news@UK

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

Volume 18, Number 2 **ISSN 0965-9412** **June 2009**

Contents

News from the Secretariat

Jane Morrison

I am very pleased to note that the Spring Conference held between the 24th and March in London was very well attended. In fact numbers really did surpass our expectations after all the gloomy reports about the recession at the beginning of the year.

Since then we have been really busy organising a full schedule of events for the rest of 2009, all of which are detailed on our web site:

- OpenTech 2009: Saturday 4th July
- Summer Conference and Tutorials: 7th–9th August, Birmingham
- Request Tracker Tutorial: 11th August, London
- EuroBSDCon 2009: September 18th–20th, Cambridge
- Advanced DNS Administration using BIND9: 13th October, London
- Perl Tutorials: 3 separate days 24th, 25th and 26th November, London

The tutorials listed above are being organised through collaboration with Josette Garcia at O'Reilly.

The next Spring Conference will be held in March 2010 (Manchester Conference Centre). A call for papers will appear shortly.

The UKUUG Annual General Meeting will be held this year on Thursday 24th September. Further details, Agendas etc. will be sent to you automatically during August.

Perhaps you are interested in joining Council, if so please let me know.

Don't forget that the Inland Revenue allows UKUUG members to claim their subscription amounts for tax allowance purposes. Further details are on our web site.

The next Newsletter will appear in September and has a copy date of: 21st August. Any interesting articles from members will be very welcome – all submissions should be sent to
`newsletter@ukuug.org`

Chairman's report

Paul Waring

Recent and Upcoming Conferences

Our annual Spring conference went with a bang, with a packed programme covering topics across the board of securiy and system administration and over a hundred delegates, not to mention dinner aboard HMS Belfast!

Preparations for next year's Spring conference are already well underway, and the call for papers should be hitting an inbox near you in the coming weeks.

Before that though we have the Summer 2009 conference in Birmingham, covering a wide array of topics from a variety of projects. I look forward to seeing many of you there for what is shaping up to be an excellent weekend of talks and tutorials.

Get Involved

UKUUG exists to serve its members and we are always on the lookout for people who are keen to get involved in any capacity, whether that be through volunteering to help with organising events, writing newsletter articles, or entirely new activities which haven't been tried before. If you have any ideas for future events or would like to get involved with one of our working groups, please do let us know.

Open Tech 2009

Brought to you by UKUUG and friends. Sponsored by 4iP

Open Tech 2009 will take place on Saturday 4th July 2009 between 11am and 6pm at ULU, Malet Street, London WC1E 7HY. Doors will open at 10 am and the bar will stay open till 10 pm.

The cost is £5 at the door: entry for students is free. Because of likely heavy demand, you are *strongly advised* to pre-register online.

There will be 33 talks across 3 sessions covering 7 hours, as well as plenty of time afterwards to talk in the bar about sessions which challenge, inspire, instruct and suggest collaborative activities. The last two events we have run sold out in advance, so you are strongly advised to pre-register.

This year's line up features…

- Two Cultures from Bill Thompson
- Bad Science from Ben Goldacre
- Peace & War
- Making things happen, from those who do
- Web of Power - what's next for Politicians?
- The Guardian and Ian Tomlinson Story
- Ways our Internet Laws are Broken

The full schedule is at:
`http://www.ukuug.org/events/opentech2009/`

OpenTech is organised by volunteers and we are now looking for volunteers to help out on the day. In return for free early entry and our eternal gratitude, we're in need of a few people to show up a bit earlier and help us set the venue up.

If you're interested, or have random other questions, email us on
`opentech@ukuug.org`

Final programme may be subject to alteration. OpenTech is a not for profit event open to everyone so please help spread the word online and offline.

EuroPython 2009 – 28th June to 4th July 2009, Birmingham, UK

EuroPython is the official community conference for Python Programmers across Europe. This year it is being held in the Birmingham Conservatoire, where previous UK Python events and some recent UKUUG events have been held. We are expecting 300-400 delegates from across Europe. However, there will be many familiar faces as several of the conference organisers and speakers are UKUUG members.

The first two days, Sunday 28th June 2009 and Monday 29th June 2009, are tutorial days, including an introductory tutorial for those who have never programmed in Python before.

The main conference, from the 30th June to 2nd July, has talks from the leading Python programmers from across Europe. Speakers include Professor Sir Tony Hoare, Cory Doctorow, Jim Hugunin, Bruce Eckel, Simon Willison, Christian Tismer, Emily Bache, Stani Michiels and Michael Foord. Tutorial and talk abstracts are available on our website:
`http://www.europython.eu`

The final two days, the 3rd and 4th of July, are 'sprint' days, where delegates splint into groups for collaborative computer programming, improving open source Python software.

Altogether, the conference consists of over one hundred talks, tutorials, plenary sessions and social events. To come for the whole week costs £290, but our booking form allows you to book for the parts you want. Concessionary rates are available for over 60s, full-time students, unwaged and nurses. There are more details at
`http://www.europython.eu/registration/`

FFII announcement: Petition to stop software patents

We have received the following announcement from Luisa Zielinski of FFII.

Petition against software patents

We are contacting you as we believe that the following topic might be of interest to your organisation. We would like to inform you of a new attempt to legalise software patents and ask for your support against it by signing our petition at
`http://stopsoftwarepatents.eu/`

In October 2008 the President of the European Patent Office issued a Referral to its Enlarged Board of Appeal (EBoA) concerning the question of software patents in Europe. In the absence of European legislative initiatives, the EBoA's conclusion on this matter is likely to have the same effect as a software patent directive. However, since the EBoA – rather than the European Parliament – is now tasked with this issue, the decision will be made on a purely legal basis, unaccompanied by more extensive political or economic debate. The FFII is currently working on several different strategies to communicate its concerns to the EboA; the deadline to do so is at the end of April. This is why your organisation's signature is needed to put an end to software patents right now.

It would be extremely helpful if the members of your organisation could also aid us in our efforts and sign the petition individually. After signing the petition, you will receive a link which you could then forward to them. The adapted link will allow us to find out which organisation individuals who signed the petition belong to. If you prefer to withhold that information, you can alternatively send the following link to the members of your organisation:
`http://stopsoftwarepatents.eu`

LugRadio Live 2009 announcement

We have received the following announcement for LugRadio Live 2009.

LugRadio Live 2009: Newhampton Arts Centre, Wolverhampton, West Midlands, United Kingdom

Greetings from LugRadio Live UK

On Saturday 24th October 2009, several hundred Linux users from all over the world will be descending on Wolverhampton for LugRadio Live UK 2009.

LugRadio Live is the largest British community driven event for the members of the worldwide Linux community of users and developers, and attracts a wide range of interested individuals and organisations. Speakers at previous events included Chris DiBona from Google, Mark Shuttleworth of Canonical, Michael Sparks of the BBC, Becky Hogge of the Open Rights Group and Gervase Markham of the Mozilla Foundation.

The event has developed a strong reputation for providing a range of topics about free software, Open Source, digital rights, technology and much much more, with a compelling list of speakers, and exhibitors, wrapping it all in a unique, fun and social event which is open to everyone. The event is often described as combining the atmosphere of a rock concert and a computer conference. Now in its 5th year LugRadio Live has firmly established itself as a 'must go to' event in the Linux community calendar and ranks amongst as one the foremost events of the year.

LugRadio Live was born out of LugRadio which is a podcast that takes a relaxed, humorous look at Linux and open source. The show was formed by several members of the Wolverhampton Linux users group, Jono Bacon, Stuart Langridge, Stephen Parkes and Matthew Revell, with the first episode being made available for download on 26 Feb 2004 (this was before the word 'podcast' was invented!). While there have been some changes in personnel and guest presenters over the years, the four regular presenters have always been members of the WolvesLUG.

We hope to see you there.

Spring Conference report

Roger Whittaker

The UKUUG Spring Conference was held between 24th and 26th March at the Park Crescent Conference Centre in central London. London. The first day (Tuesday 24th) was devoted to a tutorial on Kerberos given by Simon Wilkinson of the University of Edinburgh. I did not attend the tutorial, but heard good reports of it from those who did.

The conference proper took place over the next two days. There were about 110 attendees. There was a single track, and all the talks took place in a good-sized and comfortable lecture room / theatre.

The foyer outside the theatre was somewhat cramped and a little too small for its dual purpose as an exhibition area and lunch and coffee area. But it was big enough to hold table-top stands from Sun, Google, Novell and Bytemark. The usual O'Reilly book stall was also in attendance.

Wednesday opened with a talk by Barry Scott of Centrify on integrating Unix and Linux systems into Active Directory. This talk struck me as a little too close to the line that divides technical talks from vendor sales pitches.

Bill Quinn then spoke on LPI certification, and described some of the changes that have taken place recently (revisions at levels 1 and 2, and new certifications on a modular plan at level 3).

John Collins and John Pinner jointly presented GNUspool: this was an interesting introduction and demonstration of a print spooler whose capabilities go well beyond those of CUPS, and which is now an official GNU project.

Andrew Findlay then spoke about Access Control Policies for LDAP: a very clearly presented talk.

The afternoon began with a systems monitoring theme, with a general survey by Tom De Cooman, followed by a talk on Zenoss by Jane Curry, in which she explained her reasons for selecting Zenoss from a wide field of similar tools, and went into detail on its capabilities.

There followed an interesting talk and by Chris Proctor, entitled "LVM, sysfs and me". Chris very entertainingly described the empirical learning process through which he had gained an understanding of the (largely undocumented) features of sysfs, as well as some of the scarier aspects of LVM.

The first day of the conference ended with two security talks: first from Darren Moffat of Sun about the OpenSolaris security model, and then from Clive Blackwell of Royal Holloway College. Clive described a methodology for thinking about security threats of all kinds based on looking at systems as composed of a series of levels (physical, logical and social, with possible sub-divisions of these), and where different security threats fit into this model. The concepts presented were interesting, but I personally felt the talk would have been more illuminating if there had been more real life examples.

In the evening the conference dinner took place on board HMS Belfast. Before dinner was served, there was an interesting talk by a veteran of the Belfast about the history of the ship.

The first two talks on Thursday morning were related to OpenLDAP. Gavin Henry discussed OpenLDAP Replication Strategies. Then Howard Chu presented a talk entitled "OpenLDAP and MySQL: Bridging the Data Model Divide" in which he described how OpenLDAP can be served by a relational database back-end using MySQL. To the surprise of the audience, Howard illustrated some points in his talk with solo violin playing.

Craig Gallen offered an introduction to OpenNMS, and Jos Vos presented a very interesting talk on the use of custom RPMs for system administration purposes, in particular using "trigger" scripts which are executed when a particular RPM package is installed or upgraded.

After lunch, Kris Buytaert gave a wide-ranging talk surveying all the available forms of open source virtualisation, with some thoughts on future developments. Matthias Rechenburg of openQRM then described this open source framework for virtualisation and cloud management, and demonstrated its web front end, creating and destroying machine instances in the cloud before our eyes:
`http://demo.openqrm.com/`

A paper by Matthias Rechenburg based on his conference talk is included elsewhere in this Newsletter.

The day and the conference proper ended with a talk by Alex Howells about network security threats.

Andrew Elias ended the day with an extra talk on Robotics education describing the work of the UK Robotics Education Foundation:
`http://www.ukref.org.uk/`

In general this was a most enjoyable conference, with some very interesting talks and plenty of opportunities for socialising and discussion. Papers and slides are available at:
`http://www.ukuug.org/events/spring2009/slides/`

Press release from the Free Software Pact

We have received the following press release from the Free Software Pact.

London, UK – Friday 22nd May 2009 – The Free Software Pact initiative calls upon UK citizens and MEP candidates to stand up for the principles of a free society by backing free software in the upcoming European Parliament elections on June 4, 2009.

The Free Software Pact is a European initiative to allow candidates for the upcoming European elections to show the voting public that they favour the development and use of free software, and will protect it from threatening EU legislation. It is also a tool for citizens who value free software to educate candidates about its importance and why they should, if elected, protect the European free software community. The European Parliament is the venue for crucial talks concerning free software, including software patents, interoperability and net neutrality. It is therefore vital to show election candidates why they should support, and sign, the Free Software Pact.

Mark Taylor, the coordinator for the Free Software Pact in the UK, said, "The current UK Government is embarrassingly behind the rest of Europe in formulating public policy on the use of free software. Across the rest of the continent we see significant adoption and political support for free software. The Free Software Pact is therefore an ideal way to draw attention to the reform the UK public sector needs and the enormous cost savings yet to be realized. For too long the UK has been dependent on the relationship with proprietary software companies like Microsoft, who are hell-bent on keeping our politicians confused on this matter. If you care about this situation, and the resulting cost to our economy, society and political culture, please contact the MEP candidates in your region and ask them to sign the Free Software Pact."

The Free Software Pact is also supported by Richard M. Stallman, founder and president of the Free Software Foundation, who said, "Big dangers threaten the freedoms of free software in Europe: software patents, digital restrictions management (DRM), bundled sales and treacherous computing…I call on all

European citizens who value free software to join this campaign, contact their candidates and have them sign the Free Software Pact."

A list of UK MEP candidates and their contact details can be found at
`http://www.bond.org.uk/pages/mep-candidate-contact-details.html`

Candidates can support the Free Software Pact by signing a copy of the pact and faxing, mailing or emailing a copy by following the instructions at
`http://www.freesoftwarepact.eu/post/The-Free-Software-Pact`

About The Free Software pact

The Free Software Pact (FSP) is a citizen initiative, launched by Free Software advocacy associations April (France) and Associazione per il software libero (Italy), to coordinate a European scale campaign in favour of free software. The FSP is providing materials and software to any volunteer who contributes to the initiative. More information can be found at their website:
`http://www.freesoftwarepact.eu`

The Birth of UNIX

Peter H Salus

Last column I wrote of the beginnings of the Internet. This time, UNIX. In September I'll return to the net, and in December we will 'celebrate' Linus' 40th birthday. But now . . .

- Are you running OS X?
- Are you running some form of Linux?
- Are you running a BSD?
- Are you running Minix?
- Are you running Solaris?
- Or (even) trying to run Windows?

If you've responded 'yes' to any of these, you owe a debt to UNIX. And UNIX will be 40 this year.

What? UNIX? Nah. I've heard of it. But . . .

No joke. All of the first five are true descendants; and Microsoft employs parts of UNIX systems (like the Berkeley TCP/IP stack) in every one of its Windows variants.

Are you comfy? Let me tell you about it.

In 1964, MIT, General Electric and Bell Labs embarked on an ambitious time-sharing system, Multics (Multiplexed Information and Computing Service). Nearly five years later (in spring 1969), with Multics over-budget and behind schedule, Bell Labs pulled out of the project.

This left several researchers in Murray Hill with nothing to focus on. Several of them had liked aspects of Multics, but hadn't been happy with the size and the complexity of the system. Three of them, Ken Thompson, Rudd Canaday and Dennis Ritchie would gather in front of a whiteboard or around a table and discuss design philosophy. The Labs had Multics running on its GE-645 and after March 1969, Thompson continued to work on it "just for fun".

Doug McIlroy, who was the manager of the group, told me: "When Multics began to work, the very first place it worked was here. Three people could overload it."

But in July Thompson became a temporary bachelor. His wife, Bonnie, took their year-old son to meet his relatives on the West Coast for the month of August. Thompson recalled: "I allocated a week each to the operating system, the shell, the editor, and the assembler, to reproduce itself, and during the month she was gone, it was totally rewritten in a form that looked like an operating system with tools that were sort of known, you know, assembler, editor, shell – if not maintaining itself, right on the verge of maintaining itself, to totally sever the GECOS connection... Yeah, essentially one person for a month." [GECOS was General Electric Comprehensive Operating System]

Steve Bourne, who joined Bell Labs the next year, told me about the cast-off PDP-7 that Ritchie and Thompson used: "the PDP-7 provided only an assembler and a loader. One user at a time could use the computer ... The environment was crude and parts of a single-user UNIX system were soon forthcoming... [The] assembler and rudimentary operating system kernel were written and cross-assembled for the PDP-7 on GECOS. The term UNICS was apparently coined by Peter Neumann, an inveterate punster, in 1970."

(This was a single-user system, obviously an 'emasculated Multics'. Several people told me that Brian Kernighan changed the spelling to UNIX, but Brian said that he hadn't.)

But while there were aspects of UNICS/UNIX that were influenced by Multics, there were also – as Dennis Ritchie said – "profound differences".

"We were a bit oppressed by the big system mentality," he said. "Ken wanted to do something simple. Presumably, as important as anything was the fact that our means were much smaller – we could get only small machines with none of the fancy Multics hardware."

"So UNIX wasn't quite a reaction against Multics ... Multics wasn't there for us any more, but we liked the feel of interactive computing that it offered. Ken had some ideas about how to do a system that he had to work out ... Multics coloured the UNIX approach, but it didn't dominate it."

Ken and Dennis' "toy" system didn't stay that simple very long. By 1971, user commands included `as` (assembler), `cal` (print calendar), `cat` (catenate and print), `chdir` (change working directory), `chmod` (change mode), `chown` (change owner), `cmp` (compare two files), `cp` (copy file), `date`, `dc` (desk calculator), `du` (summarize disk usage), `ed` (editor), and over two dozen others. And, as you know, most of them are still in use.

And, of course, there were system calls and subroutines and special files.

But use of the system was confined to central New Jersey ... until it "leaked out". The first user outside Bell Labs was Neil Groundwater, who had joined New York Telephone in February 1972. The group he was with was "mechanising" the Electronic Switching System, and Neil was sent to Whippany, NJ, to learn about the OS. By a year later, February 1973, there were 16 UNIX installations. And two big innovations.

The first of these was a "new" programming language, C, based on B, which was a "cut-down" version of Martin Richards' BCPL (Basic Combined Computer Language); the other was | – "pipe".

Pipe is a simple concept: a uniform mechanism for connecting the output of one program to the input of another. The Dartmouth Time-Sharing System had communication files, which anticipate pipes, but in a far more specific way. The notion of the general pipe was Doug McIlroy's. The implementation was Ken Thompson's, at McIlroy's insistence ("It was one of the only places where I very nearly exerted managerial control over UNIX", McIlroy said.).

"It's easy to say 'cat into grep into ... ' or 'who into cat into grep' and so on", McIlroy remarked. "It's easy to say and it was clear from the start that it would be something you'd like to say. But there are all these side parameters ... And from time to time I'd say 'How about making something like this?' And one day I came up with a syntax for the shell that went along with piping, and Ken said 'I'm going to do it!' "

He did. And then he had a orgy of rewriting and Thompson changed all the programs in the same night. The next morning there were one-liners. Thompson also invented | – because he hated McIlroy's "ugly syntax".

This was the real beginning of the power of UNIX – not from the individual programs, but from the relationships among programs.

UNIX now had a language of its own and a philosophy:

- Write programs that do one thing and do it well.

- Write programs to work together.
- Write programs that handle text streams, because that is a universal interface.

but it had yet to acquire an audience. That began in October 1973.

ACM held its Symposium on Operating Systems Principles (SOSP) in the auditorium at IBM's new T.J. Watson Research Center in Yorktown Heights, NY. Ken and Dennis submitted a paper and on a beautiful autumn day drove up the Hudson Valley to deliver it. Thompson actually gave the paper. There were about 200 in the audience. It was a smash hit.

Over the next six months, the number of UNIX installations tripled. When the paper was published in the July 1974 issue of CACM the response was overwhelming.

'So what'? you ask.

One in the audience was Professor Robert Fabry of the University of California at Berkeley. He got the University to buy a DEC PDP-11/45, wrote to Ken, obtained a tape, and had it installed. By January 1974, the seed of Berkeley UNIX had been planted.

Mel Ferentz at Brooklyn College got a tape.

So did Lou Katz at Columbia University.

And Lew Law at Harvard.

And Richard Langridge at Princeton.

And George Coulouris at Queen Mary College in London.

Mel and Lou announced a users' meeting for May 15, 1974. About two dozen people from a dozen institutions showed up. And it was two months before CACM published the article!

And people all over wrote tools and toys for UNIX.

The first Berkeley tape (1977) was a Pascal system and the vi editor for the PDP-11. 2BSD came the next year. 3BSD, the first Berkeley release for the DEC VAX, was late 1979. Late the next year 4BSD was released and became quite popular, largely because it was the only UNIX that ran on the VAX 11/750. Listing all the Berkeley releases would be excessive. But it is important to note that 4.2BSD was what Bill Joy took to Sun Microsystems in 1982 to become SunOS. And in 1992, SunOS was succeeded by Solaris.

At the same time, Andrew S. Tanenbaum, a UNIX user at the Free University in Amsterdam, decided to write a system incorporating no AT&T/Bell code which could run on the 'new' PC with an Intel chip. By 1986 you could run Minix.

A few years later, Linus Torvalds, a college student in Helsinki who was playing with Minix, began writing another clone. "Just for fun", he said. Beginning as a terminal emulator, by 1991 Linux was available via ftp. By 1992, Both SuSE and Yggdrasil Linux were available; in 1994 Red Hat and Linux Pro were established.

UNIX to Minix to Linux.

Macintosh OS X appeared in 1999. OS X was derived from BSD, more specifically FreeBSD and NetBSD. It has undergone a great deal of development in the past decade, but its roots are quite clear. If you've got an Apple mini or a laptop, open a terminal window, enter `man cat`. The text is nearly identical to that originated by Dennis Ritchie on November 3, 1971; and the page tells you it's the "3rd Berkeley Distribution" of "May 2, 1995".

Oh, yes. If you encounter a Microsoft system, it's running the Berkeley TCP/IP stack.

And what about C? Well, it's still going strong. Among its children and grandchildren are: awk, csh, C++, C#, Objective-C, BitC, D, Java, JavaScript, Limbo, Perl, PHP, Python, and many, many more.

Decades later, it still gets hot in central New Jersey in the summer, but there are fewer mosquitoes. Doug McIlroy is back at Dartmouth; Ken Thompson is at Google in California; Rudd Canaday is in New Mexico; Brian Kernighan and Dennis Ritchie are still in New Jersey, though Brian is at Princeton.

Happy 40th Birthday, UNIX!

* * *

Remember The SCO Group? Well, something happened in May.

First, in the bankruptcy case in Delaware, the U.S. Trustee's Office appears to have lost patience and has filed that court force SCOG into liquidation. The hearing on this will be June 12.

Second, the next day (May 6), the Court of Appeals in Denver heard SCOG's appeal of Judge Kimball's ruling of August 2007. From reports, things do not look good for the chaps from Salt Lake City.

Instant Cloud Computing with openQRM

Matthias Rechenburg

What is Cloud Computing? Approximately since the beginning of 2008 the term "Cloud Computing" came up as new hype in the IT world directly after "Virtualisation". Since then this term is used for different new and also well-known technologies for managing large data centres and IT infrastructures. Some use it as a synonym for "Software as a service" (SAAS) coming from the Web 2.0 evolution, some others connect the term "Cloud Computing" with automated provisioning of a large number of virtual machines and "servers on demand". In general the definition of the term "Cloud Computing" leaves it open but defines in more generic way:

> *Cloud computing is Internet ("cloud") based development and use of computer technology ("computing"). It is a business information management style of computing in which typically real-time scalable[4] resources are provided "as a service" over the Internet to users who need not have knowledge of, expertise in, or control over the technology infrastructure ("in the cloud") that supports them. (Wikipedia → Cloud Computing.)*

Even more, both aspects of Cloud Computing, SAAS and Automated Provisioning, are coexisting in a kind of symbiotic relationship. SAAS means to provide a Service via a network (Inter- or Intranet) to its customers on demand. Most common for this case is that the applications and services are deployed to a separated virtual machine. This isolation of the application in its own VM results in better flexibility and security for the SAAS provider. The requirement to deploy a huge number of VMs (one per application instance) needs automatism which is solved through Automated Provisioning.

Cloud Computing is not a new technology but a new name for a combination of two or more already known technologies like HPC and HA Clustering, Grid-Computing, Utility-Computing, Distributed-Computing etc. They are all there to manage large IT infrastructures used for different purposes in an automated way.

The role of the open-source community especially for Cloud Computing

Modern Clouds mostly consists of already existing, well-known and often open-source components. For every aspect of a Cloud environment a different set of utilities is used e.g. Puppet for automated configuration management, Nagios for system and service monitoring, one or more virtualisation technologies, Linux-HA for high-availability etc. Those utilities are already used in large production environments and are accepted by system administrators and IT managers. It would make no sense to re-write all those tools from the scratch just because of a new name (Cloud Computing) for known methods of deploying and managing large server environments. For that reasons Clouds are mainly a set of "loosely connected" tools. The challenge for modern Cloud Computing is the integration of those "loosely connected tools" into a single management User Interface. It can only be archived via a plug-ability which combines all the separated utilities to benefit from their cooperation. This is exactly the goal of the openQRM data centre management platform.

The openQRM data-center management framework

Modern data centres are often a very complex environment with many different aspects require to perfectly work together to keep all services up and running. Integrating all those facets of the IT infrastructure is a huge task which most likely costs a lot of development and QA time and on the other hand would balloon a single, standalone application.

OpenQRM is following a different approach. Instead of implementing all required features into a main application server basically all mechanism are "out-sourced" to plugins. In general the openQRM-server itself provides only the back-end and framework to manage plugins. Via a well defined API openQRM provides lots of integration options and "hooks" for plugins to execute certain commands in specified situations in the data centre. This enhanced plugability also makes it very easy to integrate other already existing or new third party utilities.

The concept of openQRM is then to break the different aspects of a data centre into small, generic and manageable objects. Those objects are purely generic like a Lego brick and openQRM can puzzle and combine them in every way for the system administrator.

One of the first questions which came up about this approach of the separation into manageable objects is: *What is Linux?* A Linux server as we know it today consist of a kernel file, an initial ram disk file (initrd), some kernel module files and a root filesystem (/) which also consist of "only" files. So basically a Linux server (the software-part of it) is just a bunch of files which are then loaded onto a physical server (or virtual machine).

The most common way to deploy a server today is still to install all those files on the servers hard disk. Most likely this deployment method is causing troubles because as experiences hard disks are the weakest part of our current hardware. At some point the hard disk, or even hard disk RAID, may break and stop working. Without a good backup solution this situation can be harmful to any company. OpenQRM solves this situation via a full separation between hardware (the physical systems) and software (the server providing services). With this separation physical hardware becomes replaceable at any time.

openQRM is using an image based deployment mechanism to rapidly inject server images to physical (or virtual) servers via netbooting. In the openQRM environment the servers are located and stored on dedicated storage-devices (NAS/SAN/FC/iSCSI/AoE/*) as "server images". Those server-images are basically "just" the root file systems for the deployment the software-part of the provided service in the data-center). Differing from automatic installation deployment, this method directly connects server root-file systems to starting servers after their PXE request.

System administrators benefit from this architecture in several ways. Deploying new servers gets very easy and fast because they simply can be started up immediately with "clones" (snapshots) of existing server-images. The snapshotting features of modern storage devices are also very useful to keep different version of the same server with the option to rollback and roll-forward at any time.

OpenQRM is also able to transfer and install server images during deployment time e.g. a server can be assigned to start an empty server images located on an iSCSI LUN which is being populated on boot-up with a server template from an NFS location. This fully pluggable mechanism can be also used to create server-images from already existing servers by "grabbing" the servers local disk content and transfer it to a Storage location. Also deployment to local hard disk is supported. In this case (since we experienced that local hard disk may break at some point) the server image is still located safe on the storage-server and can be redeployed to new hardware within minutes.

With its integrated and plug-able storage-management openQRM automatically takes care to authorise server-images selected for deployment to the actual resource (the physical or virtual hardware). This security mechanism makes sure that each server (-hardware) is only able to access its dedicated root file system.

Plug-able virtualization types

Not only the storage types but also the virtualisation types are fully plug-able in openQRM. Via its open plugin API openQRM integrates with VMware-ESX, Vmware-Server, Xen, KVM, Citrix XenServer and Linux-VServer. Adding support for further virtualisation technologies like Virtualbox, openVZ and others is on the future road map. To be able to seamlessly handle all those different kinds of virtual machines openQRM puts a layer on top of the virtualisation methods to unify their management. In openQRM virtual machines are simply net-booting into the openQRM management environment in the same way as physical systems. Continuing with the full separation between hardware and software, meaning on one side physical and also virtual machines (because the VMs are running on the Hypervisor which is running on the bare metal) and on the other side the software layer, the server-images located on a safe storage device, in an openQRM environment a Hypervisor becomes “just” a resource provider, just being responsible to host the virtual compute resource of the users choice. That way the appliance running on a virtual machine also gets fully independent from its Hypervisor Host and can be transparently (live-) migrated to another hypervisor of the same or different virtualisation technology or even from physical systems to virtual machines and the other way around. OpenQRM supports P2V, V2P, V2V, P2P migration without any changes on the actual server-images itself.

OpenQRM cannot only manage different types of Hypervisor technologies but it can also deploy them via the regular generic deployment mechanism of its framework. That offers scalability for the complete IT infrastructure because the data centre can grow (and shrink) as demanded by just adding (or removing) more hypervisors.

Enhanced High-availability with N-to-1 fail-over

When talking about “Green IT” the current approach is to use virtualisation to consolidate “many” physical servers to run virtualised on “one” (or more) Hypervisor Hosts. While this method is good to save the overall power consumption it is often forgotten that the in the new, virtualised situation in the case the “one” Hypervisor Hosts breaks those “many” servers running in virtual machines on this Hosts will also get unavailable. This means that in the modern, virtualised worlds we need to especially take care of the high availability.

The usual method to keep a system highly available (10 custom servers):

- get an additional 10 servers preferred of the same manufacture consisting of the same parts
- configure syncing of the disks between the 10 pairs of servers
- implement a fail-over solution for the service running on the 10 clusters

As a result this method requires 20 physical systems to keep 10 servers highly available.

HA in openQRM:

- deploy the 10 custom servers via openQRM
- add 1 server as Hot-Standby

In the case one of the 10 custom servers break openQRM will use the one available system to restart it via its rapid deployment methods. As the result 10 (or more) servers can made high available with just a single hot-standby system (since resource types are not linked to the actual server-image in openQRM physical servers even could fail-over to virtual machines). This saves the power consumption of 9 servers!

Plug-ability → Automatism → Cloud Computing

On top of this straight and generic openQRM framework sits the Cloud plugin as a “simple UI” for the end user. It provides a fully automated provisioning cycle from physical systems or virtual machines deployment through automated application and configuration management according the users requests up to automated de-provisioning to free up the users compute resources. Via an additional external Cloud web-portal users registered to the Cloud can login to manage their own Cloud environment by requesting new resources, de-provisioning existing ones or managing their active Cloud requests.

For the system administrator the Cloud provides an internal interface plugged into the openQRM server. It provides a fine grained overview about the current Cloud activities and several configuration options to tune the Cloud behaviour e.g. the Cloud can be set to automatically or manual approve new Cloud request, automatically create new virtual machines if not enough existing one are available, enable or disable the clone-on-deploy features etc. It also consist of a Cloud IP manager which is used to automatically configure the external network-interfaces of the provisioned machines.

The default behaviour of the openQRM Cloud is to use the “clone-on-deploy” mechanism to provision resources requested by the users. This means that for every Cloud request the integrated storage management will execute a clone command on the storage server hosting the “golden image” (server image template) and then use the snapshotted server image to deploy it for the user. This method has the huge advantage that the snapshots on the storage server (e.g. via the LVM snapshot storage feature) normally do not consume any space because they are only read-only copies from the original logical volume (server image). This means only the changes the users makes to its server are saved to the storage and no additional space for the actual root file system plus its applications.

To allow for automated leveraging of the application layer of the provisioned systems according to the users request openQRM is integrated with the Puppet configuration management system. Puppet looks after the setup and pre-configuration of the users' machines via pre-made and known-to-work recipes. The integration of Puppet as an additional plugin and the cooperation of the Puppet-plugin and the Cloud add-on provides a selection of "out-of-the-box" application servers and gives added value by automatism for the users and administrators.

The openQRM Cloud comes with an integrated billing system for the compute resources consumed by the end users. Based on "Cloud Computing Units" (CCU's), the virtual currency in the openQRM Cloud, system administrators and users can plan and keep track of their used compute power and costs. OpenQRM simply calculates the amount of CCU's per request and subtract it from the Users account. That way the Cloud manager is completely open how to sell the computing power to the market end users (e.g. via Ebay).

Private and public Cloud Computing / Security Aspects

Clouds can be divided into "Public Clouds" and "Private Clouds". While within a private cooperated data centre environment security aspects may depend on the system administrator and on their actual usage. For a internally QA network, protected by the companies firewall, security may not be that critical compared to the load balancer cluster of the main database application. In any way for Public Cloud Computing an exhaustive security concept is needed to ensure the integrity of the users data and provide continuously reliability for the Cloud service.

OpenQRM's thoroughly designed security concept consist of automated authentication of the server-images, the separation of the hard- and software layer to ensure the data-integrity and enable service high availability via rapid re-deployment plus the separation between the actual Cloud management system and the end users Cloud portal.

To gain additional security on top of the Cloud security virtual private networking (VPN) and file system encryption can (and should) be additionally used in the application layer.

Conclusion

With its unique architecture and generic concepts of rapid and fully automated deployment openQRM provides a complete and extensible data centre management framework for large IT-infrastructures. Its Cloud Computing User Interface additionally gives instantly and on-demand access to the compute resources for the end-users.

Algorithms in a Nutshell (In a Nutshell (O'Reilly))

George Heineman, Gary Pollice and Stanley Selkow

O'Reilly Media
ISBN: 978-0-596-51624-6
362pp.
£ 38.50
Published: 21st October 2008

reviewed by James Youngman

The authors state "We intend this book to be used frequently by experienced programmers looking for appropriate solutions to their problems". This focus allows them to cover a wide field in a fairly short book.

“Algorithms in a Nutshell” is the first newly-printed O’Reilly book I’ve come across in a while, and the first thing I noticed about it is that the paper seems less smooth than older O’Reilly books. Perhaps it’s recycled paper or something, but I found it feels less pleasant in my hands than older O’Reilly books.

This is an interesting time to be reading algorithms books; several fascicles of the long-awaited volume 4 of Knuth’s The Art Of Computer Programming have recently been published. Also but less famously though, the second edition of Steven Skiena’s “The Algorithm Design Manual” was published in 2008. More of this later.

Many algorithms books are largely formal in nature, dealing with the analysis of the correctness and asymptotic efficiency of algorithms. Most of the time I spend reading algorithms books I’m actually looking for an algorithm applicable to my problem in order to implement it. Quite often the style of pseudo-code presentation in such books isn’t very suitable for easy implementation. The book singles out Cormen’s widely recommended and authoritative “Introduction to Algorithms” book as an example of such books. They certainly have a point.

In terms of presentation, the “Algorithms in a Nutshell” book has much more in common with “The Algorithm Design Manual” or perhaps Robert Sedgwick’s “Algorithms in . . . ” books, though it’s certainly shorter than either.

The Nutshell book includes Java, C or C++ implementations of each algorithm. It’s divided into three parts; part 1 contains introductory material, part 2 presents the algorithms and parts 3 and 4 discuss non-optimal algorithms and benchmarking. Part 2 breaks down into:

- Sorting Algorithms
- Searching
- Graph Algorithms
- Path-Finding (mostly dealing with decision trees)
- Network Flow
- Computational Geometry

The ground covered is roughly the same as Sedgwick’s work, though there is much less emphasis on data structures. I think this is a reasonable decision for books intended for working programmers; most professional programmers will have available standard libraries providing implementations of advanced data structures (indeed, if they are not already built into the language they’re using). Omitting material on data structures would have been a bad idea for an introductory text, but that’s not what this book intends to be.

Like most programmers I need to be careful about the origins and license of any code that I re-use. The O’Reilly book attempts to set out clearly that you can re-use their code in your programs (though they fail to define the boundaries of this any more closely than “unless you’re reproducing a significant portion of the code”). That seems much more helpful than Skiena’s blanket inhibition “Permission is granted for use in non-commercial applications provided this copyright notice remains intact and unchanged.” That would certainly make it impossible for me to incorporate his code in a GPLed program, for example. That would make Skiena’s book quite irritating to use if it weren’t for the fact that he indicates for each algorithm which online code collections provide good implementations.

Although I have a copy of each of the books I’ve mentioned in this review, I don’t use them all equally. I probably make least use of Cormen (it sits on my work bookshelf looking imposingly authoritative but almost never gets consulted) and so far, probably most use of Skiena’s “The Algorithm Design Manual”. My guess is that I’d probably refer to both the Skiena book and the Nutshell book when looking for an algorithm (and, in all likelihood, search the web once I figured out a shortlist of appropriate algorithms). It’s a little early to say for sure whether I think it’s worthwhile to own both books, but if I had to choose between them I’d probably choose Skiena.

Learning JavaScript

Shelley Powers

O'Reilly Media
ISBN: 978-0-596-52187-5
396pp.
£ 26.99
Published: 26th December 2008

reviewed by Gavin Inglis

Javascript. It lurks there on the key sites of the modern Web, tucked into the `<head>` element. At its best it flows like water, eliminating awkward clicks and server transactions and making interaction a smooth, effortless joy. At its worst, it deletes form content, insists on inputs one doesn't wish to supply, and condemns the browser to a grinding death whilst slowly fading in a badly framed photograph of a cat in a cardboard box.

O'Reilly publishes a "Learning . . . " volume for several major programming languages. These are intended to complement the more definitive reference books and lead a learner through the key features of the language, observing its quirks along the way. When selecting a tutorial book it's important that the student match their expectations to the text's approach.

Learning Javascript is a thorough introduction to the language. It assumes some familiarity with programming concepts, but it begins with basics and almost 150 pages pass before the browser magic appears. As such it is not a good choice for a web developer who needs flashy Javascript effects *right now*. After the nineteen page early chapter discussing the subtleties of data types and null versus undefined variables, such a reader will likely move on to another book, if not another career.

The ideal audience is probably someone who has decided to learn Javascript and is willing to put in the time to gain a thorough understanding of the language with no expectation of immediate results. After the obligatory Hello World example, the road leads slowly and surely through variables, operators and statements, objects, functions, and so on, each chapter building on the last until we reach event handling, DOM, dynamic pages and finally AJAX.

Throughout there is an emphasis on detail: when you might want explicitly create a String object rather than a string primitive, and why; how to avoid circular references and memory leaks; planning to avoid cross-site scripting attacks. Where the language displays unintuitive behaviour, this book explains why and is normally illuminating.

Browser compatibility is always a question with Javascript, and *Learning Javascript* deals with this square on, making clear immediately its target browsers: Firefox 3, Opera 9, Safari 3 and Internet Explorer 8 (with IE6 and IE7 dragging their heels at the rear). We receive occasional glimpses into the tortured past of browser wars but thankfully are advised to leave all this behind. Accessibility is also given a welcome early focus.

This is a volume for the programmer who is willing to sit down and grind through the basics in order to reach a deeper understanding of the Javascript language. It does feature a useful quiz at the end of each chapter to check your knowledge. However if you simply need to pick up some Javascript quickly to spread glitter on your web site, the pace may leave you feeling like the black rhinoceros on the cover.

MediaWiki (Wikipedia and Beyond)

Daniel Barrett

O'Reilly Media
ISBN: 978-0-596-51979-7
374pp.
£ 30.99
Published: 30th October 2008

reviewed by Sam Smith

This book is one of those that you either need or don't need. If you don't do anything with wikis, or you have something already which meets all your needs, then other than a little light reading, there's not much in here for you.

On the other hand, if you think that communal editing of documents might be useful, or you're looking at a using (or changing) a wiki platform, then the book has enough useful and detailed content to be worth the money. MediaWiki is one of the heavyweights in the wiki software world, with an international userbase and many features both in usability and scalability. Being written by the folks at wikipedia helps, and also has high quality, detailed online documentation.

While much of the material is in places on the web, it's well put together for someone who isn't a MediaWiki expert, and doesn't quite know what they're looking for, but is aware of that it can probably do something like that, but not sure where to start looking.

This book isn't the Definitive Guide to Everything Ever about MediaWiki, and doesn't try to be. It covers the material it covers well, and is very useful to its target audience.

Using Drupal

Angela Byron, Heather Berry, Nathan Haug, Jeff Eaton, James Walker and Jeff Robbins

O'Reilly Media
ISBN: 978-0-596-51580-5
492pp.
£ 34.50
Published: 16th December 2008

reviewed by Paul Waring

If you haven't heard of Drupal, you've probably visited a site powered by it at some point, from KernelTrap to Yahoo! Research and MTV. Using varying combinations of modules, you can create a content management system customised to your needs, though this enormous flexibility comes at the cost of sometimes not knowing where to start. Thankfully help is at hand from the six authors who have contributed to Using Drupal, three of whom are directly involved in developing the software.

The introduction to this book is mercifully brief and the authors appear to have avoided the temptation to pad out the entire first chapter with a tedious step by step explanation of how to install the software on every platform imaginable (for this see the comprehensive appendix or the Drupal site). Instead we are treated to a simple explanation of how content management systems work with specific examples from Drupal. The next chapter moves at a rapid pace and takes you through a number of common tasks, such as setting up user permissions and changing themes.

After getting you up and running with the basics, the remainder of the book consists of a number of case studies, from a job board to event management, with multi-lingual sites and customised themes also getting a look in. Most impressive of all, the appendix lists the modules and themes used in each chapter, so you can easily recreate the same environment on your own hosting platform, should you wish to.

The most useful chapter for me covered building a wiki with Drupal. This is something which I've been looking at for some time, particularly because the existing wiki solutions tend to have poor access control (particularly Mediawiki) or are difficult to integrate with an existing site. With this chapter to hand, I now have all the software and configuration steps I need to get started, as opposed to having to trawl through dozens of search results trying to work out which wiki module to use and how to set it up as the developer hasn't given any thought to documentation.

So far, so good. However, the major potential problem with any book which covers a web application is that the content is often out of date even before the text hits the printers. Fortunately, *Using Drupal* manages to cover the latest major version, and six months on it is still sufficiently up to date to be a useful guide to the software.

Overall, this book is well worth a look if you are thinking of setting up a Drupal site or want to do more with your existing instance. Whilst the Drupal web site does include some excellent documentation to get you started, I have yet to find an online source which takes me through the whole process of setting up and customising a Drupal site in a coherent manner. This book fills the gap nicely.

Masterminds of Programming

Federico Biancuzzi and Shane Warden

O'Reilly Media
ISBN: 978-0-596-51517-1
494pp.
£ 30.99
Published: 30th March 2009

reviewed by Paul Waring

Masterminds of Programming is a series of interviews (twenty seven in total) with the people responsible for many of the programming languages used in everyday programming, and some more obscure ones too. From Perl and C++ to Haskell and ML, most readers will have used at least some of the languages under discussion. The interviews contain some fascinating insights within the text as to why some languages developed in the way that they did, the various design decisions which influenced them and the particular itches which they were created to scratch.

For me, however, the most useful aspect of this book was reading about all the programming languages which I had never heard of before, such as APL and Forth, and those which I had heard of but knew little about, including Eiffel. Although there is virtually no discussion of code within the interviews, the questions and answers provide enough information to get a feel of the unique aspects and applications of some of the less well known languages. As a result, I have been encouraged to check out these languages and see if they can be applied to some of the projects which I am currently working on.

My only minor criticism of the text is that many of the interviews seemed to end abruptly, there were no questions which seemed to wrap up the chapters, such as "finally, where do you see <language> heading in the future?" I would have liked to have heard more about plans for the next few years, as this is often more interesting than what has happened in the past. PHP was also noticeably absent from the list of interviews, which is somewhat surprising given the sheer number of web sites which rely on this language, including the one which I spend the majority of my working day developing. However, at nearly five hundred pages one could justifiably argue that the book is large enough as it is, and some languages had to be left out.

Overall, this is a genuinely interesting text with insights into such a wide variety of languages that everyone should find something new. I would recommend it to anyone who is interested in broadening their horizons beyond the two or three core languages which they use on a regular basis, and it is well worth a read if you are interested in the history of computer languages in general.

Ubuntu Kung Fu

Keir Thomas

Pragmatic Bookshelf
ISBN: 978-1-934356-22-7
400pp.
£ 21.99
Published: 28th September 2008

reviewed by Andy Thomas

Having been brought up on a diet of O'Reilly books, I found the house style of *Ubuntu Kung Fu*, published by Pragmatic Programmers, refreshingly different – and that's not just down to the quirky title. This is the first book I have read from this publishing house and has a lighter and more conversational feel to it; liberally illustrated with screenshots, large boxed headings in the tips section and wide page margins, it is written for 'ordinary users' and is about as different as you can get from tomes like O'Reilly's Sendmail book, for example.

Aimed at the growing army of users defecting from Windows to Ubuntu, the 'meat' of the book is sandwiched between a very extensive contents list at the front and a comprehensive index at the back. Finding a tip relevant to whatever you want to achieve with Ubuntu is easy as each one-line entry in the contents list is descriptive yet concise – this is something that sets this book apart from most Linux books written with a non-geek audience in mind and will appeal to many users.

The main part of the book is divided into just three chapters but do not be deceived – chapter 3 runs to 308 pages of this 367 page book and contains no fewer than 315 tips. 'Chapter 1' is a short introduction of less than 4 pages that would be called a preface in most other books but we finally get started with chapter 2 which covers basic Ubuntu system administration. Although no substitute for a dedicated book on this topic, this chapter does make a reasonably successful attempt at softening the transition from Windows to Linux and it is good to see the use of the command line explained in some detail as this is where the real power of Linux lies. The reviewer feels CLI usage should be encouraged more – although desktop environments such as KDE and gnome do a good job of blurring the distinctions between different Linux distributions, I often hear end users wail "... but I can't use this system as it's not Red Hat" or some such. Dropping down into the command line is a great leveller and if a user takes the trouble to learn how do things that way, they can pretty much admin any Linux, UNIX or Mac OS X system.

I did spot one or two technical errors in this chapter – for example, on page 20 it is claimed that the bash shell includes the handy command 'sort' for sorting a shopping list into alphabetical order but the following command line example then goes on to correctly illustrate the use of the (quite separate) sort utility. These small slips don't really affect the end result though.

A section devoted to software package administration then follows – Debian is known for its somewhat bewildering (at least compared with some other Linux distributions) array of package management methods and Ubuntu goes one step further by adding synaptic. Fortunately most of these tools are at least touched upon with the emphasis being on the GUI way of doing things but `apt-get` and even `dpkg` have not been forgotten.

But the real raison d'être for this book has to be chapter 3, the tips. Ranging from the really useful to the humorous and even trivial, readers of all skill levels will almost certainly find this section holds something of interest for them. Most of the tips concern desktop tricks and short-cuts and this reviewer, being mainly involved in Linux/UNIX infrastructure (servers, networks and the like with little exposure to gnome for instance) learnt quite a bit about Linux on the desktop and the Ubuntu way of doing things. The author's infectious enthusiasm for imparting these little nuggets of wisdom continually shines through in each tip and he is to be congratulated on his way of making Ubuntu complement, not fight, Microsoft Windows and its applications and this will make Windows users feel very much at home and 'wanted' within the Ubuntu community, unlike some books on Linux. The acid test was when I lent the book to my 14 year-old son who had abruptly switched from Windows to Ubuntu some months previously; although well up the Ubuntu learning ladder by this stage, he too found the book useful.

All in all, this will be a useful book for those making the big step from Windows to Linux – much of the information in chapter 2 and many of the tips apply equally to other Linux distributions so readers moving to OpenSUSE or Fedora instead of Ubuntu or its variants will find much relevant information about gnome, its configuration, applets and the various applications available to all Linux users. Sprinkled through the book are frequent references to Windows, Windows partitions, Windows files, etc – a clear reminder that the target readership is likely to be dual-booting between Windows and Linux, at least in the early days before they feel more confident with Linux. But this should not put off Linux-only readers and any book that encourages users – and in such a human and friendly way as this book does – to upgrade their computing experience from Windows to Linux deserves to be applauded.

The Google Way

Bernard Girard

No Starch Press
ISBN: 978-1-59327-184-8
256pp.
£ 19.99
Published: 3rd April 2009

reviewed by Roger Whittaker

The Google phenomenon has spawned quite a number of books about the company's history and the reasons for its success. David Vise's *The Google Story* was published in 2006 and is an interesting and mostly admiring factual history of Page and Brin and the company that they built, drawing its moral and practical conclusions mainly only implicitly. Also published in 2006 was *The Search: How Google and Its Rivals Rewrote the Rules of Business and Transformed Our Culture* by John Battelle, which I have not read.

This year at least two books have been published which try to draw more explicit lessons from Google's success for others to learn from: one is *What Would Google Do?* by Jeff Jarvis, and another is this book **The Google Way** by Bernard Girard.

There will undoubtedly be more such books, because the Google phenomenon is so extraordinary, and because there is a natural human tendency to try to look at any greatly successful individual or organisation (Caesar, Napoleon, Microsoft. . .) and try to boil down the essence of their success into a formula, a magic medicine which, if you take two drops of it each day, can give you the same success.

However, this book does *not* come into the category of those cruder "self-help" histories that fall too easily into the trap of *post hoc ergo propter hoc* and prompt the critical reader want to ask the author "if you know so much about how it's done, then why are you writing books and not a billionaire / world emperor / whatever?".

Girard's book was originally published in French in 2008. He explicitly makes large claims for what can be learned from Google: the introduction is entitled "A Management Breakthrough", and throughout the book he looks at particular attitudes, working practices and strategies that Google has adopted and analyses why he thinks they have been so powerful.

The author seems to draw on quite a lot of insider knowledge for some of the detail of his descriptions of what goes on at the Googleplex, and to some extent breaks through the company's combination of openness about the wider generalities of how they work and secrecy about the detail.

Despite often talking as though Google's overall approach constitutes unique genius, when discussing the details Girard is more than ready to admit that many of the particular features that he singles out had been adopted with success in the past by other companies. So, for example, Google's well-known "20 per cent" rule (one fifth of your time for your own projects) was anticipated by similar practices years ago at 3M and HP. Similarly Google's preference for small teams is not unique, and nor is its understanding of why there is an optimum team size for many kinds of collaborative work.

The book particularly emphasises the contribution of unfettered research (try something and see what it might lead to) and of a deliberately fostered culture of peer review as factors in Google's innovative fertility. Girard is also impressed by the company's policy of "recruiting the best", devoting a chapter to hiring practices in which he details an extraordinary emphasis on extremely thorough interviewing and selection methods. He quotes a figure of one Google employee in 14 working in recruitment in 2005 (presumably this is quite apart from the highly organised but very time consuming process of informal but very exacting "peer interviewing" that puts every successful candidate through perhaps eight separate interviews). He includes an example of what he claims is a genuine Google Labs Aptitude Test question: his example is a rather wacky "psychological" question based on a social situation (*on your first day at Google, you discover that your cubicle mate wrote the textbook you used as a primary resource at graduate school*, followed by humorous multiple choice suggested reactions). I suspect this is not typical.

I was impressed by the description of how Google has automated the process of advertising sales and customer relationships, and also by the author's speculation that the particular personalities of Page, Brin and Schmidt make the triumvirate at the top particularly productive and stable.

The book concludes with some interesting speculations about the future ("Can Google Evade Conformity" and "A look ahead").

I have one criticism of the design of the book: for some reason No Starch Press have decided that each chapter should begin with a page that is half blank and half set in super-sized text (about 18pt). I suspect that I will not be alone in finding this style annoying and pretentious.

I found this an interesting book about a very interesting phenomenon, and I can recommend it.

Python for Unix and Linux System Administration

Jeremy Jones and Noah Gift
O'Reilly Media
ISBN: 978-0-596-51582-9
456pp.
£ 38.50
Published: 2nd September 2008

reviewed by Roger Whittaker

The aim of this book is to encourage people to use a better scripting language for general tasks in the area of system administration. This is an aim with which I have full sympathy.

The first two chapters describe the philosophy of the book and then launch into an introduction to Python using the IPython interactive Python shell throughout, and going into rather a lot of detail about the specific features of IPython. The authors themselves state that this is unusual, and I personally did not see the point of concentrating so heavily on IPython rather than just discussing general language features and using the standard interactive Python prompt.

The book emphasises practical examples, and in many sections offers simple enough examples to get the reader started easily, which is a good test of the usefulness of a book like this. I personally learned how to use both the `xml.etree` (ElementTree) and `subprocess` modules (both of which are relatively new in the standard library, and which I hadn't used before) by playing with the examples in the book.

Some of the types of tasks that the book covers are: writing log file parsers and analysers, writing networking clients, using Python to automate backups and similar tasks, using Python with SNMP for systems monitoring, and using Python in cross platform environments.

There is a large chapter devoted to Python package management, and brief introductions to building GUIs with PyGTK and creating web applications with Django. There are also sections on using python with LDAP and on persisting data with shelve, pickle, yaml, ZODB and sqlite.

A nice feature of the book is that dotted about in it there are occasional little boxed "celebrity profiles" with photos and brief profiles of prominent Python contributors and community members.

As I have already indicated, what I like best about the book is the accessibility of the examples and the practical approach. I would recommend it, but not as your only Python book.

Contributors

Gavin Inglis works in Technical Infrastructure at the EDINA National Data Centre in Edinburgh. He is a teacher, photographer and musician who has recently discovered the joys of spreadsheets as a recreational tool.

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.

Matthias Rechenburg is project manager of the openQRM project. For many years he has been involved in all kinds of data-center related open-source projects like high-performance and high-availability clustering, consolidation, network and enterprise storage management. Currently, his most serious interests are in virtualisation technologies, their features and capabilities and integration by a unified virtualisation layer. He lives in Bonn, Germany, and is working as a freelancer for virtualisation- and storage-management projects. Mostly, he enjoys to code in his home-lab but also likes traveling, meeting other Linux-people and joining all kinds of Linux-related events and congresses.

Peter H Salus is the author of *A Quarter Century of UNIX* (1994), *Casting the Net* (1995) and *The Daemon, the Gnu and the Penguin* (2008), and the editor of nearly a dozen other volumes.

Sam Smith served on UKUUG Council for 6 years, and now herds cats for UKUUG's series of training and tutorial events. He also runs the OpenTech Conference and is heavily involved in EuroBSDCon 2009 to be held in Cambridge.

Andy Thomas is a UNIX/Linux systems administrator working for Dijit New Media and for Imperial College London and as a freelancer. Having started with Linux when it first appeared in the early 1990's, he now enjoys working with a variety of UNIX and Linux distributions and has a particular interest in high availability systems and parallel compute clusters.

Paul Waring is chairman of UKUUG and currently trying to complete the "final" draft of his MPhil thesis whilst working as the Technical Manager for an insurance intermediary. He is also responsible for organising the UKUUG Spring conference in 2010.

Roger Whittaker works for Novell Technical Services at Bracknell supporting major Linux accounts in the UK. He is also the UKUUG Newsletter Editor, and co-author of three successive versions of a SUSE book published by Wiley.

James Youngman spends his days downloading web pages and his nights emptying `/dev/null` as a member of the UKUUG's Waste Disposal directorate. Programs written by James that you might have used include "find", "xargs" and "epicycle". His hypothetical interests outside computing include rock climbing and photography.

Contacts

Paul Waring
UKUUG Chairman
Manchester

John M Collins
Council member
Welwyn Garden City

Phil Hands
Council member
London

Holger Kraus
Council member
Leicester

Niall Mansfield
Council member
Cambridge

John Pinner
Council member
Sutton Coldfield

Howard Thomson
Treasurer; 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

Roger Whittaker
Newsletter Editor
London

Alain Williams
UKUUG System Administrator
Watford

Sam Smith
Events and Website
Manchester