news@UK

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

Volume 14, Number 4 **ISSN 0965-9412** **December 2005**

Contents

News from the Secretariat

Jane Morrison

As you will have seen from the AGM minutes circulated with the previous Newsletter, we have two new Council members, John M. Collins and Dean Wilson. Both are enthusiastic about UKUUG and we are hoping they can bring some new ideas and assist with work on ongoing items.

We had the first meeting of the new Council on 26th October (a phone conference) and I am pleased to announce that Ray Miller will continue as Chairman and Sam Smith has been appointed as Treasurer. A face-to-face Council meeting was held in London on 30th November.

A major activity for the new Council is the arrangement of the programme for the Spring Conference which will be held at Durham University on 22nd and 23rd of March 2006. Full event details will be available early in January.

Planning has also started for the Linux 2006 event which, at the time of writing, is going to be held at the University of Sussex, Falmer, Brighton from 30th June - 2nd July. The call for papers for Linux 2006 will appear on the web site soon. We are currently looking for sponsors for the event and if you know any company who may like to get involved please let me know.

The annual membership subscription invoices will be sent out in January, please look out for your invoice and as always prompt payment will be gratefully received!

Please note the copy date for the next issue of the Newsletter (March 2006) is 10th February 2006. We are always looking for interesting subsmissions from members, so if you have any news, articles etc. please send copy direct to
`newsletter@ukuug.org`

The Secretariat will be closed betweed 19th December to 3rd January. I would like to take this opportunity to wish you all a very Happy Christmas and a Peaceful New Year.

UKUUG Spring conference

UKUUG's annual Large Installation Systems Administration (LISA) conference will take place in the historic city of Durham on Wednesday 22nd and Thursday 23rd March 2006.

This is the UK's only conference aimed specifically at systems and network administrators, and attracts a large number of professionals from sites of all shapes and sizes. We are also planning a series of talks on the BSD family of operating systems to run alongside the main systems administration stream. As well as the technical talks, the conference provides a friendly environment for delegates to meet, learn, and enjoy lively debate on a host of subjects.

We have provisional agreement from Samba core team member Jerry Carter to run a one-day workshop, "Managing Samba 3.0", and from Ryan McBride of the OpenBSD project to lead a half-day workshop on OpenBSD networking. The call for participation closes on 23rd December, so you still have a few days to submit a proposal for a talk of your own.
The programme will be announced and bookings open on 16th January. More information is available on the conference web site
`http://www.ukuug.org/events/spring2006/`

UKUUG Tutorial: Python for Programmers

UKUUG has teamed up with Clockwork Software Systems, developers of the award-winning PayThyme payroll application, to offer members a 3-day course on programming with Python. The course will be held at Clockwork's premises in Birmingham from Monday 23rd to Wednesday 25th January 2006.

The cost of the 3-day course (including lunch, refreshments, and printed notes) is £225 + VAT for UKUUG members, or £450 + VAT for non-members. Book online at:
`http://www.ukuug.org/events/python2006/booking/`

Places are strictly limited, so early booking is recommended.

Course Summary

This course is for people who are experienced programmers but who have not used Python before. It teaches the basics of Python, concentrating on gaining an understanding of the language, especially its dynamic nature and introspection.

The course is highly interactive, giving students the opportunity to try out the features as they are introduced, allowing them to gain familiarity with the interpreter and learn how to use Python's self-documenting features to find out what they need to know.

Topics covered include Data Types, Functions and Methods, Functional Programming, Dynamic Typing, Sequences and Mappings, Control Constructs, Modules (using and writing), Persistent Storage and File Handling, and Object Oriented programming with Python.

We assume that students will be familiar with Object Oriented programming, and will concentrate on the differences between Python and other OO languages. However, if you do not have such experience please get in touch with the UKUUG office, and if there is sufficient interest we can either adjust the course accordingly or run an introductory OO day or half-day.

After the basics, we proceed to more advanced Python techniques, including introspection and ensuring that there is a full understanding of the dynamic nature of Python, including the differences between Python 'variables' and variables as we know them, say in C.

The tutors will be David Chan and John Pinner.

After a career in the aircraft and automotive industries, John Pinner founded Clockwork Software Systems, a UK business software supplier, in 1987. Having used several computer languages over a thirty year period, he reached a state of enlightenment only when he discovered Python some seven years ago. He is a strong advocate of open systems, with much of Clockwork's software being licensed under the GPL. Outside work, John's interests include baroque music, valve audio and classic Armstrong Siddeley motor cars.

David Chan is a programmer at Clockwork Software Systems, he is an enthusiastic believer in Free and Open-Source software and is interested in its social and commercial ramifications. In his spare time, David deconstructs enablement software to document its underlying functionality and terrorises owners of Welsh-language bookshops.
For more information, please see the event web-site:
`http://www.ukuug.org/events/python2006/`

Concessionary Membership Category

Council agreed earlier this year to introduce a concessionary membership category for UKUUG. This replaces student membership and is open to Students, Retired and Unwaged people. Proof of status must be seen by the UKUUG Secretariat before the membership is accepted and an annual check of continuing status will be made on subscription renewal.

Concessionary members receive the same rights as individual members, so this change also removes the anomaly of student members not having voting rights.

Annual membership rates remain unchanged:

- Concessionary: £20 + VAT
- Individual: £35 + VAT
- Institutional: £170 + VAT

Full details of UKUUG membership benefits are listed at
`http://www.ukuug.org/membership/`

Data retention laws: FFII press release

EU introducing "Big Brother" anti-privacy law, warns FFII
The EU is passing a "Big Brother" law to track every electronic communication, warns the FFII, an international information rights group based in Munich.

"Imagine a world in which the state follows everything you do. A world where computers watch every step you make. A world in which privacy is dead and the machines can track down every dissident in minutes. A world ruled by unelected agencies, working hand-in-hand with powerful commercial interests. A world in which citizens have no rights except to consume. Science fiction? The Age of the Machines? No, this is Europe, coming to you in 2006."

So warns Pieter Hintjens, president of the FFII. He says, "the EU is about to pass a directive to track every communication you make. This law makes the old Soviet spy states look like amateurs."

He continues "This law goes against our European traditions of civil liberty. It appears to break Article 8 of the European Convention on Human Rights. It will destroy small ISPs and raise prices. To enforce it, the EU will have to shut or monitor every cybercafe, web mail access, and wifi hotspot. Such a regime would be more authoritarian even than China. Even the US, after 9/11, does not have such oppressive laws. The EU does not need this law: it is a bad law, pushed through without respect for the democratic process."

Erik Josefsson of the FFII says: "We are entering into an era of 'I don't have time' legislation. With the expanded competence of the Commission (see consequences of the ECJ Judgement September 13, case c-176/03 Commission v. Council), the underarmed and weakened Parliament stands no chance to do its job properly. The 'sausage machine' is far too easy to abuse."

The Big Brother "data retention directive" makes Internet and telephony providers record "communications traffic data" for up to several years. These huge amounts of detailed personal data can be easily leaked, stolen, and abused. The forces – mainly the UK government – pushing the Big Brother law claim it will prevent terrorism. The FFII does not accept this simplistic argument. The real targets, it appears, are ordinary citizens, going about their daily business.

The FFII president points out, “almost everyone carries a mobile phone. With this law, your mobile phone and web browser becomes Big Brother’s way of watching you. You will never be alone again. If you do not like this idea, contact your MEP today, urgently, and explain why it worries you. On 13 December 2005, personal privacy becomes history.”

About the FFII

The Foundation for a Free Information Infrastructure (FFII) is a non-profit association registered in several European countries, which is dedicated to the spread of data processing literacy. FFII supports the development of public information goods based on copyright, free competition, open standards. More than 850 members, 3,000 companies and 90,000 supporters have entrusted the FFII to act as their voice in public policy questions concerning exclusion rights (intellectual property) in data processing.

`http://www.ffii.org`

OSDL Linux desktop survey

The OSDL has been conducting a survey into Linux desktop usage.

The survey is here:

`http://developer.osdl.org/dev/dtl/survey/start.shtml`

Results from the survey can be found at

`http://www.osdl.org/dtl/DTL_Survey_Report_Nov2005.pdf`

Proposal for an All-Party Parliamentary Open Source Software (OSS) Group

Leslie Fletcher

UKUUG has begun working for the setting up of an All-Party Parliamentary Open Source Software (OSS) Group; this article explains some of the background to this. There will be further articles as the project progresses.

The need for parliamentary scrutiny of UK government policy and practice

According to many observers, UK government policy on the deployment of OSS in the public sector is opaque, confused and implemented differently between departments. In October last year, the Office of Government Commerce (OGC), published a report that said open source is “a viable desktop alternative for the majority of government users”. But, shortly after the publication of this report, the NHS awarded Microsoft a nine-year contract to put its software on 900,000 computers. More recently, in June 2005, Microsoft signed a £6 million contract to update and support Windows-based applications at the Foreign and Commonwealth Office for another three years. Neither the NHS, nor the FCO appear to have made any attempt to “examine carefully the technical and business case for the implementation of Open Source software” as required by the OGC report.

A sponsorship deal signed in March between Microsoft and the Department for Education and Skills has positioned the giant to be the dominant supplier in English schools, and according to The Register, is causing some schools to cancel open source projects “in case they upset the sponsor [i.e. Microsoft] or the DfES failed their bid.” The rules of the mechanism Microsoft has used, sponsorship for 100 applicants for specialist schools status, at a claimed value of £1.5

million, appear to have been relaxed or even subverted, the effect being to tilt the playing field dramatically away from OSS.

Other European countries have shown what appears to be a more proactive approach to encouraging the use of OSS in public institutions. The Danish and German governments have published extensive guidelines on the use of free software in government and recently the French government established a collaborative platform allowing the sharing and development of open source software across the entire French public administration. One highlight comes from Extremadura, Spain: free software made one of EU's poorest regions the winner of the EU Regional Innovation Awards in 4 years. The regional government has more than 80000 desktops running the free gnu/Linux operating system and other Spanish provinces are following this example.

What would an All-Party Parliamentary OSS Group do?

At first sight, the appropriate parliamentary body to scrutinise UK government policy on OSS is the Select Committee Committee on Science and Technology which, the ordinary citizen might imagine, looks at any aspect of government policy affecting science and technology. In fact its terms of reference are limited to overseeing the work of the Office of Science and Technology in the DTI and its associated bodies. Similarly, OGC is part of the Treasury so reports to the Treasury Select Committee, schools expenditure is looked at by the Education Select Committee and so on. According to former MP Richard Allan, “this is a real pain as the separate elements [of OSS] are often too small to be taken seriously by their respective [select] committees.”

A way round this used to get parliamentary scrutiny of government policy on the Internet was the All Party Parliamentary Internet Group (APIG), a kind of ad-hoc select committee which can set its own terms of reference. There are recognised all-party groups on a whole host of subjects.

An all-party group works by gathering evidence on a specific topic within its terms of reference and writing a report. An All-Party Parliamentary OSS Group should aim at one topic per year – so four in the lifetime of the present parliament. The group would gather evidence by holding two morning meetings at the House of Commons at which invited experts offer their views and are questioned by group members. A report is prepared and launched with whatever fanfare the group thinks appropriate. This is very effective for profile raising and getting the ear of ministers. The sequence of events is illustrated by APIG's report on spam.

If and when an All-Party Parliamentary OSS Group is set up, its first topic could be “OSS in Schools” following up the BECTA report.

Establishing an All-Party Parliamentary OSS Group

An all-party group needs members from all (major) parties in parliament, including 4-5 MPs with a keen interest in the topic to serve as officers. For example, according to its website, APIG has 5 officers and around 60 other members. Each party has a quota of members. Richard Allan retains good contacts in parliament and is approaching key parliamentarians to start the process of establishing an All-Party Parliamentary OSS Group.

What is UKUUG doing?

An all-party group needs resources from outside parliament. Fundamentally, this means providing resources to employ people to

- set up and manage the meetings
- draft and edit the reports.

The launching of the report also has to be paid for from outside (all-party groups can only accept outside resources in kind). On the other hand, parliamentary resources are used to publish the report and Hansard report writers can be used to minute the meetings. An All-Party Parliamentary OSS Group will need about six person weeks per year of outside support to function effectively, plus whatever is needed to give reports high profile launches. In financial terms, a total of about £12,000 per year is needed.

UKUUG, in collaboration with Open Forum Europe (OFE), is coordinating financial support from the UK OSS community for an All-Party Parliamentary OSS Group. UKUUG has earmarked some of its own funds for this purpose and three other groups within the OSS community have already offered financial support. OFE has offered the help of its own parliamentary lobbyists to arrange preliminary meetings in the Palace of Westminster; a steering group is being set up to coordinate this.

What can UKUUG members do?

UKUUG members have vital parts to play. Right now, further financial support would be very helpful; please send details of organisations who could be approached to me at
`Leslie.Fletcher@ukuug.org`

Next, it will be necessary to convince MPs that an All-Party Parliamentary OSS Group is likely to have a useful purpose. In due course, members will be asked to press their MP to give their support; please watch for a further announcement about this.

Once an All-Party Parliamentary OSS Group is established, it will need to have

- Access to a pool of technical expertise
- Ready communication with a membership who can be recruited to provide information
- Experience from the sharp end of IT, to avoid the pitfalls of purely academic research
- Independence of vendors and other sectional interests
- Advice on topics for investigation and identification of suitable experts

Members will recognise that this means them!

UK Open Source Media Watch

UKUUG is pleased to announce that we have received funding from the Open Source Academy for our Media Watch blog, which has been re-launched as the “UK Open Source Media Watch”.

The Open Source Academy is a partnership of local authorities and organisations with experience in the Open Source world, backed by the Office of the Deputy Prime Minister through the e-Innovations programme. Further details of its activities are at
`http://www.opensourceacademy.gov.uk/`

You can read the blog online at
`http://www.ukuug.org/mediawatch/`

The URL for the blog’s RSS feed is
`http://www.ukuug.org/mediawatch/wp-rss2.php`

We have also introduced a mailing list to receive notifications when entries are added to the blog. To subscribe, send an empty message to
`mediawatch-alerts-subscribe@lists.ukuug.org`

OSS Watch Conference

Sebastian Rahtz

OSS Watch, the JISC-funded advisory service on open source for further and higher education, invites you to a 3 day conference on Open Source and Sustainability next spring. It will be held in Oxford, in the comfortable surroundings of the Said Business School, from April 10-12 2006, with a mixture of plenary speakers and workshop sessions.

Open source has proved itself as a development and distribution model that can deliver software which is functional, efficient, innovative, and cost-effective. What is the long-term future?

- Will the communities we see developing software still be working in the same way 10 years from now?
- Should open source thinking be part of the curriculum as well as the deployment strategy?
- Will the next generation of open source programs continue to innovate?
- How can current publicly-funded projects establish a long-term presence?

The conference will address these subjects from the point of view of

- educational institutions needing a strategic vision for their software needs
- publicly-funded projects looking for a long-term development plan
- software companies aiming to maintain or build a business around open source
- public funding agencies mandated to develop an infrastructure

To register your interest in the conference and receive further information as soon as booking opens, visit
`http://www.oss-watch.ac.uk/events/2006-04-10-12/`

Press Release: Linux Emporium comes to Brum!

Steve Whitehouse, who has been running the well known Linux Emporium for the past two years has transferred the business to Thyme Software Ltd, in a seamless move facilitated by Steve's involvement in training the people at Thyme.

John Pinner, Managing Director of Thyme Software: "At Thyme we have great plans for expanding the repertoire of the Linux Emporium, adding appropriate hardware and applications to its list offerings, plus a lot of other great stuff besides, to build on its role as the one-stop buy-on-line shop for everything Linux."

Pinner is perhaps best known as the founder of Clockwork Software Systems, one of the leading UK Open Source software developers and a major contributor to local, national and international Open Source community activities, ranging from local LUGs to PyCon and many points between. Clockwork will continue under Pinner's stewardship as an distinct entity under the Thyme Software umbrella from early next year.

"We recognise that customers are the bedrock of any business, and the Linux Emporium will be looking to consult the people who use it about the products and services they want to see it providing," said Pinner; "We know that fulfilling their needs is our key to success."
The Linux Emporium can be contacted on by phone on 0121 313 3857 and the web site is at
`http://www.linuxemporium.co.uk/`

O'Reilly European Open Source Convention

The European Open Source Convention organised by O'Reilly was held in Amsterdam in October. The following information is excepted from O'Reilly's press release published after the conclusion of the event.

Europe has become fertile ground for open source projects and innovators: European governments are beginning to integrate FLOSS (free/libre and open source software) in innovative ways, and open source communities – particularly on the professional level – are multiplying and gaining influence across the continent. To support and further this open source momentum, O'Reilly Media held its first O'Reilly European Open Source Convention (EuroOSCON) at the Hotel Krasnapolsky in Amsterdam on October 17-20, 2005.

Nearly 500 developers, programmers, hackers, and systems and network administrators attended tutorials, sessions, on-stage discussions, informal events, and hallway conversations focusing on almost every aspects of the open source platform. Mature technologies were explored alongside newer and less developed tools, allowing attendees to take in the full range of open source's capabilities.

Like OSCON, its US counterpart, EuroOSCON brought together diverse people, projects, and communities. Delegates heard from leaders, experts, and alpha geeks leading the open source charge throughout Europe and the world, including: Simon Phipps, Sun; Alan Cox and Michael Tiemann, Red Hat; Jason Matusow, Microsoft; Damian Conway, Monash University; Larry Wall, Perl guru; David Heinemeier-Hansson, Less Software; Cory Doctorow, EFF; Tim O'Reilly, O'Reilly Media; Chet Kapoor, IBM; Jeff Waugh, Canonical; Paul Everitt, Zope Europe; Paula LeDieu, Creative Commons International; Marcel den Hartog, Computer Associates; Luis Casas Luengo, Extremadura; Rasmus Lerdorf, Yahoo!; Alex Martelli, Google; Autrijus Tang, open source advocate; David Axmark, MySQL AB; Ben Laurie, Apache Software Foundation; Danese Cooper, Intel; and Karoly Negyesi, Drupal.

O'Reilly Media's MAKE magazine made a big international impression at the conference, hosting a packed evening reception that featured hands-on demonstrations from makers. EuroOSCON also offered an exhibit hall filled with exciting, cutting edge products. Sponsors include Computer Associates, IBM, Microsoft, ActiveState, Alfresco, Intel, Linagora, MySQL AB, Oracle, Red Hat, Sleepycat Software, SpikeSource, and Zimbra.

Widespread adoption by European governments and organisations has signalled strong support for FLOSS technologies and has put that part of the world squarely in a leadership position in the open source space. The O'Reilly European Open Source Convention will be back next year to continue to support the open source cause in an international forum.

O’Reilly conferences bring together forward-thinking business and technology leaders, shaping ideas and influencing industries around the globe. For over 25 years, O’Reilly has facilitated the adoption of new and important technologies by the enterprise, putting emerging technologies on the map. O’Reilly conferences include: the O’Reilly Emerging Telephony Conference; ETech, the O’Reilly Emerging Technology Conference; the MySQL Users Conference, co-presented with MySQL AB; Where 2.0 Conference; OSCON; and Web 2.0 (co-hosted by Tim O’Reilly and John Battelle, and co-produced with MediaLive International).

Details of the conference can be found at
`http://conferences.oreilly.com/eurooscon/`

Speaker presentation slides etc. can be found at
`http://conferences.oreillynet.com/pub/w/41/presentations.html`

Other upcoming O’Reilly conferences include the O’Reilly Emerging Telephony Conference, the O’Reilly Emerging Technology Conference, the MySQL Users Conference, Where 2.0, and the next US and European Open Source Conventions.
Details of all these events can be found at
`http://conferences.oreilly.com`

Why I love my Mac

Elena Blanco

It is a well known fact that people who have Macs love them. Non-believers might even say they get a bit evangelical. It’s a bit like people who drive Skodas. They have always been fantastically brand loyal but somehow it has suddenly become a pretty cool brand outside its traditional customer base. In the same way that a Skoda is a VW underneath the paint, Mac OS X brought us a Unix desktop with an exceptional paint job. The result is that the Mac has become the Unix desktop I always wanted.

To my mind the biggest problem with all those Linux distributions out there is that no-one has yet cracked the GUI problem. How do you give the user a desktop environment of the same quality as the underlying operating system? My background is as a Unix sysdamin. No-one needs to convince me of the qualities of a Unix operating system. I do, however, believe that Unix for the desktop, rather than the back office, falls woefully short when it comes to the user experience. Yes there are desktop environments available. GNOME and KDE certainly have their fans, but I’m not one of them. The first time I saw Mac OS X though, it blew me away. Perhaps it was inevitable that a Mac user would say this but it *Just Works.* And as if that wasn’t enough, it looks fantastic whilst it is *Just Working.*

The care and attention to detail evident in the look and feel of Mac OS X is staggering. It is no coincidence that Apple have a stronghold in the creative industries. These are people who make careers out of aesthetics. So it’s natural that they would pick a computing platform made by people who cared about the same things. Outside this specialised community, some people remain totally unaffected by the look and feel of their computing environment. They do not have strong feelings either way. These people are unlikely to ever do anything other than go with the flow. They’ll use whatever GUI is put in front of them. Some people go further still and think that using anything other than the command line somehow marks you out as a lesser being. These people will rarely be interested in a GUI and clearly do not represent the views of the majority. Of course the command line has its place but that place is not as the desktop of an end user.

Apple’s heavy commitment to the aesthetic design of its products also extends to its hardware. It may be expensive when compared to a commodity Intel box but that is like comparing a Jaguar to a Vauxhall, they are bought by people with different criteria.

But Mac OS X is not open source I hear you cry? That’s right, it’s not. However, components of Mac OS X are open source and Apple as a company is committed to the open source development model. Darwin, the Unix core of Mac OS X is made available under the Apple Public Source License, an OSI certified licence. WebCore, the HTML rendering engine at the heart of Apple’s web browser, Safari, is open source software. Apple has encouraged the formation of a large developer community where Apple engineers and the open source community collaborate on a variety of projects, most notably on Darwin, the core of Mac OS X. The Apple web site tells us that Apple believes the open source development model is the most effective for certain types of software. In particular they stress the importance of the peer review process in the effort to make the underlying operating system both robust and secure.

Whether or not Apple’s commitment to open source is a factor in your choice of desktop, I believe that it is the best Unix-like environment when we consider the user experience as well as the underlying operating system. In exactly the same way as Windows users can, Mac users can use open source applications on top of their proprietary operating system. In fact, since Mac OS X is Unix, it is even easier for us to use open source applications than it is for Windows users as our familiar tools are all available. And if compiling from source is not to your taste it is increasingly common for open source software projects to produce binaries for Mac OS X alongside binaries for other flavours of Unix.

Some of these reasons may sound trite but for most people their computer is only a tool to do a job. Using a desktop environment is not something done for its own sake. It is done in order to achieve a specific task. The better that desktop environment suits the user, the more efficiently the end task is achieved. Most users do customize their desktop to some extent so choosing the right desktop environment is important. For me Mac OS X offers the holy grail: the power of Unix with an elegant, intuitive, rock solid desktop environment. And when I choose to use an open source application like Firefox, my browser of choice, it runs just fine on my platform of choice. That really is the best of both worlds.

Copyright ©University of Oxford. This article is licensed under the Creative Commons Attribution-ShareAlike 2.0 England & Wales licence.

Adventures in BSD

Michael Josefsson

I am sometimes asked “Why do you use FreeBSD?” My usual ‘auto-response’ is “Because it is the best!” While I truly think it is the best, this reply does not really answer the question. I hope this article will better explain my feelings.

Way back in 1994 I became the manager of a hardware laboratory at my old University. I was to manage the labs in all aspects, from computers down to data sheets, pliers, and students. With over 300 students attending each year, this was a very time-consuming task. Of the three labs, only two had computers and networks in them. In the third lab, 16 serial lines were connected through a Xyplex terminal server to a computer of unknown pedigree. (It later turned out to be a Sun 3.) I had been into computers since the early 80s and PCs since 1989, but I had never touched a network of any kind.

The two labs were connected by a Lantastic LAN. Only one area of the server’s disk was shared and every student stored his own home-directory on it. As long as everything worked fine, and it generally did, I had better things to do than trying to understand what this network setup was all about. The odd downtime was mostly due to students tampering with the RG-58 coax that ran from one machine to another continuing on into the next lab. Problems were easy to locate. I knew the coax had to be terminated with a 50-ohm BNC terminator and moving this terminator to strategic places along the cable always led me to the culprit.

Then there was a software upgrade. The labs had been running MS-DOS with a DOS-based compiler for many years. The new version of the compiler required MS Windows. (At the time this meant Windows for Workgroups 3.11.) I could not get the old LAN software to run under Windows. I don't remember why, perhaps I was a victim of my own ignorance. After reading the Windows docs I managed to set up a common workgroup for the two labs. With a 486/66 Windows box working as a file server we again had a fully functioning lab.or so we thought!

I found that it was possible to access the server's shared area and was quite pleased with my doings. I compiled some typical code from a client machine and everything ran as expected. I did not test network performance under a higher load. The idea never crossed my mind.

During one compilation, several large, multi-megabyte files were created in the project's home directory. This was in itself not a problem as the network's bandwidth was not saturated. However, all this data moving in and out of the server turned out to be detrimental to its health. With 10-15 clients chewing away, the load on the server increased. After a period of time ranging from a few minutes to several hours, the server would unexpectedly crash. A reboot got everything working again. All source files had been auto-saved before entering the compilation stage so it was easy to start over and hope for more success the second time. Still, the situation was less than satisfactory.

At this point, OS/2 entered the story. In the outside world, Windows 95 was being deployed. I got my hands on an old OS/2 2.1 demo CD. After several problems I got it up and running. I thought it was cool and ordered my own OS/2 Warp soon after. Warp was my favourite desktop for years to come. I really liked the smooth user interface and enjoyed the increased stability compared to the Windows version I had used earlier. A co-worker with a similar interest in computers used Windows 95 and had to re-install every three to four months. The system would somehow clog itself up beyond recognition. My OS/2 machine ran happily all the time. Great! Except for the RSA DES Challenge. Being outside of the United States I took an interest in the Swedish-run SolNet DESattack. With SolNet's limited resources there was quite some time before an OS/2 executable was published. The situation annoyed me.

So what to do? I was unwilling to give-up OS/2 and surrender to Windows 95. I had started to experiment with Linux, a Unixlike system but felt very little enthusiasm for it. Because of an earlier experience with HPUX, I was under the impression that Unix only represented an extremely complicated way of doing things, and therefore, Unix was ruled out.

At this time, there were executables for FreeBSD available. Not knowing much about FreeBSD, I made an FTP install of 2.1.7-STABLE. The DES-client ran as expected. I figured out nice commands; `kill -STOP`, `kill -CONT` and that putting an ampersand (&) after the command line ran a job in the background. Cool! I could manage every aspect of the program. A taste of a new world! This was very enticing.

My success was short-lived. There was soon a new DES client requiring newer libs, which forced me to install 2.2.2. This time I ordered the 2 disc FreeBSD set from Walnut Creek sometime around June 1997.

Before the summer I had plans to replace the lab server with OS/2, but during the summer, I experimented with FreeBSD and learned about TCP/IP, Samba, FTP and Unix.

The client machines in the lab were using Windows 95 and since we had to get rid of Windows for Workgroups anyway, I installed FreeBSD on each and every machine. I made further experiments with the r*-services (`rsh`, `rcp`, `ruptime`, etc.) among other things. All in all, the experiments made me confident with managing FreeBSD. In the beginning of August, when I had to wire-up the lab's new infrastructure, I made FreeBSD run the lab for me. That turned out to be a wise move; not one spontaneous reboot occurred during the years to come.

With a busy lab, reliability is of major importance. The logs show students accessing the server around the clock. FreeBSD really is reliable; the server's longest uptime to date is 220 days.

By now, things were moving fast. The third lab was also computerized. (The introduction of Microchip's PIC line of processors necessitated this.) I added a couple of hubs and the labs were interconnected. The next summer I set up a local name server and a private domain on a FreeBSD box with two NICs – one for the internal net and one for external access. There was no forwarding between the two interfaces. This setup allowed me, as admin, to download software which I then installed on the clients. Each project team in the lab had its own home, with permissions fixed so they could not peek at each other's work. They could also edit files at home and upload via FTP. I added the system user as a way to keep a central repository of programs and files that every client would need – Adobe Acrobat for Windows for example, printer drivers, upgrades and other stuff.

Until now all data sheets had been in a folder in my room. The projects' supervisors had access to the folder and make copies which they handed out to the project teams. There were 95 steps to the photo copier, and with 95 steps back from it, handling the data sheets became more and more cumbersome as a variety of components were added. What to do? Install Apache! Putting the data sheets on the server in PDF format has eliminated the actual paper handling. This was a truly brilliant move that I wish I had thought of earlier. Now each project group could peruse the available components' data sheets, both from home and when in the lab before committing the device into their project. Some students printed out data sheets of course, but I see a trend of more and more students reading the specs directly off the screen.

There is a lot of activity in the labs. With up to 75 students there at the same time, not all of them can be busy with their assignment; installing MP3s on the clients has become a popular pastime. It is quite possible that the especially zealous students fiddle around with the system settings of Windows 95 and cause damage! To simplify the Windows re-installation, we have made a complete 2 gigabyte image of the Windows partition which easily can be downloaded and written onto the machines. While this may sound rather brutal, it works well in practice. In fact, every machine in the lab was born this way.

Needless to say, the client machines dualboot between Windows 95 and FreeBSD. It is a particular pleasure to find the students shutting down Windows and rebooting into FreeBSD because they feel more at home with it.

Earlier I mentioned we had a Sun 3 box. That particular machine has gone to the scrap heap and the FreeBSD server has replaced it. Using the ports, I installed the `m68k-coff` cross-compiler and spent some time rebuilding the libs needed for our particular hardware. We were able to get rid of `gcc-1.40` and now use `gcc-2.8.1` for our in house built 68008 computer card.

Some small bits and pieces complete the picture. The server has been endowed with a Tandberg SLR-5 tape backup. Once a week I do a level 0 dump and every night a cron job initiates a level 1 incremental backup. The machines' disks are mirrored with `vinum(8)`. Thus the entire `/usr` is safeguarded from disk failure. To ease new installations and administration, each client has a `P:\` (P for program) directory with all non-standard Windows 95 software. It is now sufficient to install new software on only one client; it will become available to the other clients at the next log-in. Every home share is attached as `Q:\` on the client. To this end, Samba is configured as an NT domain controller. Log-in scripts handle the actual attachment of the devices at log-in time. On the FreeBSD side I use NFS and `amd(8)` to automount the users' home directories upon log-in.

I have unfortunately left OS/2. I really liked the Workplace Shell, but the functionality built into even a basic FreeBSD system makes me more productive. I have now been using FreeBSD exclusively for over two years for all my desktop needs. My experience is best summarized by something I saw someone post to one of the FreeBSD mailing lists: "I'm home."

Introducing the Template Toolkit Part 1 - The Basics

Dave Cross

Template processing is a method of producing output that takes some fixed (or boilerplate) text and puts variable data values inside it. The most obvious example is that of a form letter. When you get that prize draw win notification from the Readers Digest it has been made to look as though it is personal to you when, in fact, a few pieces of information about you have been inserted into a letter template.

The Template Toolkit is a piece of software that allows you to carry out powerful template processing operations. It is written in Perl so it runs on pretty much any computing platform that you can name, but you don't need to know any Perl in order to use it. If, however, you do know Perl, then a whole extra level of power is opened up to you.

Using the Template Toolkit

Installing TT gives you two new command line utilities – `tpage` and `ttree`. You use `tpage` to process a single template file and write the output to `STDOUT`. If your needs are more sophisticated, `ttree` can process a whole directory tree of templates and write the output to another directory. We'll start by using `tpage`.

The `tpage` command takes only one argument, `--define`. This allows you to define data variables to be inserted into your template. You can use many variable definitions at once. You would therefore use it from the command line something like this

```
tpage --define name='Mr Cross' --define amount='100' \
--define due='1st April' letter.tt
```

This defines three variables called `name`, `amount` and `due` which can now be referenced within the template. The template itself is in the file `letter.tt` and it looks like this:

```
Dear [% name %],

According to our records you owe us £ [% amount %].

Please pay before [% due %] or we will send the boys
round.

Regards.
```

You can see from this simple example that the places where you want you variables to be inserted are marked with `[% ... %]` tags. These are known as template directives. We'll look at many different types of directives over the next few months, but currently we are looking at the simplest of directives which just contains the name of a variable. When the template is processed, this directive will be replaced by the current value of the variable.

So, if we process this template using the command line we saw above, we'll get this output:

```
Dear Mr Cross,

According to our records you owe us £ 100.

Please pay before 1st April or we will send the boys
round.

Regards.
```

Using Plugins

Obviously, if we had to type a command line like that every time we had to send a form letter, we really haven't gained much over typing them all in manually. In a while, we'll see a way to make this easier, but first let's take two minutes to tidy up the output a little.

Currently, our template just takes the values we give it and reproduces them. For the amount due, it might look better if we forced the value to always have two decimal places. We can do this by using the Format plugin.

In TT, plugins are a way for the template processor to provide extra functionality by interacting with external resources. In this case the Format plugin provides an interface to a C-style "`printf`" feature. We change the template so that it looks like this:

```
[% USE money=format('%.2f') -%]

Dear [% name %],

According to our records you owe us £ [% money(amount) %].

Please pay before [% due %] or we will send the boys
round.

Regards.
```

We've added a `USE` directive which loads the Format plugin and we've used the plugin to create a formatting function called `money`. Any expression that is passed to this function is formatted according to the format string that was used when `money` was created. In this case, the `%.2f` ensures that the amount has two decimal places. If we process out template using the same call to tpage as before, we now get slightly changed output.

```
Dear Mr Cross,

According to our records you owe us £ 100.00.

Please pay before 1st April or we will send the boys
round.

Regards.
```

Another small change that we've made to our template is the addition of a '`-`' character at the end of the `USE` directive. Generally, all whitespace outside of a directive is passed straight through to the output, but the '`-`' character tells TT to ignore the whitespace (including newlines) that follow the directive. The net effect of this is that adding the `USE` directive doesn't add a blank line to the output.

TT comes with a useful set of standard plugins. We'll see some more of them in the next few sections.

Reading data from a file

As I mentioned before we haven't really gained much if we have to type in a long command each time we want to process a form letter. It would be far easier if we could read the data in from some external source. And, of course, we can. We can read data from all sorts of external sources. We'll start by reading the data from a text file.

We'll assume that our data file has the following format:

```
name : amount : due
Mr Cross : 10 : 1st April
Mr Smith : 20 : 1st March
Mr Jones : 50 : 1st February
```

The first line of the file contains the names of the fields and the other lines contain the actual data. We can now change our template to look like this:

```
[% USE money = format('%.2f') -%]
[% USE debtors = datafile(file) -%]
[% FOREACH debtor = debtors %]
Dear [% debtor.name %],

According to our records you owe us £ [% money(debtor.amount) %].

Please pay before [% debtor.date %] or we will send the boys
round.

Regards.
[%- END %]
```

As you'll see, we've added another directive which loads the Datafile plugin. This plugin opens the given data file and returns an iterator object which can be used to access the data in the file. We assign this iterator object to the variable `debtors`. Note that the name of the file to be used is in the variable `file`. This is now the only variable that we need to pass to the template processor.

Having opened the data file we can process it a row at a time using the `FOREACH` directive. `FOREACH` iterates across a list, setting the loop variable to each element of the list in turn. In this example the loop variable is called `debtor` and each time round the loop it gets one of the values returned by the `debtors` iterator. These values are objects that contain the individual data items from each row in the data file. You can access these items using a dot notation, for example the name from the current row of data is in `debtor.name`. We have changed the names of the variables used in the rest of the template to reflect this.

The end of the `FOREACH` loop is marked with an `END` directive and notice the '`-`' and the start of that directive which strips out any whitespace preceding it (i.e. the newline before it.)

Of course, the random format that I chose for the data file just happened to be the default format for the Datafile plugin (using colons as the delimiter) but it's easy to use an alternative delimiter. For example, if our debtors file had been delimited with pipe characters we would have used code like this:

```
[% USE debtors = datafile(file, delim => '|') %]
```

The delimiter can be surrounded by optional whitespace which is removed from the values. The first row must contain the names of the data items and any blank lines or comment lines (which start with a # character) are ignored.

Splitting the output

The remaining problem is that this prints out all of the letters in a continuous piece of text, but actually we want each letter on a separate page. We can achieve this by inserting a form-feed character (`0x0C`) at the end of each page like this:

```
[% USE money = format('%.2f') -%]
[% USE debtors = datafile(file) -%]
[% FOREACH debtor = debtors -%]
Dear [% debtor.name %],

According to our records you owe us £[% money(debtor.amount) %].

Please pay before [% debtor.date %] or we will send the boys
round.

Regards.
[% UNLESS loop.last -%]
^L
[%- END %]
[%- END %]
```

The `^L` is the representation of a form-feed character when displayed in many text editors.

There are a couple of other changes that I've made to the template. These prevent an extra form-feed being output after the last letter. A printer will automatically perform a form feed at the end a print job, so if our output contains one as well there will be two form-feeds in the job and an extra (blank) sheet of paper will be used. We can prevent that using the code shown.

The code uses an `UNLESS` directive to optionally display part of the template. `UNLESS` works like `IF` but the logic is reversed, the code within the block is executed only if the `UNLESS` condition is false.

In this condition we check the `loop` variable. This is a special TT internal variable which contains various interesting pieces of information about the current `FOREACH` loop. There are boolean flags `first` and `last` which return true only if you are in the first or last iteration respectively. The `size` data item contains the number of elements in the list and the `count` and `index` items contain slightly different views of the current iteration. `count` gives you the number of the iteration (from one to `size`) and `index` gives you one less than that number (useful if you're a Perl programmer and used to array indexes that start from zero).

In our current example we use `loop.last` to avoid printing the extra form-feed.

That's all fine if you have your data in a suitable data file. But maybe it's stored in a database instead.

Accessing a database

The great thing about TT's plugin system is that it's very easy for your template to get data from all sorts of interesting places. For example, there's a plugin to Perl's database interface system, `DBI`. This allows your template to access data stored in most kinds of database.

Assuming that our data is stored in a table called debtors in a MySQL database called `accounts`, we can change our template to look like this:

```
[% USE money = format('%.2f') -%]
[% USE DBI(database = 'dbi:mysql:accounts'
username = 'acc_user'
password = 'sekrit') -%]
[% FOREACH debtor = DBI.query('select name, amount, due from debtors') -%]
Dear [% debtor.name %],

According to our records you owe us £ [% money(debtor.amount) %].

Please pay before [% debtor.due %] or we will send the boys
round.
```

```
Regards.

[% UNLESS loop.last -%]
---------------
[%- END %]
[%- END %]
```

It's interesting to note just how few changes we have had to make here. All of the loop code is identical, it's just the code that sets up the loop that is different.

The new code simple enough to follow. We load the DBI plugin, passing it the various parameters required to connect to the database. We then run an SQL query against the database. This returns an iterator object that we can use in a FOREACH directive in exactly the same way that we used the iterator that the Datafile plugin created. Each value returned by the iterator is object which contains data items for each of the columns selected from the database. The names of these data items are given by the column names in the SQL statement.

Formatting dates

There's another formatting improvement that we can make at this time – we can reformat the date. If we assume that the due date is stored in a database date column then most databases will return it in the format YYYY-MM-DD which isn't very user friendly. We can use the Date plugin to fix that. We can load the Date plugin with a directive like this:

```
[% USE date(format = ’%d %B’) -%]
```

And then use it with a directive like this:

```
[% date.format(debtor.due) %]
```

Here we have given the Date plugin a format string. This is in the same format as the format strings used by the Unix date command. Our %d %B format gives us the day of the month followed by the full month name. Having seeded the plugin with that format, any dates that are passed to the date.format function are converted to that format.

If you want to override the default format at any time, you can pass a new format definition as the second argument to the date.format function like this:

```
[% date.format(debtor.due, ’%A, %d %B %Y’) %]
```

As %A is the full weekday name and %Y gives the year, this example will display the due date in the format “Sunday, 01 February 2004”.

One small problem with the Date plugin is that it is a little fussy about the format of the date. It only accepts dates as either the number of seconds since 1st Jan 1970 (known as the Unix epoch) or in the format h:m:s d/m/y. That's not the format that we're currently getting from the database so we'll need to use MySQL's date_format function to correct that.

```
[% FOREACH debtor = DBI.query(’select name, amount,
date_format(due, "%h:%i:%s %d/%m/%Y")
as due
from   debtors’) -%]
```

Two things to notice here. Firstly, whilst the MySQL date foramt strings look at lot like the standard Unix ones used by TT, they are actually different (`%i` for minute instead of `%M`, for example). Secondly, we’ve added a column alias (“as due”) to the date column so the the `DBI` plugin continues to use that name for the column.

Dealing with XML

So we’ve managed to extract our date from data files and databases. What if our data is stored as an XML document? As you’d expect, TT has plugins to handle that too. In this example I’ll use the XML.XPath plugin to access data within a document. Let’s assume that our debtor document looks like this:

```
<debtors>
  <debtor>
    <name>Mr Cross</name>
    <amount>10</amount>
    <date>1st March</date>
  </debtor>
  <debtor>
    <name>Mr Smith</name>
    <amount>20</amount>
    <date>1st February</date>
  </debtor>
  <debtor>
    <name>Mr Jones</name>
    <amount>50</amount>
    <date>1st February</date>
  </debtor>
</debtors>
```

Here’s the template that we’ll use to process it.

```
[% USE money = format(’%.2f’) -%]
[% USE debtors = XML.XPath(file) -%]
[% FOREACH debtor = debtors.findnodes(’/debtors/debtor’) -%]
Dear [% debtor.findvalue(’name’) %],

According to our records you owe us £ [% money(debtor.findvalue(’amount’)) %].

Please pay before [% debtor.findvalue(’date’) %] or we will send the boys
round.

Regards.
[% UNLESS loop.last -%]
^L
[%- END %]
[%- END %]
```

The template first creates a `XML.XPath` object by passing the name of the XML file to the `XML.XPath` plugin in the `USE` directive. This `XML.XPath` object can then respond to `XPath` queries in a number of ways. The first way that we use is to use the `findnodes` function to get a list of all of the nodes matching the query ‘`/debtors/debtor`’. This returns an iterator that we can use in a `FOREACH` loop like all of the iterators that we have seen previously. Each object returned by this iterator is another `XML/XPath` object representing one of the nodes in the set. In this case each element is a ’debtor’ node that is contained within the main ‘`debtors`’ node. We can then use the `findvalue` function to get the text value contained in the various nodes that we are interested in.

Getting the Template Toolkit

If you're happy installing Perl modules from the Comprehensive Perl Archive Network then you can go to
`http://search.cpan.org/dist/Template-Toolkit/`
and download it from there. It is installed in exactly the same way as most other Perl modules.

If you'd rather not get involved with CPAN then there are also Debian packages and RPMs available. The Debian packages can be downloaded from the Debian packages respository at
`http://packages.debian.org/unstable/interpreters/libtemplate-perl`
and
`http://packages.debian.org/unstable/doc/libtemplate-perl-doc`
The RPMs can be found by searching at
`http://rpmfind.net`
The package name is `perl-Template-Toolkit`.

If you need any more information about installing TT or you want to use the bleeding edge CVS versions, then you can access those at the official TT web site at
`http://www.template-toolkit.org/`

Template Toolkit Documentation

This articles has just scratched the surface of the Template Toolkit. We'll be going into more detail later in this series, but in the meantime you can get more information from a number of sources.

Distribution Documentation

The TT distribution comes with a large number of manual pages. The best place for a beginner to start is probably with `man Template::Manual` which is a guide to all the other TT manual pages and `man Template::Tutorial` which introduces a couple of tutorials that are part of the documentation set.

Web Page

The official TT web page is at
`http://template-toolkit.org/`
Here you'll find all the man pages online together with a number of talks about TT, the latest version of the software and a number of other interesting things.

Mailing list

There is a mailing list for the discussion of things relating to TT. You can subscribe to it at
`http://www.template-toolkit.org/mailman/listinfo/templates`
All of the core TT developers are regular posters to this list.

Book

The book "Perl Template Toolkit" by Darren Chamberlain, David Cross and Andy Wardley has recently published by O'Reilly. You can get more details about it from
`http://www.oreilly.com/catalog/perltt`

Podcasting Hacks

Jack D Herrington

O'Reilly and Associates
ISBN 0-596-10066-3
453pp.
£ 17.50

reviewed by Gavin Inglis

Gone are the days when you were cutting edge because you had a blog. Now, you need your own internet radio show. Podcasting is the powerful combination of easy-access audio production and global web publishing, with a little help from the syndication capabilities of RSS. O'Reilly's new book is a manual to the creation and "broadcast" of audio shows using the web.

Although in the Hacks series, and compiled at speed, this is a surprisingly comprehensive work. Rather than focus on the technical aspects of podcasting, it takes a wide view and attempts to cover the entire process from concept to delivery.

The first chapter introduces podcasts and takes a new listener through the process of locating and playing them. Linuxheads may become irritated at the focus on iTunes, then soothed by the Perl listings for a command-line podcatcher and the re-broadcaster which generates RSS directly. The hacks here go as far as listening to podcasts on a PDA, or even the Playstation Portable.

Chapters two and three is where this book really begins to shine. There is a whole mini manual to creating broadcast-quality journalism, from breaking down the responsibilities of the producer, writer, engineer and host, to choosing appropriate hardware and getting the best performance from it. Reducing noise, training your voice to sound natural when recorded... much of the technical content here is studio wisdom for the enthusiastic geek with an IT background but no experience in audio production, and it is pitched at an appropriate level. MP3 encoding is covered too; most users of LAME will not have considered its low- and high-pass filters and how these could be of use.

More than sixty pages are devoted to considering particular show formats, from a regular news, sports or politics show right down to a "beercast", which is almost certainly what you imagine it to be. One interesting idea is to record commentary to films, so the listener can simply synchronise their DVD playback with your programme. A timely warning is included against allowing the film soundtrack to leak into your commentary and thus infringe copyright. A later section covers how to obtain permission to play commercial music legally on your show.

One aspect that will be no doubt overlooked by many eager podcasters is that of publicity. Chapter seven covers this, with a rundown of podcast directories and some less obvious marketing ideas. Funding and connecting with the wider podcasting community are given a little space.

The last essential read is chapter eight, which covers audio editing. This ranges from choosing appropriate software to tweaking the EQ and using plugins to produce an ideal sound. Although this section does cover the free audio editor Audacity, it could have used an extra hack with more detail on this, probably the most popular piece of software used in podcasting production.
This book is well positioned as a fascinating and comprehensive guide to something which is, after all, an emerging phenomenon. It would make a great present for someone who is considering podcasting but doesn't know exactly how to start. Of course not all the hacks will be of interest to everybody, but it's hard to imagine a prospective podcaster who wouldn't find *Podcasting Hacks* of value.

Asterisk

Jared Smith, Jim Van Meggelen and Leif Madsen

O'Reilly and Associates
ISBN 0-596-00962-3
404pp.
£ 28.50

reviewed by Greg Matthews

Asterisk is an open source telephone exchange system or PBX. Unless you work in this field you can be forgiven for not knowing that this software is whipping up a storm of interest in the telecommunications community. For a long time, telephony has been a closed shop for independent developers, the market dominated by monopolistic players. But with the long touted, voice over IP (VoIP) finally finding a toehold and DSL delivering cheap bandwidth, the time is ripe for an open solution.

I have to admit that I only had the vaguest idea of what Asterisk was and what it could do before I received this book. In my line of work it pays to be conservative about technology that "just works". However, even I will admit that traditional telephony just doesn't cut it as a modern communication medium. The closed attitude of the big telecoms industries has stifled innovation and hindered equipment and standards compatibility, but all that is about to change. Asterisk, along with low cost interface cards, such as those from Digium, can do everything an expensive PBX system can do and more.

The authors are Asterisk users and all have been greatly involved in the documentation of the project at
`http://www.asteriskdocs.org/`
It's impossible to ignore the enthusiasm they have for this software which certainly helps to keep the subject matter interesting. This book is aimed at a technical audience, the early chapters walk through the installation and configuration of the software and the emphasis is on familiarising yourself with Asterisk on commodity hardware like an old PC. After that, the information starts to get more interesting and in-depth as the authors begin to explain the concepts of dialplans and telephony protocols.

The beauty of Asterisk is that it can be slotted into an existing PBX system in lots of different ways which makes it easy to add services to an old system and migrate users onto a new and featureful system one step at a time. This makes it extremely attractive to ICT staff wanting to upgrade an old private exchange but not wanting to risk downtime. Examples of this sort of configuration are given in the text.

So what exactly will Asterisk do? Well, anything you can think of really. It will provide VoIP, voice-mail, interactive services, you can get it to route calls to you wherever you are, read your e-mail to you, the possibilities are literally only limited by your imagination and technical ability. This could very possibly be the next big application for Linux and BSD. If its popularity continues unabated, it will be a major driver and enabler for standards compliance and interoperability between vendors.

It is difficult to criticise the technical merit of this book, though the authors do point out that they cover only a small percentage of the subject. It is written to quickly get you up to speed with the technology rather than provide an in depth reference guide to the software. The North American slant is a bit annoying, a little research would have provided equivalent information for readers on other continents.

A significant problem with this sort of book is that the technology is moving so fast, version 1.2 is already available for download, and the more popular it gets the faster it will develop. But the authors of this book are stout hearted – they even devote a chapter to predicting the future, which as we all know, is a brave thing to do. That said, they are sensibly sceptical

about technologies like voice recognition, and they are right to worry about a backlash from the telecoms industry (think deprioritisation of unapproved VoIP traffic). The book is littered with criticisms of telecoms providers which may alienate people already working in the industry but it is good to see such enthusiasm for the philosophy of free software and the empowerment that it will bring to ICT workers.
There is a strong business case for switching the company PBX to Asterisk and there are already commercial vendors and support for such systems in the UK. It should also appeal to hackers who want to replace their home phones with cool technology. This book will get you started, and I fear, once you've started you'll be hooked.

Digital Identity

Phillip J Windley

O'Reilly and Associates
ISBN 0-596-00878-3
254pp.
£ 24.95

reviewed by Greg Matthews

Identity is a hot topic right now. Sociologists are debating the meaning of individual identity, politicians are arguing over ID cards. What is perhaps less well known are the big issues being tackled by the technical departments of corporations and organisations, surrounding the concept of digital identity.

It seems fairly obvious that when I log into my computer in the morning, my identity is checked against a directory, authentication takes place and then I am authorised to perform most of my day to day activities on this system. On closer inspection, it turns out I have multiple identities existing on various directories of many different kinds and that's just the organisation where I work. I have an identity stored in OpenLDAP, another in eDirectory, several more for various parts of our internal and external web space, a few more for various corporate systems: I could go on. These various identities are used for different purposes and contain greater or lesser amounts of overlap in the information that they contain and optionally make available. Most of the examples above are for authentication or authorisation or both (although they needn't be – I also have a "white pages" identity) and each can have completely separate authentication tokens. This can be a nightmare for the user to remember and manage all those passwords. It can also become very difficult to maintain and synchronise all these various repositories of information.

This is only the tip of the proverbial iceberg. Identity verification is required for just about every digital transaction. For instance, I can go to any newsagent in the country and buy my newspaper completely anonymously, provided I can pay in cash. However, if I want to access funds from my bank, my identity must be verified, by my PIN or by the signature on my cheque guarantee card or credit card. The same is true for hosts on a network, businesses must be determine who has access to which applications or data and this can only be done with accurate identity information. Traditionally this is done by building a wall around systems and keeping very tight control on the flow of information into and out of the organisation using a firewall.

Ideally, I'd like to sign on to my computer in the morning and have access to all relevant systems needed to do my job without having to remember another pass phrase. Some people will immediately think of "single sign-on" (SSO) and groan inwardly. In the past, SSO solutions have depended on a single vault of information with all systems relying on this vault for resolving identities. However, such systems tended to lock customers into a limited set of technologies. More modern thinking on digital identity is based on the concept of trust. For instance, I may log on to an airline website and book tickets to New York, The airline may recommend a rental car from a particular company. When I click on the link the rental company will receive an assurance of my identity (an authentication assertion in the lingo) allowing me to use the services of the site without another lengthy sign on. Here, two different companies can set up a trust relationship and their users reap the benefit.

Phillip Windley is well placed to speak authoritatively on these issues – he was the CTO for iMall.com which provides e-commerce services where identity issues are paramount. In 2000 he became the CIO for the state of Utah helping to develop e-government systems. He is currently an associate professor of Brigham Young University. This book is a timely and informative introduction to the burning issue of digital identity. Windley's writing demonstrates his strong grasp of this difficult subject. He introduces each concept and defines it carefully in technical terms. Without this attention to detail, chapters on trust and privacy would be too woolly to be useful. This careful approach means the text doesn't descend into stultifying technical language or impenetrable management-speak. It is clearly laid out and the sections are short and to the point.

This book is not a technical book. It will not tell you how to install and configure an identity management architecture (IMA). In fact, it's almost impossible to find a reference to an existing product. This makes it all the more interesting as the subject is in turns, philosophical (what do we mean by "privacy"?) and pragmatic (how is it implemented?). Each chapter is peppered with relevant examples, many from Windley's personal experience, many from the banking world where issues of digital identity have been paramount for a long time. When discussing related technologies, he is quick to point out interoperability problems between standards and the fact that implementing an IMA is politically challenging to say the least.

I expected the book to be worthy but dry, this was not helped by the picture on the cover of a couple of girls attending what looks like the most boring fancy dress party ever. I found instead, that the subject was interesting and the text informative. Some of the diagrams did little to enlighten the text and had inadequate explanation. Those looking for technical book with code examples will find little to interest them, instead this is a thorough review at the architectural level of the technology required to implement identity management. Highly recommended.

MySQL in a Nutshell

Russell Dyer

O'Reilly and Associates
ISBN 0-596-00789-2
348pp.
£ 28.50

reviewed by Joel Smith

O'Reilly's Nutshell series are subtitled *A Desktop Quick Reference* and are designed to allow you to drill down to the information you need when you need it.

As with all the Nutshell series, this new addition to the stable provides a reference listing of SQL statements, functions etc which are used within MySQL, and provides them in sections grouped by type. The structure gives a quick listing of the command syntax, with more detailed explanation if you read on.

In addition to this basic reference, there are guides to the API functions for Perl (via the Perl DBI module), PHP and C. Again, these are comprehensive and useful as a reference.

These are what you will buy this book for, and they do the job very well. This book will sit by your machine to be easily thumbed through. There are additional sections at the beginning of the book covering an introduction to MySQL, a chapter on installing MySQL for the various platforms, and a quick basic tutorial.

The section on installation feels much like padding - you would be better off reading the documentation that comes with the distribution, but then you don't need to read this section. The tutorial is a useful pointer to get you going.

As I said before, this book will be bought as a reference, and it does that job very well. There are other books which cover issues in more depth, but it is always handy to have a Nutshell book waiting by your keyboard for to be *A Desktop Quick Reference.*

Mac OS X Tiger for Unix Geeks

Brian Jepson and Ernest E Rothman

O'Reilly and Associates
ISBN 0-596-00912-7
415pp.
£ 24.95

reviewed by Joel Smith

I have to confess it. I am a Unix Geek. I have also used Mac OS X since its early beta days, and I really enjoyed this book. I guess I must fall into the target audience of Unix developers, web developers who spend most of their time in `~/public_html` over an ssh connection, and experienced Unix hackers (their definition of the target audience).

This book is actually the third edition, although originally it was *Mac OS X for Unix Geeks*, with the second being *Mac OS X Panther for Unix Geeks.* It has been substantially revised and expanded to cover Tiger, and goes into a glorious amount of geeky detail.

The first section of the book is aimed at helping the reader map their existing Unix knowledge into the framework of OS X. This section also includes information on many of the new features of Tiger, such as the Searchlight search engine, together with methods of accessing these features from within a terminal session. After all, this book is aimed at Unix geeks!

The next section covers building applications, compiling source code, and the details of libraries and headers, and porting issues. This explains why there are the differences between OSX and Unix when porting applications, and I finally understand it. Mac OSX is generally well catered for these days when compiling applications from source, but it is invaluable information for those times when

```
./configure
make
make install
```

does not give you the simple result you are looking for.

The book rounds off with sections on package management options, using Mac OS X as a server (including system management tools) and useful reference information. There is even a mini GUI tutorial for those Unix geeks who have stumbled into the Mac environment and need a bit of help navigating the Mac bits on top.

If you fall into the target audience for this book, you will find it very useful and easy reading. It is worth it for the little nuggets you will pick up which can make your life that much easier. The tendency with Macs is to just get on with it. This book gets you into the internals, but from the Unix side, rather than esoteric Macisms. Enjoy it.

Classic Shell Programming

Arnold Robbins and Nelson H F Beebe

O'Reilly and Associates
ISBN 0-596-00595-4
558pp.
£ 24.95

reviewed by Jan Wysocki

This book is comprehensive and reassuringly dense (over 500 pages). It not only teaches you shell scripting, but also champions the early Unix "Software Tools" philosophy. It has the feel of a 'golden age' O'Reilly book. Working as a Unix Systems Administrator, I've been writing and mending shell scripts for over 15 years. I like to have Bruce Blinn's slender "Portable Shell Programming" (Prentice Hall, ISBN 0-13-451494-7) on my desk, but this will become another companion.

As a lot of my administrative scripts reach out across the network, I immediately looked for comparable material in the book. Carefully hidden in Chapter 8 "Production Scripts", there's a dissection of a script to automate software builds across a heterogeneous network of servers. Nothing like my scripts, but deeply instructive. One you've started looking, you can browse endlessly in this book.

It's comprehensive. So it not only teaches you how to deal with, for example, localisation or filesystem searches, but also includes a chapter to get you writing awk scripts. The primary theme is shell scripting, with the assumption that you want to create portable, Posix compliant programs. The authors' approach avoids tedious discussion of shell idiosyncrasies and gets on with the task in as straightforward a way as possible. When you want to know about shell differences, there is a handy chapter on portability issues and extensions.

The style of this book is relaxed but exact. Some of the explanations can be very basic, but then this is far from a "Nutshell" book. They get quite wordy from time to time, but if you want a book to read on the train, it's exactly the sort of style that I need.

This book is full of nuggets of advice and interesting asides, far too many to list. I think the main shortcoming of this book is a lack of organisation. I was disappointed by the summaries at the end of each chapter, they're just brief reminders of the topics covered. I'd prefer heavier digests of the chapter material. I'm sufficiently enthusiastic to think that this book should be considerably bigger, or divided into further volumes. I'd like to see more material on scripting across network connections, and interactions with the multiple layers built upon Unix, but perhaps it's time for a rethink. This book maybe shows the limits of shell programming, what once was the all powerful tool for the Systems Administrator can sometimes be inadequate faced with some modern applications. Perhaps we need some additions to the toolbox, but in the meantime this is a good reference for the scripting tools we have.

Have I learned anything? Well for a start I think they've convinced me that I really should stop using `echo` and stick with `printf` even though their book is liberally sprinkled with `echo`s. I now know some words with 6 or 7 consecutive vowels and I've learned to appreciate a few more command options. I'm not sure if I'm going to use awk for floating point arithmetic though, I'm happy with `dc` and they don't consider it.

If you want to take the trouble to learn how to write robust, portable shell scripts, or you want to improve your skills, I thoroughly recommend this book. I should add that it's the programming style that's "classic", not the shells. You'll find material on `zsh` if that's what you want!

From the UKUUG Diary

The UKUUG maintains a web diary of future events of interest at
`http://www.ukuug.org/diary/`
The following events are a small selection of those currently listed.

Chaos Communication Congress 2005

27th December 2005: Berlin, Germany

The 22nd Chaos Communication Congress (22C3) is a four-day conference on technology, society and utopia. The Congress offers lectures and workshops on a multitude of topics including (but not limited to) information technology, IT-security, internet, cryptography and generally a critical-creative attitude towards technology and the discussion about the effects of technological advances on society.
`http://events.ccc.de/congress/2005/`

BCS OSSG: An overview of Open Source Licensing

10th January 2006: BCS Central London Office

In this talk organised by the BCS Open Source Specialist Group, Andrew Katz, a solicitor with specialist legal firm Moorcrofts LLP, gives an overview of Open Source Licensing: obligations that licences do, or do not, place on users, distributors and modifiers of OSS.
`http://ossg.bcs.org/events/calendar`

BETT 2006

11th January 2006: Olympia, London

BETT is the world's leading educational ICT event, attracting over 600 educational suppliers and 27000 visitors, and bringing together the global teaching and learning community for four days of innovations and inspirations.
`http://www.bettshow.co.uk/`

II Conferencia Internacional de Software Libre

15th February 2006: Malaga, Spain

"Innovation and Freedom" is the theme of the Second Open Source World conference. This is the 2nd part of the conference, part one being held in Merida, Extramadura Oct 25-26.
`http://www.opensourceworldconference.com/malaga06/en/modules/wiwimod/`

SANE 2006

15th May 2006: Delft, The Netherlands

The SANE 2006 conference offers 3 days of training, followed by a 2-day conference program filled with the latest developments in system administration, network engineering, security and open source software, and practical approaches to the puzzles and problems you wrestle with. You'll also have many opportunities to meet other system administrators and network (security) professionals and chat with peers who share your concerns and interests. The venue for SANE 2006 will be the Aula Congresscentre, located on the campus of the University of Technology in Delft: the city of Delft Blue, the world-famous painter Johannes Vermeer and its historical ties to the Royal House of Orange. But also a lively modern city for funshopping, going out for a great dinner or wandering around to experience its special atmosphere.
`http://www.sane.nl/sane2006/`

Contributors

Elena Blanco works as Content Editor for the JISC OSS Watch project. She has been involved in many different aspects of academic computing provision since 1991. A techie at heart she has many years of hands-on experience in Unix systems administration and network infrastructure support and in recent years has moved into techical project management. Somewhat unusually for a sysadmin she likes to put pen to paper.

Dave Cross runs Magnum Solutions Ltd, an Open Source consultancy based in London. He is a well-known member of the Perl community and a frequent speaker at Perl and Open Source conferences. Since 2002, Dave has been the Perl Mongers Group Co-ordinator for the Perl Foundation. He is the author of "Data Munging with Perl" (Manning, 2001) and a co-author of "Perl Template Toolkit" (O'Reilly, 2003). His company and personal web pages are at
`http://mag-sol.com/`
`http://dave.org.uk/`

Leslie Fletcher works part-time as UKUUG Campaigns Manager, with the brief of improving the visibility and credibility of UKUUG and its mission in key arenas - business, politics, public service, education. His main first-hand involvement with Open Source is as chair of governors at Parrs Wood Technology College in South Manchester. He also has some experience in IT management, having been head of the Department of Computer and Mathematical Sciences at Salford University for five years.

Gavin Inglis is a Technnical Infrastructure manager for the EDINA project at Edinburgh University's Data Library.

Michael Josefsson is a research engineer at Linköping University, Linköping, Sweden.

Greg Matthews is a Senior Unix and Linux administrator for the Natural Environment Research Council.

Ray Miller is a director of UKUUG and Chairman of UKUUG Council. He works as a Unix Systems Programmer at the University of Oxford, where he manages the Systems Development and Support team in the University's Computing Services.

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.

Sebastian Rahtz has been involved with free and open source software since the late 1980's as a developer in the community around the TeX typesetting system, on which he has published widely. He has been maintaining an open source TeX distribution for the last eight years, and a variety of TeX-related packages, for which he won the UKUUG Open Source Award in 2000. He is an active member of the XML and XSLT communities, and one of the technical leads for the Text Encoding Initiative. He is now Information Manager at Oxford University Computing Services, and deputy leader of its Information and Support group.

Joel Smith offers consulting services through Dales IT Ltd.

Roger Whittaker is editor of the UKUUG newsletter, and has been a member of UKUUG Council since Septermber 2000.

Jan Wysocki started out as a microbiologist, and has made use of computers in scientific work since 1964, but got into systems administration on a Primos mini-computer about 20 years ago. After a brief flirtation with Unix administration on a Mostek 68000, he grew to love Unix as a platform for the Poplog environment. After a few years programming AMT DAPs as Sun 3 attached processors, he has been administering a variety of Unixes in various academic and commercial environments. He's currently working at BT's Adastral Park, integrating software on the BT Internet Reference model.

Credits

The article by Michael Josefsson was originally published by Daemon News:
http://ezine.daemonnews.org/200107/adventure.html

Michael Josefsson's article also appears as the first article in the compilation "BSD Success Stories", a pamphlet put together by Dru Lavigne for O'Reilly:
http://linux.oreilly.com/news/bsd_ss.pdf

The article by Dave Cross was first published in Linux Format and is reproduced here by kind permission of the author.
The article by Elena Blanco was first published by OSS Watch:
http://www.oss-watch.ac.uk/resources/ilovemymac.xml

Contacts

Ray Miller
Council Chairman; Events; Newsletter
Oxford
Tel: 01865 283284
`ray.miller@ukuug.org`

Mike Banahan
Ely
`mike.banahan@ukuug.org`

Sam Smith
UKUUG Treasurer; Website
Manchester
`sam.smith@ukuug.org`

Alain Williams
Watford
`alain.williams@ukuug.org`

Roger Whittaker
Newsletter
London
`roger.whittaker@ukuug.org`

John M Collins
Welwyn Garden City
`john.collins@ukuug.org`

Dean Wilson
London
`dean.wilson@ukuug.org`

Newsletter
`newsletter@ukuug.org`

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