on a few different options, it doesn't mention VMware's Consolidated Backup product. This comes as standard with VMware's Infrastructure Enterprise edition, centralising and decoupling backups from the virtual machines and I think it would have been worthwhile mentioning (unless he's saving that for his other book).

The book is filled with anecdotes and wisdom – they are good learning points. In my case I could tell you about the time a silent tape drive write error was causing data corruption on the cpio backups of an SAP system. Of course I saved the day ;-) ... but learnt a lesson the hard way (it was an all-nighter). I won't mention deleting all files on a system that were over 28 days old because in a script I didn't check the return code of a `cd` statement before the usual `find -exec rm` business. Ooops, too late. Hope you don't make the same mistake.

Overall I quite like this book. It talks about issues around the whole backup and recovery subject, not just the mechanics. The three chapters on the backup software (Amanda, BackupPC and Bacula) are brief but give you enough to familiarise yourself with what they're about and how to set them up. In a similar way the database chapters give you an overview of what issues you should consider with each type of database. I haven't mentioned that its also about 700 pages long - altogether there's a lot of material.

So for anyone charged with ensuring data protection in a standard environment with various different types of servers and products and the usual challenges this brings with it, you could do worse than buy this book. Once this next one comes out it'll be interesting to see if it's a good companion volume.

Network Security Hacks

Andrew Lockhart

O'Reilly Media
ISBN 0-596-52763-2
455pp.
£ 20.99
Published: 10th November 2006

reviewed by Raza Rizvi

A clever play on words 'Network Security Hacks' because this suitably detailed book deals not with breaking into remote systems but with protecting your own Internet (or indeed Intranet) based services. The author has added an extra 25 'hacks' to this second edition but manages to retain a book that can be picked up when a particular issue needs attention or, though the sensible groupings, when a specific area is of interest.

UNIX and then Windows host security start the collection. Whilst some of the hacks might seem obvious there are also a good many that fall into those overlooked backwaters like sandboxing and auditing. I don't mean to imply that this is some dry read, far from it, each hack is covered very adequately in 3-5 pages with plenty of links to delve into the deeper points. The text is set out following the normal O'Reilly house style and that makes it both easy and interesting to the eye.

OSX based email gets coverage in chapter 3, which deals with privacy and anonymity such as SSH, though this is the least clearly defined section of the book as many of the topics could also have fallen into a wider remit chapter 10 (secure tunnels). Mail returns in force in chapter 5 where SSL based imap/pop3, together with TLS SMTP stand shoulders with secDNS.

Network interested readers will like the altogether sensible suggestions in chapter 6 covering ARP spoofing, NTP, certificate distribution, sniffer detection, and vulnerability detection. The emphasis for (the short) chapter 7 is wireless security, and chapter 11 covers network intrusion detection with a number of excellent SNORT based examples. (Firewalls are in chapter 4 which although 10% of the book seems too short to cover the breadth of systems described).

Logging and monitoring form chapters 8 and 9. Even though it is every sysadmin's favourites of NAGIOS and RRDTool, again the descriptions are written to the right length to educate the reader without overwhelming them with detail or skimping on the important details.

The complexity of each hack is diagrammatically illustrated with a thermometer symbol, and in the end that also sums up the book. It is taking the current temperature of the tools that form the kitbag of a modern day system administrator. Like any collection of hacks, it is the opinion of the author that these will be relevant to you and I think the for the most case Andrew Lockhart has guessed close enough for most techies to be happy.

CSS Cookbook

Christopher Schmitt

O'Reilly Media
ISBN 0-596-52741-1
516pp.
£ 31.99
Published: 27th October 2006

reviewed by Bob Vickers

There are two ways of documenting software: you can either systematically describe all its features, or you can think about the tasks a user might like to perform and describe how to do them. The Cookbook series is definitely in the second camp, and all the better for that.

Software cookbooks are more ambitious than a typical food cookbook: they don't just provide recipes for you to use, they also discuss them so that you can understand them and produce new recipes of your own. A good recipe will be an elegantly written example of good practice as well as something to get the job done.

So how well does the CSS cookbook manage this? It certainly contains a lot of useful information, and it is well-organised to help you find what you want. I am a relative newcomer to CSS and have found it helpful.

But I couldn't help noticing flaws which made it fall short of being excellent. One obvious one is that the book is not in colour. This is clearly a matter of economics rather than the author's fault, but it is much easier to illustrate web design if colour is available. I was reminded of a snooker commentator in the 70's who once remarked “for those of you with black and white sets the blue ball is the one behind the pink”!

Anyone learning CSS quickly discovers its Achilles Heel: bugs in the various browser implementations mean you still have to put ugly hacks into your website to make it work on all browsers. The book has detailed information about which features work on what browsers, but I felt there was an unspoken assumption that I wanted my website to look perfect on every conceivable browser all the way back to Netscape 4. I value simplicity and elegance in my code, so I am prepared to put up with a little ugliness for a small minority as long as the content is still accessible.

Sensibly the book does not restrict itself to CSS: it occasionally mentions Javascript and Flash when the author feels they provide a better solution to the problem or a good workaround for a browser bug.

So to summarise: a useful book, but with room for improvement.

HTML and XHTML: The Definitive Guide

Chuck Musciano and Bill Kennedy

O'Reilly Media
ISBN 0-596-52732-2
654pp.
£ 35.50
Published: 27th October 2006

reviewed by Graham Lee

The first thing I do when I'm looking at any book about the web is something I learned while perusing second-hand bookshops: go to the index and look for 'gopher'. If gopher is mentioned in some kind of 'history of HTML and the web' section, that's fine; if it appears in some other context then the book is possibly out of date. This book contained a reference of the second kind, which may be surprising as it's also modern enough to discuss XHTML. My overall impression of the earlier chapters in "HTML and XHTML" is that they've survived the first five editions without much change, accreting new information but never really losing any. Deprecated and obsoleted features appear alongside supported features, and browser-specific extensions which only ever worked in Netscape 4.

Conversely the later chapters are much more succinct and relevant, probably because they represent newer developments and therefore have not seen so much revision. The chapters on Cascading Style Sheets and HTML forms in particular are very useful and thorough, describing well not only how to write CSS or forms, but also the implementation details (although an interesting choice on the part of the authors sees XForms left out due to its status as a working document, while XFrames which is in a similar state was included). A chapter on targeting mobile devices is included, which while brief is a welcome overview of the particular considerations of mobile browsing. The book is finished off with a chapter of Tips and Tricks, describing a few hacks to achieve certain layouts or styles with HTML, though without mentioning style sheets.

HTML and XHTML: The Definitive Guide certainly is definitive. The subject is dealt with exhaustively, however I did find it problematic that the various obsolete or proprietary features are included in the main chapters on HTML, as in some cases a lot of wading must be done to find out the current and standards-compliant state of play.

Programming Embedded Systems

Michael Barr and Anthony Massa

O'Reilly Media
ISBN 0-596-00983-6
301pp.
£ 35.50
Published: 20th December 2006

reviewed by Paul Waring

As a software developer who has occasionally worked with small systems, a book which promises to get me started writing embedded software sounds like it will be just up my street. Even better, the subtitle of the book suggests that all the development will be done with GNU tools, which is a good invitation for someone who just wants to dip their toes in the water without forking out thousands of pounds for expensive development kits.

The book begins with a good introduction, which explains exactly what an embedded system is and gives some examples of existing projects. Of particular interest is the explanation of why C is the language of choice for embedded systems - although I felt that a little more time could have been spent on this area. In particular I would have liked to have seen a digression on languages other than C – the authors mention that "the Ada language has many features that would simplify software development if used instead of C or C++", but give no hint as to what these features might be.

The explanation of hardware schematics was also useful, although I still struggled to fully understand some of the diagrams that were presented, so perhaps a little more time could have been spent on explaining them. Having said that, the book is really aimed at software developers and does suggest finding a hardware guru who you can ask about more complex issues.

The main chunk of the book manages to cover every major topic related to embedded systems that I could think of, including chapters dedicated to memory, peripherals and interrupts. Plenty of information on operating systems is also included, although most of the issues will already be familiar to most people with a background in computing.

The final chapter on optimisation techniques was particularly helpful, as this is a topic which is rarely covered in other programming texts because the trade off between optimisation and code readability/reuse rarely makes it worthwhile. However, in an embedded environment with significantly fewer resources it is obviously an important topic and it is good to see that it has not been overlooked.

The chapter that I was most impressed with overall though was the coverage of memory, which was particularly detailed. Not only were the main types of memory discussed and explained but there was also a significant amount of space dedicated to discussing testing and troubleshooting all common problems with memory. In addition to this, there was plenty of information about the problems of endian issues when working at this level, which is an issue that most software developers will not have experienced before.

Overall, this book is detailed and quite easy to get to grips with, provided of course that you are already comfortable with programming in C. However, I would recommend refreshing your knowledge of electronics before reading the book if, like me, you haven't looked at hardware from a low level for some time. As the somewhat shallow introduction to hardware is the only area where I feel the books falls down, I would award it 9/10.

Learning MySQL

Hugh E Williams and Saied Tahaghoghi

O'Reilly Media
ISBN 0-596-00864-3
618pp.
£ 31.99
Published: 28th November 2006

reviewed by John Collins

This book introduces the reader to MySQL.

It kicks off with 93 pages about obtaining and installing MySQL on various systems including Linux, WIndows and MAC OS X. Then we have a section about database design before we move on to discuss how to use SQL to perform all the various database operations. Some of the more advanced operations like JOIN and such are explained quite well although the author clearly has his own strongly-held opinions about which are useful and which should be avoided. The coverage of the operations is far from complete, for example he doesn't go into any detail about locking and use of keys isn't very well covered or explained.

The final parts of the book (nearly 200 pages) cover use of the PHP and Perl facilities to access MySQL. In each case he spends several pages teaching you about PHP and Perl before talking about the MySQL facilities available within those languages. There are extensive exercises and examples at the end of each section, some of which read like student exam type questions.

I think the book is good as far as it goes and generally I wish I'd had it to hand when I started using MySQL, but you will obviously need a proper reference manual for MySQL for any serious use. I rather felt that the huge section on obtaining and installing MySQL wasn't necessary in a "Learning" book where the "learner" will probably start with a system on which it's installed already (most Linux distributions provide it by default) and the introduction to PHP and Perl weren't necessary at all. I think that he would be better referring the user to books which introduce those languages on their own. As a result the coverage of the language facilities for access to MySQL are a bit skimmed over in my view, not even mentioning some facilities in Perl which I use regularly.

Contributors

Mike Banahan is a UKUUG Council member and runs GBdirect and the Cutter Project.

John Collins is a UKUUG Council member and runs Xi Software Ltd which specialises in System Management tools for UNIX and Linux systems. Xi Software is based in Welwyn Garden City and has been in existence for nearly 21 years.

Clive Darke first came across UNIX in the early 1980s as a developer for a U.S. software house. He is now an instructor for QA-IQ and teaches a wide range of UNIX programming subjects, and even some Windows. The only Perl modules he wrote that made it onto CPAN are Win32 specific, since Windows programmers need all the help they can get. His pet peeve is when programmers excuse poor technique with ”but machines are so big and fast these days...”.

Sunil Das is a freelance consultant providing Unix and Linux training via various computing companies.

John Davies is Programme Director of IT4C. He was responsible for commissioning the new IT4C system (www.it4communities.org.uk). Before IT4C, John was in mobile telecoms but he started in IT as a comms programmer and he still likes to tinker with network protocols when nobody is watching. If you want to distract him from IT, try offering him a pint of decent bitter and mention your enthusiasm for science fiction or flying boats. John lives with a highly experienced Unix system administrator but remains an amateur of Unix and open source.

Graham Lee is a Software Testing Manager, in which he ensures his popularity with his colleagues by telling them their code is broken.

Lindsay Marshall developed the Newcastle Connection distributed UNIX software and created the first Internet cemetery. He is a Senior Lecturer in the School of Computing Science at the University of Newcastle upon Tyne. He also runs the RISKS digest website and the Bifurcated Rivets weblog.

Jane Morrison is Company Secretary and Administrator for UKUUG, and manages the UKUUG office at the Manor House in Buntingford. She has been involved with UKUUG administration since 1987. In addition to UKUUG, Jane is Company Secretary for a trade association (Fibreoptic Industry Association) that she also runs from the Manor House office.

Raza Rizvi is the Customer Technical Services manager for Opal Solutions, a B2B ISP and Network Integrator. He is also a non-exec director of LINX, one of the largest ISP peering points in the world. After 12 years working for an ISP, he professes to being a little tired...

Peter H Salus has been (inter alia) the Executive Director of the USENIX Association and Vice President of the Free Software Foundation. He is the author of “A Quarter Century of Unix” (1994) and other books.

Kelvin Vanderlip created and runs the ZFT Ltd legacy asynchronous file transfer service, built entirely from GNU/Linux parts. He has recently fallen in love with Asterisk.

Bob Vickers manages the Computer Science computers at Royal Holloway, University of London.

Paul Waring is a postgraduate student at the University of Manchester and is currently part way through a masters course in Classics and Ancient History. When not lost in the fields of Athenian Democracy or the Late Roman Army, he works as a freelance IT consultant to help pay for an ever expanding library of historical texts.

Alain Williams is Chairman of UKUUG and works as an independent Unix and Linux consultant, running Parliament Hill Computers Ltd.

Contacts

Alain Williams
Council Chairman
Watford
Tel: 07876 680256

Sam Smith
UKUUG Treasurer; Website
Manchester

Mike Banahan
Council member
Ely

John M Collins
Council member
Welwyn Garden City

Phil Hands
Council member
London

John Pinner
Council member
Sutton Coldfield

Howard Thomson
Council member
Ashford, Middlesex

Jane Morrison
UKUUG Secretariat
PO Box 37
Buntingford
Herts
SG9 9UQ
Tel: 01763 273475
Fax: 01763 273255
office@ukuug.org

Sunil Das
UKUUG Liaison Officer
Suffolk

Leslie Fletcher
UKUUG Spokesperson
Manchester

Roger Whittaker
Newsletter Editor
London