news@UK

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

Volume 19, Number 1 **ISSN 0965-9412** **March 2010**

Contents

News from the Secretariat

Jane Morrison

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

We now have everything in place for the UKUUG Spring Conference and Tutorials, being held in Manchester (23rd - 25th March). The event will again be sponsored by Google and Bytemark which has enabled us to organise a Conference Dinner at the famous Yang Sing restaurant.

We have also been able to keep the event delegate fees low for members via the UKUUG Gold sponsor members Novell, IBM and SUN Microsystems.

A full list of talks can be found in this Newsletter – all abstracts and speakers biographies can be found on the web site at
`http://spring2010.ukuug.org/Talks`

Remember that by attending the Spring event you can:

- Keep abreast with new/emerging technologies
- Network with some of the people who are responsible for developing critical applications
- Become part of the UK Unix community – build up informal relationships that can be invaluable in problem solving
- Benefit from the experience of delegates with similar interests
- Keep staff happy and feeling valued
- Make valuable contacts with sponsor organisations such as Google, Bytemark, IBM, Novell and Sun
- Secure essential O'Reilly books at a significant discount to the cover price

Following the success of the 3 days of Perl tutorials in November we are organising a repeat of the 3 days in April (13th, 14th and 15th) - please see the flyer enclosed with this Newsletter. This event is another successful collaboration for UKUUG & O'Reilly.

Plans are also afoot to organise an unconference (more details coming soon).

The UKUUG AGM will be held on Thursday 23rd September – please put this date in your diary now.

The next Newsletter will be the June issue and the copy date is Friday 21st May.

Chairman's report

Paul Waring

Upcoming Conferences

By the time you read this, Spring 2010 will be almost upon us, although it's not too late to secure your place if you haven't already! We have a full day tutorial on SCons (a replacment for make) by Russel Winder, and two tracks of conference talks including speakers from Google, Canonical, FreeBSD and many more. A busy social programme, including drinks and a curry on the Tuesday evening and a conference dinner at a local chinese restaurant, provides an opportunity to network with other delegates whilst enjoying some of Manchester's varied cuisine.

Looking forward to the rest of the year, we are planning to run an unconference in the autumn in Birmingham. This event will take place on a Saturday so that delegates can travel to the venue and back in

one day and without having to book time off work. More details will be released shortly, but if you are interested in getting involved with this event, please let us know – you can email us at:
`council@ukuug.org`

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. We are particularly keen to get members involved in running next year's spring conference, so if you would like to help out in any capacity please do get in touch.

UKUUG – Spring Conference 2010

23rd, 24th and 25th March 2010 – Manchester Conference Centre

Tuesday 23rd March – SCons Tutorial (tutor: Russel Winder). SCons is a Python-based replacement for Make, and effectively the whole Autotools chain. A full day, including refreshment breaks and lunch.

Wednesday 24th and Thursday 25th March – 2 day Conference. The provisional progamme, abstracts and bios can be found at
`http://spring2010.ukuug.org/Talks`

Some highlights of the conference:

- The New VVorld – Bjoern Zeeb and Robert Watson (FreeBSD)
- Herding a highly available MySQL cluster at Google scale – Andrew Stribblehill (Google)
- Ubuntu Server 10.04 LTS: The making-of Ubuntu Enterprise Cloud: From bare metal to cloud in 30 minutes – Thierry Carrez (Canonical)
- Hudson hit my puppet with a cucumber – Patrick Debois and Julian Simpson
- MySQL HA with pacemaker – Kris Buytaert
- 21st Century Systems Perl – Matt S Trout
- Situation Normal, Everything Must change – Simon Wardely
- Building an infrastructure for open source development – Simon Wilkinson (Univ of Edinburgh)
- PostgreSQL High Availability and Scalability – Simon Riggs
- Securely storing large volumes of data at rest with ZFS – Darren Moffat (Sun Microsystems)
- The Perdition and nginx IMAP Proxies – Jan Piet Mens
- Grid and Cloud Computing From Research to Deployment – Ruediger Berlich
- Libferris teaching filesystems new tricks – Ben Martin
- C Language Variants - An Overview + The state of Linux Installs – Donald Forbes
- Designing and deploying a manageable virtual desktop infrastructure – Mike Banahan (GBdirect)
- Simplifying software appliance creation with SUSE Studio – Matt Barringer (Novell)
- Building an infrastructure for open source development – Simon Wilkinson
- GNUBatch - John Collins (Xi Software Ltd)
- Distributed parametric optimization with the Geneva library – Ruediger Berlich
- Security for the virtual datacentres – Sas Mihindu
- Coherent and Integrated Configuration of Virtual Infrastructures – Panagiotis Kritikakos

- Creating an affordable Smartphone that can be called Desktop PC – Luke K. C. Leighton
- Conference Dinner: Yang Sing restaurant – included in Conference delegate fees!

Places still available -book on-line at:
`http://spring2010.ukuug.org/Registering`

Event Sponsored by: Google, IBM, Novell, SUN, Bytemark Consulting and antibodyMX

OpenTech 2010 – 11th September, London

Sam Smith

UKUUG is once again facilitating OpenTech in 2010 on 11th September.

Open Tech 2010 is an informal, low cost, one-day conference on slightly different approaches to technology, politics and Justice. This year's theme is "Working on Stuff that Matters" guarantees a day of thoughtful talks and conversations with friends.

The bar will remain open for informal discussions into the evening after a day of sessions which will challenge and inspire you. The last three times we have sold out in advance, so you are strongly advised to pre-register.

For further details, to offer a talk, or to find out how to register, visit
`http://www.ukuug.org/opentech`

OggCamp10 announcement

Dan Lynch of Linux Outlaws has passed on details of the OggCamp10 unconference which will be held in Liverpool on the 1st and 2nd of May. The event will be held at the Black-E, Great George Street, Liverpool L1 5EW.

This follows on from last year's OggCamp which was held the day after LugRadio Live in 2009. It is a Free Software and Free Culture Barcamp, expanded this year to become a full weekend of technology, art, music, creativity and fun in Liverpool.

OggCamp10 is being organised by Linux Outlaws in conjunction with the Ubuntu UK podcast team.

For more details see
`http://oggcamp.org`

Erlang Factory London 2010

We have received the following announcement.

Erlang Factory, the biggest Erlang gathering in London is coming back! The Erlang Factory conference will last two days from 10 to 11 June 2010. Like in previous years the conference is aimed at Erlang enthusiasts from architects to newbies. The Erlang Factory is a perfect event for networking, learning and for sharing knowledge, experience and passion of the Erlang language.

At the Erlang Factory London 2010 there will be over 40 speakers, 35 talks and 6 separate tracks. Delegates can choose which talks they want to attend and move freely among the tracks. The keynote speakers are Martin Odersky and Robert Virding. Robert Virding, inventor of Erlang will give a talk on "The history of the VM" and Martin Odersky inventor of Scala talk about "The influences of Erlang's concurrency model when inventing Scala".

In addition to the conference, there will be a three day Erlang University, three days of training courses at all levels allowing you to learn Erlang from the experts. You can attend the Erlang Express course, taught by O'Reilly Erlang Programming author Simon Thompson. Experienced users can attend the OTP Express course, taught by Erlang Solutions' Training Manager Henry Nystrom, or a 3-day introduction to Quick Check given by John Hughes and Thomas Arts of QuviQ. By following three days of training in Erlang with two days of conference talks and networking, you will benefit far more than by only having the training by itself.

In 2009, the first year of the Erlang Factory Conferences, a total of over 300 people attended, showing the level of interest for the technology. If you want to see how the Erlang Factory looked last year, read the programme, download the presentation slides and watch the videos, you can do all that here:
`http://www.erlang-factory.com/conference/London2009`

Booking

Registration for the Erlang Factory London will be open at the beginning of March.
`http://www.erlang-factory.com/conference/London2010`
For further details please visit
`http://www.erlang-factory.com`

News from the ODF Alliance

This is an edited selection of items of news that we have received from the ODF Alliance showing progress towards the use of open document standards across the world.

Denmark opts for ODF

Beginning 1 April 2011 governmental authorities in Denmark will be required to send and receive documents in formats designated in a list now including ODF. ODF is unique as the only editable document format listed in the decision of the Danish Parliament. To the extent Danish government authorities publish editable documents on their home pages, they must also do so using ODF and, optionally, using other document formats that may be included on the list at a later date. For a format to be included on the list, a five-part "openness" test was developed. This included a requirement that any other format considered for inclusion be interoperable with the existing standard on the list, meaning that it must be interoperable with ODF. For non-editable published documents PDF/A-1 is listed. The action was taken in accordance with Danish parliamentary decision B103 of 2006 requiring the government to ensure that the use of information technology by the public sector is based on open standards.

Open standards a "first-choice" solution for Sweden's e-government strategy

Citing the opportunities for long-term cost reductions, the avoidance of lock-in and dependence on individual suppliers, a government-appointed group of senior officials, the so-called E-Government Delegation ("E-Delegationen" in Swedish), has recommended that Sweden make open standards a "first-choice" solution in the public administration. ODF is specifically referenced as an example of an open standard, the designation of which is included in the list of actions that the E-Delegation recommends should be carried out by the end of 2014. The report – "Strategy on the work of the Public Agencies in the field of eGovernment" – proposes ways of increasing the efficiency of the Swedish public administration and promoting societal innovation through eGovernment.

OOXML unsuitable for use by Norwegian Government, according to study

A study published by Norway's Agency for Public Administration and ICT ("Direktoratet for forvaltning og IKT") has concluded that OfficeOpen XML (OOXML) is not suitable for use by the Norwegian government. Among the reasons cited were the lack of alternative office applications able to process and edit docx files in a satisfactory manner, OOXML's unsuitability for collaboration, and its "unstable" nature given the number of changes to the format currently being considered. Norway recently affirmed its policy that, beginning 1 January 2011, it will be obligatory to use ODF when exchanging editable files between government institutions and users, PDF/A for non-editable (read-only) files, and HTML for publication of public information on government websites.

Obama Adminstration gives green light to open formats

U.S. federal government agencies will soon be required to make information available in open formats. According to the Open Government Directive issued by the Obama Administration, each agency will be required to "take prompt steps to expand access to information by making it available online in open formats. . . To the extent practicable and subject to valid restrictions, agencies should publish information online in an open format that can be retrieved, downloaded, indexed, and searched by commonly used web search applications." An open format is defined in the directive as one that is platform independent, machine readable, and made available to the public without restrictions that would impede the re-use of that information. Within 45 days of the publication of the directive on 8 December 2009, each agency was required to identify and publish online in an open format at least three high-value data sets. Agencies are required to produce a first draft of an Open Government Plan by April 2010.

ODF approved for use by Slovak Government

Slovakia has approved amendments to a decree on the use of standards for information systems in the public administration. Government bodies in Slovakia must now be able to receive text documents in ODF, PDF 1.3, RTF and HTML. They may publish documents in any of these formats, though PDF is preferred. For intra-governmental document exchange, the use of DOC will continue to be allowed, though its use as a format for published documents is explicitly prohibited. A working group of the Committee for Information Systems in the Ministry of Interior was able to reach agreement on the amendments to the decree, which is legally binding and took effect 1 February 2010.

UK Government to support open standards based solutions, ODF

In a refresh of the "Open Source, Open Standards and Software Re-Use: Government Action Plan," part of the Government's ICT Strategy, the UK has reiterated its support for open standards and ODF. According to the plan, the UK government will use open standards in its procurement specifications and require solutions to comply with open standards. Regarding formats, the government "will support the use of HTML(ISO/IEC 15445:2000), Open Document Format (ISO/IEC 26300:2006) as well as emerging open versions of previously proprietary standards (eg ISO 3200001:2008 ("PDF") and ISO/IEC 29500 ("OfficeOpen XML formats"). It will work to ensure that government information is available in open formats, and it will make this a required standard for government websites." The strategy applies to all of the UK public sector, whether central government, local government, wider public sector or devolved administrations.

Hungary to make open standards mandatory

The use of open standards in public-sector infrastructure will now be mandatory in Hungary. According to an amendment passed by the Hungarian Parliament on the law governing electronic public services (Act LX of 2009), open standards are now required in electronic communication conducted through the central governmental system between public administrative bodies, public utility companies, citizens, and private entities, who may comply on a voluntary basis. The modification was supported by the Open Standards Alliance and ODFA Hungary.

Assam Government to create and store documents in ODF

The Government of Assam's policy requiring government departments and bodies to ensure adherence to ODF in creating and storing editable documents has now come into force with the publication of the state IT policy on 4 August 2009. Open source also received a major boost under the new policy, which commits the government to promote the use of and workforce training in Free and Open Source Software

(FOSS) in all public bodies in Assam, India's fourteenth largest state.

Munich completes migration to ODF

ODF is now Munich's primary internal document exchange format, with PDF used for non-editable files. According to the deputy leader of LiMux, the city's project for migrating its 14,000 workstations to free software, the migration involved 20,000 templates that were consolidated and converted into new templates, macros or web applications. The standard workstation for Bavaria's capital and Germany's third largest city now consists of OpenOffice, Mozilla Firefox and Thunderbird.

New York State to hold open government summit

The New York State Chief Information Officer/Office for Technology (CIO/OFT) and the NY State Archives will host the Open Government Summit in Albany, NY, on March 19, 2010. The one day summit will address the many hot-button issues in the "open government" discussion, including the meaning of "open government" in the digital age, operationalizing digital openness, and archival implications of digital records. A report published by CIO/OFT in May 2008 – "A Strategy for Openness: Enhancing E-Records Access in New York State" – recommended that the state identify open formats as a technology feature specifically desired by the state and integrate the acquisition of this feature of openness into the state's technology planning and procurement processes. More information concerning how to register to attend the Summit will be available shortly.

Australian Government 2.0 task force endorses open-standards based approach to public-sector information

Starting with the premise that "public-sector information is a national resource, and that releasing as much of it on as permissive terms as possible will maximize its economic and social value and reinforce a healthy democracy," the Australian Government 2.0 Task Force, in its report "Engage: Getting on with Government 2.0," has recommended that public-sector information should be free, based on open standards, easily discoverable, machine-readable and freely reusable.

Spain to propose government-wide standards for interoperability

Spain's national interoperability framework for eGovernment ("Esquema Nacional de Interoperabilidad en el ámbito de la Administración Electrónica"), published 29 January 2010 by decree in the country's official journal, establishes criteria and recommendations, together with the specific principles necessary to enable and encourage the development of interoperability in public administrations. They include the development at a later date of a catalog of technical standards that will enforceable by the government, the selection of which will be based on specific criteria set out in the decree. Public administrations are encouraged under the framework to use open standards and, where appropriate, standards that are widely used by citizens to ensure freedom of choice from competing technologies. It is recommended that documents and other electronic administrative services be made available via open standards under conditions satisfying the principle of technological neutrality.

PHP Conference 2010

Alain Williams

This one day event was in London on Friday 26th February. It had a good buzz and some interesting talks. After the opening keynote there were three parallel sessions, I report on what I saw.

Josh Holmes of Microsoft was evangelising about simplicity. What he meant was that developers often fail to understand what the user really needs and gives them too much information and options. This results in a confused and frustrated user who finds it difficult to do his job.

Stefan Priebsch talked about AntiPHPatterns – things that people do wrong when writing PHP programs. Much of this was about methods that are OK in small scripts but don't scale well – such as the use of global variables; a common way of avoiding these is to use a singleton (object), but this does not scale very well either.

Febien Potencier was trying to point out some of the ways that PHP 5.3 could help us to solve problems more elegantly and flexibly. Things such as `__get`/`__set` methods and lambda functions.

After lunch the PHP release manager Johannes Schlüter looked at some of what was available in PECL. APC an Xdebug are the ones that I already use and so remember most. There were some interesting features that he mentioned that I will have to explore.

Chuck Hudson talked about how PayPal is making its API more accessible and how they are trying to make managing payments easier for small and distributed organisations. For example you can 'front' a transaction with parts of the payment going directly to your suppliers; this works especially well if a refund is needed – PayPal does the work, you don't have to worry about extracting funds from a supplier who may have gone bust.

Lorna Jane Mitchell wrapped up the day. Her talk should have been titled "How to do your job well and spend more time in the pub". She was talking about providing web services that were usable and did not cause problems. Her essential recommendations are: good and complete design; good error handling and reporting; good documentation – all things that it is tempting for a programmer to take short cuts on and end up paying the price later on.

I enjoyed the talks, especially the more technical ones. It is a shame that the slides of the talks are not available on-line.

The conference was held at the Business Design Center in Islington, London. This used to be the Agricultural Hall with large rounded metal and glass ceilings looking like Alexanrda Palace or the back of St Pancras. The main concourse was filled with some kind of stamp exhibition.

The PHP conference was to one side with one large auditorium and two smaller ones. Coffees, lunch and exhibition were in a large central foyer that easily took the 450 delegates, even with enough chairs for those who bothered to walk up to the mezzanine floor.

The last of these that I went to was in 2007, I had not found the time since then. I enjoyed the day and will try to make time for it next year.

After the talks had ended there was an open bar courtesy of Facebook and several draws: endless books and entrance to various conferences for those who had returned comment forms, a trip to see PayPal in the USA. The main interest was in the two X-Boxes that Microsoft had put up. After failing to win anything I left before I became too drunk to drive to, or dance at, my folk club.

A Note On Setting Up the SheevaPlug Linux Plug Computer

Damon Hart-Davis

I ordered a SheevaPlug (512MB Flash, 512MB RAM, 1.2GHz ARM CPU, USB and Ethernet, running Ubuntu Linux, in a small 'wallwart' mains-plug package) with a view to getting my entire main Internet-facing server down to under 10W, using external plug-in low-power external HDD/SSD, I/O board and ADSL/WiFi, with the whole lot possibly powered from my off-grid PV system even in winter.

The plug arrived in August 2009 after a little prompting from me, and uses ~7W from the mains out of the box. (A subsequent order was rather quicker to my door at two weeks.)

This note will describe how I get the Plug initially set up, first as a slave/mirror of my Gallery, then running other services such as NTP and SMTP email and file storage, and finally becoming the master server.

The next stage will be to move from the current separate ADSL/WiFi using ~8W, to USB-powered ADSL and WiFi connectivity, necessarily limiting power consumption to well under 5W by USB power specs, and possibly nearer 3W total with luck, with Linux on the box providing any firewalling and routing services needed, and the on-board Ethernet port for wired (non-WiFi) access.

Power

The SheevaPlug contains a 'universal' mains power supply (AC 110–240V, 50–60Hz), and indeed can plug straight into a wall socket as if a power adaptor. I would expect it to consume ~5W from the mains when deployed this way, which would be significant saving from the ~20W consumed by the current laptop (when not on solar power, which is most of the time). One potential issue is that there is apparently no 'power fail' signal within the SheevaPlug visible at the Linux level which might allow a graceful shutdown, e.g. sync the storage and quit.

A longer-term aim is to reduce the drain on the mains to zero by powering this entirely off my small solar off-grid system (~120Wp with a 12V 40Ah SLA battery, thus ~250Wh storage, as of September 2009), even in winter, though possibly with some emergency mains fallback. To this end I purchased from Maplin a 3A Digital Car Power Adaptor (A79GW) which provides regulated outputs up to 3A from a DC 12V or DC 24V battery, with protection against over- and under-voltage (~11.5V cutoff apparently), and handily for me a voltage and current display (though the continuous consumption turns out to be higher than I want, ~1W–2W, since I can't turn off the LCD backlight which is a bit of a waste). I believe that internally the SheevaPlug uses 5V (or slightly higher) and thus the 5V setting on this unit should be suitable. Later I ordered a Texas Instruments PT78HT205V from RS, which claims to have efficiency of 80%–90%, max 2A (10W) output; but the power into it seems to be still 3.5W even when the system is quiet. What it doesn't have is lead-acid battery-sensitive low-voltage cut-off, so I'm routing it via my solar controller to provide that. With my off-grid system boosted to a little under 200Wp, ~150Wp south-facing, the system seems to be getting enough energy in to have run uninterrupted off-grid since I switched to the TI regulator, in spite of some very gloomy days, so I have high hopes of making it through mid-winter!

(With no load, withthe SheevaPlug mains supply disconnected from the CPU board, the mains supply itself draws ~0.8W according to my most sensitive meter, thus ~80% efficiency with a 4W load from the plug's innards.)

I've verified which Linux build is most power-efficient with the 'tickless' kernel as I use in my x86 low-power laptop.

Note that as of September 2009 simply plugging in a USB hub causes 10 extra wakeups per second (as seen by powertop) which raises power consumption by ~1W compared to the base ~4 wakeups/second.

When measuring DC power draw at (just over) 5V with my E-flite inline meter with 100mA current resolution, I'm seeing about 2.6W with an idle-ish plug with 8GB SD mounted card and Ethernet, 3.6W with a two-port hub plugged in and 4.3W with a 4-port plug (that gets slightly warm to the touch after a while).

I can also bump up consumption slightly by pulling a page repeatedly from the Apache server that I have installed.

With the external USB digital I/O board and a USB flash drive plugged in consumption is shown as 4.1W, which is very good indeed!

Without the USB stuff, but instead running a busy loop in the shell: `while true; do true; done` consumption goes up to 4.6W, which is still very good! (So going from a fully-idle system to one that wakes up frequently costs ~1W, and then running the CPU flat out eats ~1W more.) Doing a large Java compilation took the consumption to 5.1W in bursts, presumably reflecting the SD card activity (the SD card was warm), and took almost 50× longer than on my Intel Core Duo Linux laptop/server, about 25× of which is through lacking JIT (ie running fully interpreted), and the rest from having a single slower CPU (I parallelise when possible). Without the USB stuff, and unplugging the Ethernet, seems to reduce consumption by ~0.5W which is roughly what I would expect; a WiFi connection might use a similar amount.

I note that something like 0.5W is being wasted from continual break-in attempts across the Net on the untuned system, including the logging which will also be wearing out my Flash storage! So I've tuned settings a little to save energy and storage life...

15th September here had probably only 2/3rds the insolation of the typical mid-December day, and I have had some small extra loads on the battery, but there was still plenty of juice left which suggests that we may be fine mid-winter for power. The main Java app idles at about 3W+ (sans USB), with about 30 wakeups/sec. Note that by upgrading to Tomcat 6 with a built-in compiler for JSP pages I can get by with a JRE rather than a full JDK. Tomcat 6.0.20 on top of JRE 1.6 handles Java-1.6-dependent pages fine. (With the Sun 'embedded' SE JRE, total wakeups are at ~9/sec, uptime 0.02.)

I note that it takes about 25 minutes and 3Wh for the entire system to settle down to a low-power state, including the heavy start-up processing in the Java/Tomcat app.

USB Power Revisited

Expecially with a USB hub in place, it is important to have USB devices suspended when not in use, not only to minimise their own power consumption, but also to minimise (polling/wakeup) activity on any intermediate hubs which not only consume power themselves but also cause lots of CPU wakeups which waste power.

With `CONFIG_USB_SUSPEND` configured in the kernel then if everything downstream of a hub is suspended then the hub will be too. This is a very good start, but isn't quite enough.

A particular problem is the USB flash drive: if it is mounted then it cannot be suspended else errors may occur.

Using the `autofs5` package and a couple of ugly kludges I am able to automatically unmount flash-drive partitions when not in use, and when none of them are mounted, suspend the USB device, saving 1W+ of power.

I have a script that will identify a USB device under `/sys/bus/usb/devices/` by manufacturer and product.

For the automounter I use a 'program' (executable) map for the mount pount for the USB partition(s) which is run every time that a mount is required. Whenever that program/script is invoked it looks for the USB flash device and sets its power/level to 'auto' (to let the kernel manage it if possible, though in fact 'on' would be the same) then if all is well returns the map entry, thus making the filesystem(s) mountable if required.

Periodically I look for the indirect mount point being empty and then attempt to set the USB device to suspended.

(I've also disabled `hald` and `dbus` to avoid any unwanted USB device prodding.)

There is definitely a risk of a dangerous race which may end up suspending a mounted filesystem thus causing damage, so I only do this suspend if we are short of power and after issuing a general `sync` command. I also have a reasonably long timeout on the automount (>3× the `fsck` time) so the filesystem should only be unmounted if it now really is quiet. I also do some mutex locking between the 'powerup' and 'powerdown' operations, with the lock from 'powerup' lingering 15s after it returns in order to try to allow time for the mount to complete and be visible. (There's a further optimisation that, after the mount lock is released, an inotifywait is started to wait for a 'delete' on the mount point, and immediately tries a 'powerdown' if one is observed with ~30 minutes; at most one such 'watch' may run at once.)

I could probably improve the safety of this mechanism by, for example, building my own automounter which does the usb suspend/auto setting in a race-free way within the context of its own mutexes.

(Note that because my periodic reading of data from my k8055 interface board seems to wake up the USB flash (and presumably and other USB devices, I actually do the poll for powerdown just after the data read, so that might make co-ordination with a modified automounter more tricky, maybe requiring a filesystem-based lock to co-ordinate.)

With this mechanism in place, idle consumption drops back to ~2.5W, without it nearer ~4W with the 4-port USB hub.

More Detail: Automount, USB Suspend, and Other Horrors

Because the Linux usb-storage cannot suspend plugged-in storage (because for many of them, e.g. real spinning platters, it would be bad news) just plugging it in via a hub causes everything to be kept awake and chews through an extra ~1.5W (cf 2.5W for the entire rest of the system when fairly quiet) which is more than my off-grid system can spare on a dark day.

I use the automunter (`autofs5` package) to act as a gateway to the USB filesystem to mount it when busy and unmount it when not. As it happens I've picked the indirect mount point `/auto` (all this could do with a heavy dose of sensible renaming and tweaking) to mount the USB filesystem containing the Gallery files, with a symlink from its nominal `/rw/galleryDB` location. Here's the relevant line from `/etc/auto.master`

```
/auto /etc/auto.auto.sh --timeout=90
```

This causes the filesystem to get unmounted ~90s after it becomes idle. Here is the `/etc/auto.auto.sh` executable map source:

```
#!/bin/sh if [ "$1" = "galleryDB" ]; then
# Attempt to ensure that DT200 is power up, else quit.
logger -p daemon.info Powering up galleryDB storage...
/etc/suspend_inactive_DT200.sh powerup || exit 1
logger -p daemon.info Mounting galleryDB...
echo -fstype=ext3,rw,noatime,commit=600,data=ordered,barrier=1
:/dev/disk/by-label/galleryDB
exit 0
fi
# Error: unknown key.
logger -p daemon.err UNKNOWN KEY $1 for /auto exit 1
```

This checks that the correct key (mount/directory) is being requested and if so powers up the USB key if necessary (exiting if this fails) and returns the correct ext3 mount options including

```
noatime,commit=600,data=ordered,barrier=1
```

for wear-reduction and safety), using the by-label mount for robustness.

After a near-death experience apparently provoked by a bad USB cable/connection (couldn't mount, streams of horrible fsck errors, though apparently little actual corruption) I have done

```
tune2fs -e remount-ro /dev/sda2
```

to get the filesystem remount read-only if an error is detected to try to avoid causing any further damage but still giving read access which is all that is needed 99% of the time. (I also had to reinstate the journal with the `-j` option.)

The `/etc/suspend_inactive_DT200.sh` script locates the USB thumbdrive by manufacturer and product using a home-grown `/etc/findUSBdir.sh` script thus helping keep everything relatively insensitive to wherever the USB devices happen to be plugged and what hubs are involved, etc. This script can be called with powerup as here, or powerdown to force suspension of the USB device. An attempted powerdown is aborted if `/proc/mounts` shows anything mounted under `/auto`. The power-up retains the lock in the background long enough for any `fsck` and `mount` to complete. The script uses a filesystem mutex lock to prevent races between instances.

The powerdown is attempted periodically (being vetoed if inappropriate) and in particular after any other USB operation that may wake devices, such as reading from the K8055 USB digital I/O board.

Discussions with Alan Stern of Linux USB fame suggest that this may be the best that can be done with a 2.6.31 kernel.

I have suggested to Ian Kent of autofs5 that automount (the daemon) provide callouts before and after each mount and unmount, in the scope of its internal mutex locks so as to avoid races, where for example I might hang my powerup/powerdown operations. Typically the callout executable might always be passed simple arguments like:

```
{mount|unmount} {mountpoint} {mountedfile} {before|after|afterfailed}
```

e.g. for me I'd look for (to powerup):

```
mount /auto galleryDB before
```

and, to powerdown:

```
unmount /auto galleryDB after
```

and possibly:

```
mount /auto galleryDB afterfailed
```

Maybe the callout program could be indicated with a `--callout=PATH` in `/etc/auto.master`? This being open source, of course, I can hack the automount daemon myself!

Power To-Do

There are various things that may help minimise power (and system wear):

- Reduce spurious logging other than syslog, e.g. from Apache.
- Reduce Ethernet connection speed when conserving power, to 10Mbps max, for example using `ethtool -s eth0 autoneg off speed 10` to minimise power since nothing much needs go over 10Mbps here, and then re-enabling full-speed and auto-negotiation with the command `ethtool -s eth0 autonegon speed 1000`, when we think that we might be about to lose power and die.
- Find ways to minimise wake-ups (e.g. watching powertop stats) and spend more time in C1.

Already done for 2.6.31 kernel: Enable `CONFIG_USB_SUSPEND` which may save 1W or so. May be possible at run time, e.g. by adding to `rc.local` to allow all USB to autosuspend (quickly, but not instantly for better stability) something like:

```
echo 1 > /sys/module/usbcore/parameters/autosuspend
for f in /sys/bus/usb/devices/*/power/autosuspend;
do if [ -f "$f" ]; then echo 1 > $f; fi; done
for f in /sys/bus/usb/devices/*/power/level;
do if [ -f "$f" ]; then echo auto > $f; fi; done
```

Note however that `rc.local` may not be run at quite the right time to fully reduce consumption on some devices, so we may want to run some of it from cron...

Set `noatime` and a long commit (a la laptop-mode) on filesystems to minimise storage traffic and thus power consumption (and `data=ordered,barrier=1` for better crash-proofing).

Increase `dirty_writeback_centisecs` (to >1500) to reduce traffic, and use laptop-mode and other params to further increase I/O burstiness, (preferably matching up with the file-system commit interval) by adding to `/etc/sysctl.conf`:

```
vm.laptop_mode = 5
vm.dirty_ratio = 25
vm.dirty_background_ratio = 5
vm.dirty_expire_centisecs = 60000
vm.dirty_writeback_centisecs = 60000
vm.swappiness = 0
```

Note that we cannot have a `dirty_ratio` even approaching 50% as we expect the JVM to take up about (the other) 50% of physical memory. We don't want to be swapping at all if possible (and the initial configuration has no swap configured at all) but logically it may be needed in the future. Reduce spurious logging from syslog, and use the "-" option to avoid syncing to storage on every entry. Mounting `/tmp` on `tmpfs` to avoid any writes to storage. Disabled hal by moving `hal` to `hal.FCS` in `/etc/init.d` and creating in its place a file just containing `exit 0;` similarly for `dbus`, which also may be probing/waking USB devices.

Software

None of the software I use on my main server is CPU-specific, e.g. I have a mail server, static Web server, and a Java-based Web server, all of which should run my existing setup on top of ARM-based binaries rather than x86-based binaries, most of them statndard with the OS.

I have spent quite some time tuning the memory-heavy Java-based Gallery to run with a much smaller (halved) memory footprint and with some tweaks for a single CPU, but it remains to be seen as of 2009/09 if performance will be adequate.

The 'apache2' package is not part of the Sheeva base so I have installed it.

Minimising Memory Overheads

A major challenge of a 512MB machine, especially with zero or minimal swap, is limiting the memory consumed be various tasks and daemons or eliminating them entirely. (I don't want to be swapping at all, especially not to wear-prone flash, although some processes have much much larger virtual than physical memory use which may force me to set up some 'logical' swap to hopefully never be used.)

Done already:

- Disabled `hald` and `dbus` (for other reasons too).
- Tuned `apache2` (and Tomcat) to support relatively few connections and in particular few 'spare' servers/connections, which all eat memory, virtual and/or physical.
- Set following system properties for Tomcat to avoid 'leaks':

```
-Dorg.apache.jasper.runtime.JspFactoryImpl.USE_POOL=false
-Dorg.apache.jasper.runtime.BodyContentImpl.LIMIT_BUFFER=true
```

and at the end of `conf/catalina.properties` disable String cacheing with:

```
#tomcat.util.buf.StringCache.byte.enabled=true
```

Reduced the stack-space (and thus virtual memory) allocated per-apache2-thread from 8MB to 512kB with this inserted in /etc/init.d/apache2 in each of the two places (for start and restart) just before the daemons are started: `ulimit -s 512` thus reducing VM size from >100MB to maybe 20MB. Similarly for named/bind9.

More to save memory overheads: disable extra getty processed by commenting out the 'start' lines in `/etc/event.d/tty3` onwards.

To do:

- Possibly I should set `max-cache-size` for my BIND/named.
- Maybe set the default stack size (soft limit?) system-wide in `/etc/security/limits.conf` to 512kB rather than the out-of-the-box 8MB.

Firewall, Apache, Mail

I am replicating my firewall, Apache (`apache2`) and mail (`sendmail` and `qpopper`) configuration from my previous server, though with more attention paid to reduced resources such as connections, memory and storage life/writes.

Note in particular that the mail spool directories (`/var/spool/{mail,mqueue,mqueue-client}`) are displaced via symlinks from the root partition (on the plug's NAND flash) to the external SD card to reduce wear.

Long-term Data Retention

One issue to be thought about longer term is the issue of long-term storage on SSD/Flash (ie non-magnetic, non-optical) media: valuable data such as code respositories and the Gallery content might last indefinitely on magnetic media, but probably at least require explicit refreshing every few years on SSD/Flash.

It will probably be useful to automate off-site backups, e.g. of non-sensitive data/repositories to another co-lo host.

Bulk Storage

One of my applications needs a lot of storage (>50GB and growing, slowly), and I want to be able to power the storage from my off-grid solar, ideally via the plug's USB.

One particular problem with magnetic disc is the oomph required to 'spin up' even if using laptop-mode or similar I can keep it spun down much of the time. As far as I understand, bus-powered USB magnetic disc is not reliable.

To that end I have ordered a Kingston DataTraveler 200 USB 128 GB flash drive. I have put an ext3 (`noatime,commit=600`) filesystem on it, but in particular in order to minimise power consumption I explicitly automatically mount it on demand, unmounting it and explicitly suspending it when not in use to allow it and the USB hub to power down (avoids wasting ~1W). To make this less fragile I mount by LABEL or UUID, so its exact USB location will be less of an issue. (See `blkid` and `fstab UUID=` and `LABEL=` line formats.)

Using the numbers suggested by Theodore Ts'o to align filesystems to an SSD's erase block size to reduce wear and increase write performance, I used:

```
fdisk -H 224 -S 56 /dev/sda
```

to create cylinders of 49*128kB, then made a small (256MB) leading VFAT utility partition:

```
mkfs -t vfat -n PG2Kutil /dev/sda1
```

to allow the main/second partition to be cylinder- and erase-block- aligned, then to make that labelled main ext3 partition with better-than-default sizing (no space for expansion, directory indexes, sparse superblocks, an inode per 128kB which is a decent safety margin over the current gallery 500kB/file usage) I ran:

```
mkfs -t ext3 -i 131072 -L galleryDB -O ^resize_inod,dir_index,sparse_super /dev/sda2
```

which, with the current Gallery data copied in and taking about 50%:

```
Filesystem Inodes  IUsed  IFree IUse% Mounted on
/dev/sda2  975872 105540 870332   11% /auto/galleryDB
```

Note that disc sync operations, e.g. during an `svn commit`, are slower than on magnetic media, probably as is to be expected, but better read performance does speed up some critical tasks for me.

Digital/Analogue I/O

I currently use a k8055 USB card to gather digital inputs on the state of the battery (voltage level), wind turbine output, etc. The k8055 code is simple to build but needs (for example) `libusb-dev` installed first.

I have rigged up a very crude measure of raw battery volts using one of the analogue inputs (in units of 0.1V by running a 200k resistor to the 12V-nominal supply and relying on the internal 100k pot to give a 3:1 potential divider and then trimming the pot to get within a few units in the last place (ulp)) and using that to drive the internal 'low battery' flag. To eliminate some noise there is a 1u5 (25V) tantulum capacitor across the analogue input (ie the 100k pot). When the low-battery flag is set most applications that can switch to a power-conserving mode and only do the essentials and/or trade speed for power/energy.

Fast Reboot

With almost everything important loaded on the system (as at 2009/09/25) system reboot is fast, about 60s from `shutdown -r now` and about 30s from power-up (and the Marvell prompt/loader).

This abridged version of the author's original article is reprinted by kind permission. The full article can be seen at ***http://www.earth.org.uk/note-on-SheevaPlug-setup.html.***

The Art of SEO

Eric Enge, Stephan Spencer, Rand Fishkin, and Jessie C Stricchiola
O'Reilly Media
ISBN: 978-0-596-51886-8
608pp.
£ 34.50
Published: October 2009

reviewed by Paul Waring

Please see the combined review below.

SEO Warrior

John I Jerkovic
O'Reilly Media
ISBN: 978-0-596-15707-4
496pp.
£ 34.50
Published: November 2009

reviewed by Paul Waring

Search Engine Optimisation, or SEO, is something I'm often asked about when working on web sites for clients, so I was hoping that the release of two books on this often elusive and sometimes controversial subject would fill in some of the gaps in my knowledge. Broadly speaking there are three ways to get your sites listed prominently in the search engines: pay the search engine for a sponsored listing, use the latest tricks which Google hasn't cottoned on to yet, or invest in long-term methods which may take some time to show results. For the most part these two books only cover the last topic, although pay per click is touched upon in a few places.

Both books cover roughly the same ground, although if you are looking for information on social networking *SEO Warrior* covers this in more depth. However, the styles are noticeably different, possibly because of the number of authors involved. Personally I found the *SEO Warrior* title to be more technically orientated, with *The Art of SEO* pandering more to the business reader (it even goes as far as to explain SWOT analysis and other marketing jargon), which makes an interesting contrast in some places.

One thing to watch out for is that both books concentrate on the three "main" search engines, with a preference for Google. This makes sense in the US and European markets, but if a significant amount of your business takes place in countries such as Russia and China, you may need to reference other books or web sites to take account of the dominant search engines in those countries. The information on social networking falls into a similar vein – extremely useful for sites selling mass market consumer products, less so for your local shop.

Both books also seem a little lengthy, often spending a bit too much explaining why SEO is important without getting down to the details of how to get a site to rank highly for particular keywords. There is a lot of irrelevant information, such as where to host your site (the company and operating system used should have no bearing on how your site ranks), and nine pages of "praise for *The Art of SEO*", which requires filtering out in order to get to the useful parts.

Overall, I would suggest that one of these books may be useful but most readers are unlikely to make use of both, due to the overlap of content and difference in approach. If you're prepared to filter out the irrelevant sections and treat them as books to dip into when you want to learn about a particular topic, rather than being read from cover to cover, there are some useful tips and tricks to be found in both books. However, they should be read in conjuction with the latest SEO web sites and specific advice from individual search engines if you want to optimise your site's ranking.

Programming Google App Engine

Dan Sanderson
O'Reilly Media
ISBN: 978-0-596-52272-8
400pp.
£ 34.50
Published: November 2009

reviewed by Mike Smith

When I looked at Android development a few months ago for previous book review, I also installed the Google Eclipse plug-in out of interest. There it has sat, doing nothing, until I spotted this book requiring a review too. So I have now returned to actually start using it.

Incidentally, there are PHP plug-ins for eclipse too that provide a better development environment than vi!

The book covers both Python and Java programming environments for Google App Engine (GAE); pointing out that Python has a few more utilities as it was supported first. Nevertheless because of the history above I have started my testing, and review, using Java.

The introduction is great for novices like me. All about what GAE is, and some differences to expect – particularly the datastore services that are not like relational databases most of us will understand.

I worked through the example application in chapter 2. This introduces a whole bunch of concepts, step-by-step, building our skills as we go. The phases were along the lines of initial setup; authentication; datastores; using memcache; registering and uploading the application and using your own domain name. Testing is all performed locally within Eclipse, and I have my fair share of self-inflicted bugs to sort out, but it's a joy with the SDK and Eclipse environment.

Uploading the application was a breeze – just a press of the toolbar button in Eclipse too (with the requisite XML preparation as so usefully outlined in the book).

I actually host my main domain (used for testing here) with Google, using Google Apps. Application management is normally performed at `http://appengine.google.com/` but I couldn't understand why, when I was registering applications, they weren't appearing. Quite a puzzle.

It turns out that Google-hosted domains are a special case and I should have been using the URL `http://appengine.google.com/a/<domain>/` for administration. This is covered in the book, if I had only noticed!

Later chapters in the book cover some of the subjects we touched on with the example application in more detail. I liked this approach; we get up and running quickly, then return to reinforce and understand in more depth those topics briefly introduced at the beginning.

All the way through there is a dual approach, covering Python and Java environments, and in addition to those introductory topics we cover email and instant messaging with XMPP, bulk data handling, task queues. We round off with a chapter on Django which of course is Python-specific. It's quite a short chapter with just a few framework principles and how in install and configure. And finally some more details on application management – uploading, versions, administration etc.

So I found the book to be a really good introduction to GAE and it got me off the ground with my first application. I thoroughly recommend it for those of your who in a similar position. (Just starting out and need some hand-holding.)

Inside Cyber Warfare

Jeffrey Carr
O'Reilly Media
ISBN: 978-0-596-80215-8
240pp.
£ 30.99
Published: December 2009

reviewed by Mike Smith

A very exciting title, but this isn't your usual O'Reilly book. It's an account on the status of cyber warfare today, and as stated in the book itself, could be considered a set of separate essays on the matter. It's not that big at around 200 pages.

Reading the first few chapters I was frustrated by the return, time and time again, to the same few examples of cyber warfare – primarily the Russian-Georgia conflict, which to be fair is the main premise on which the book is based (and Project Grey Goose). Chapter 3 asks what is an act of cyber warfare, but doesn't answer it (I supposed that's the point – there's no legal position) and goes on to discuss the use of social networking within military networks. I didn't really get the link.

We next get into a dry legal debate about possible responses to cyber attacks.

Chapter 5 was a refreshing improvement – we get into some practical information on how the Korean DDOS attack in 2009 using MyDoom and nearly 167,000 zombies where performed. Then we mention the Russia-Georgia war again. Arrghh!

Things are on the up with a discussion on social networking sites – Twitter, Facebook and the like – together with the inherent risks. An interesting comment that the spooks are in there recruiting operatives. Common sense, but not something I had considered previously. We continue with a discussion on money trails, and an interesting (but not very advanced) dissection of "bulletproof networks" – I'm sure we've all used `whois` before.

We've over half way through now, moving on to organised crime.

Chapter 9 is oh so short, covering forensic tools and techniques. AS numbers; BGP; Darknets. This is what I'm interested in... and could have read a whole O'Reilly book on this (I'm sure there is one!)

A few other chapters round off the text. As you can probably tell I really didn't get on well with this book. Seemingly random titles followed by brief paragraphs, and disjointed chapters often covering the same ground. I think I was hoping for a far more in-depth technical discussion.

At least the UKUUG will be happy with one recommendation near the end: switch from Microsoft Windows to Red Hat Linux!

---

flex & bison

John Levine
O'Reilly Media
ISBN: 978-0-596-15597-1
304pp.
£ 22.99
Published: August 2009

reviewed by John Collins

This book is a rewrite of the earlier book *Lex & Yacc* by one of the original authors.

Flex is a rewrite of Lex and Bison is (after a long history) a rewrite of Yacc.

The purpose of Flex (or Lex) is to group characters in a stream of text data into "tokens" such as words, strings, numbers and keywords. Provided with a file describing how the tokens are composed, it produces a C function that will do this job with the input set up from a calling program. This process is called lexical analysis.

The purpose of Bison (or Yacc) is to generate a "grammar" function from a specification, which takes a series of "tokens" and recognises various structures made up by sequences of tokens. This process is called syntax analysis.

Very frequently Flex and Bison are used together, Flex to worry about the individual characters returning tokens, and Bison to compose these into structures for onward processing by the rest of the program.

Writing lexical and syntax analysis functions are well-known to be tedious and error-prone operations and having automatic processes for helping with these tasks is helpful.

It is worth remembering that neither program works completely satisfactorily in all circumstances. Examples of where Flex is hard to make work satisfactorily are the handling of "include" files and multi-line strings (especially with escaped delimiters) and comments although they are a bit easier in Flex than its predecessor Lex.

Bison is limited in that it by default understands only what are known as LR(1) grammars – ones which can be understood and alternate constructs distinguished by reading from left to right looking ahead at most one token. An example of data which Bison cannot generate a parser for is its own input. (An option allows the user to specify "GLR parsing" which is more comprehensive, however).

Nevertheless both are extremely useful in numerous situations where blocks of text are to be deciphered as well as forming the "front end" of various compilers.

This book is 100 pages shorter than its predecessor although it contains much the same material under similar headings. Some of that is accounted for by the closer spacing of the lines but quite a few paragraphs and sentences have been trimmed, I don't think with any great loss to the sense. The book does tell you about using Flex and Bison with C++ which the previous book didn't.

The book introduces Flex and Bison at a basic level and then has a 36-page section about SQL which much of the rest of the book discusses how to parse in the context of providing a reference to Flex and Bison in turn.

I personally don't like the alternation between Flex and Bison as you go through the book. I'd prefer an introduction to each and then a full reference of each myself but other people may disagree. I also dislike so much of the book being about a huge SQL parser and even having the index refer to discussions of SQL.

Another issue might be that there are systems with Lex and Yacc on and it might be helpful to have an appendix with the extensions offered by Flex and Bison so that the user can if necessary avoid these.

But those personal gripes apart, I'm sure this is a good and comprehensive reference book for those needing to use those tools.

Core Data: Apple's API for Persisting Data on Mac OS X

Marcus S Zarra

Pragmatic Bookshelf
ISBN: 978-1-934356-32-6
256pp.
£ 25.50
Published: October 2009

reviewed by Graham Lee

Core Data is one of those APIs where it's obvious how to get the really simple stuff done, but then once you need to customise the way it works or do something a little more complex it becomes much more difficult. I suppose that's an exact expression of the "simple things simple, complex things possible" philosophy that Apple's framework developers seem to have embraced, except that any non-trivial application will quickly move out of the "safe" territory.

Apple's own documentation is characteristically bimodal. They provide a useful conceptual guide to get developers up and running, and a very comprehensive reference for the whole API. But understanding how the methods work doesn't help if you don't know what you need to do or why you need to do it.

Marcus Zarra's book fills this gap very nicely. By developing a single application through the course of the book, Marcus takes his readers away from the simple act of creating a Core Data application very quickly and into encountering real-world problems, such as migrating between different versions of the managed store and performing custom sorting and fetching operations. The sequence in which these chapters are presented forms a logical story, with each part being presented as the solution to an issue the developer of the example application really needs to address.

The last few chapters of the book move away from this story into a collection of "recipes" for advanced work, such as synchronising between different computers and integrating with operating system features like the Spotlight search engine. Many of these recipes still use the example application (which is itself about recipes) as a starting point, so they don't lose the feel of applying to real scenarios. For example, the chapter on iPhone Core Data migrates the (Mac) application worked on throughout the book over to an iPhone app.

So, in short, this book is well-written, and fits a gap in the official documentation. I'd recommend it to anyone using Core Data in their application development.

CSS the missing manual (2nd edition)

David Sawyer McFarland

Pogue Press
ISBN: 978-0-596-80244-8
560pp.
£ 26.99
Published: August 2009

reviewed by Lindsay Marshall

I've always been slightly puzzled by the "Missing Manual" series. Some of the books really do fill a gap but others just seem to be hanging around with the cool kids and just pretending to be missing. This book is one of the latter. Let's face it, there are dozens of books on CSS – I have a shelf full of good ones, heaven alone knows how many bad ones there are (probably not as many as for Java though). So this is not really missing. And I am not sure what box it should have been in either – this the trouble when you start a brand, you have to stick with it through thick and thin no matter how daft it sounds.

But, what of the book? Well, in a bright light it looks OK, but in poor light the text is fuzzy and the cover a bilious green (damn you presbyopia!). There are wide margins on the pages too and a slightly odd layout which means that it weighs in at over 500 pages. It could have been thinner I think.

And the content? Actually the content is really pretty good. There is coverage of all the major topics: floating, tables, forms, print, and all the usual stuff on padding and margins. There is some discussion of reset stylesheets and a tiny mention of grid systems but there could have been more on those I feel, particularly as this is a second edition. As always with books on anything web related, you can find all the information contained here, and much, much more, on the net, if you can be bothered to look. I do like to have a hardcopy manual to hand (though I confess to using the web more and more these days) but whilst most of the stuff you need to know is in this book, it is not really in a manual form. This goes at quite a gentle pace and steps through examples slowly and carefully making it much more of an introductory text than something seasoned web designers would use as it would just take too long to find the particular nugget of information they needed. However, beginners might be daunted by the size of the book (which, when I think about it, reminds me of the Dummies series but with much smaller type), so I am not entirely sure who will buy this. But if you think this sounds right for you I can definitely recommend it.

The Art of Community

Jono Bacon

O'Reilly Media
ISBN: 978-0-596-15671-8
400pp.
£ 30.99
Published: August 2009

reviewed by Roger Whittaker

Jono Bacon is perhaps almost uniquely qualified to write this book. He came to Linux as a student, with no prior knowledge of Unix, initially as an end-user "*clueless at the prompt*" as an old Linux Journal column's title so nicely put it. (His description of his first steps with Linux rang a bell with me: wondering what to do next when seeing `darkstar login:` on the screen after installing Slackware for the first time.)

What inspired him were the possibilities he saw in the open source development model: he saw the power of the communities that were creating Linux, KDE and other projects and was convinced of their long-term viability.

That conviction led him to a life in Linux and Open Source software: as a freelance journalist, a LUG organiser, as a founder of the LUGRadio and LUGRadio Live, as inspirer of Jokosher, and to a full-time job at Open Advantage and now (since 2006) for Canonical as the Ubuntu Community Manager.

He has come a long way in the last ten years or so, but throughout that time his main focus has been on the communities that he has been a part of and how to help them to work as effectively as possible. Although the book draws heavily on his own experience, it is far more than (as I admit I feared it might be) a book of anecdotes or personal success stories: he has thought deeply about the issues involved and articulates his conclusions clearly and thoughtfully but in unpretentious, entertaining and readable language. He also writes frankly and honestly about some of the mistakes he has made in the past and what he has learned from them.

Although such communities can be self-organising and self-perpetuating, they can also self-destruct. Understanding the dynamics of communities is valuable for anyone who values the output of a community that they belong to, but particularly for anyone who aspires to be a leader or guide.

The main message of the book is that creating and sustaining a community that does something useful is both hard work and requires a lot of deep thought. The successes of Linux and many open source software projects as well as Wikipedia and other high profile online communities have created an atmosphere in which *crowdsourcing* has become a buzz-word and there have been failures as well as successes particularly among attempts to create living communities around products or projects that were previously managed internally by corporations. This fact in itself shows the need for an understanding of the art of community.

Jono notes some of the necessary conditions for success: open communications, fair appreciation and licensing of work, low barriers to entry and above all *belief* in the shared aims of the community. Among the pitfalls he mentions are undirected and inconclusive discussions about detail (including *bikeshedding* and *resource fetishism* – look those terms up if you need to). He stresses the importance of choosing communications media that make a good fit with the types of community members they are aimed at. He discusses the best ways of dealing with and (better) avoiding whenever possible the inevitable conflicts that will arise, and distinguishing between the genuinely destructive or poisonous personalities and others whose modes of communication may be a mask for misunderstandings that can be overcome.

Every section of the book is illustrated by examples of what has and has not worked in practice. As the author points out early on, the knowledge that this book covers is soft science (maybe even at some level common sense), but we have all seen enough examples of failure in these areas to know that getting it right (particularly with online global communities of geeks) is not easy and requires both thought and experience. Jono has done that thinking and has that experience.

The book is published under a Creative Commons (Attribution-Noncommercial-Share Alike) licence and the entire text can be downloaded in its entirety as a PDF from the book's web site. But if you are interested in the subject matter I would urge you to buy a paper copy of the book, both to encourage publishers to publish materials that are freely licensed and because this is a book you will want to idly pick up and dip into from time to time, something you are much less likely to do with a PDF file on your desktop.

Learning Python (4th edition)

Mark Lutz
O'Reilly Media
ISBN: 978-0-596-15806-4
1216pp.
£ 42.50
Published: September 2009

reviewed by Roger Whittaker

One of the things that most children find terribly annoying about adults is the tendency of parents' friends and distant relatives to exclaim "how you've grown" on meeting them after a period of years. Since being fully grown, I have always tried to suppress this tendency in myself. But when this "doorstep" arrived on my doorstep, I could not quite manage to suppress that exclamation.

Admittedly the slim 384-page version of this book that has been sitting on my bookshelves for all these years is the first edition, relating to Python 1.5 and published back in 1999. This fourth edition is more than three times the size and covers both 2.6 and 3.0/3.1. This fact in itself adds a considerable percentage to the book's volume because there have to be parallel explanations and examples in many places where the language has changed between 2.6 and 3.x.

Although the book has grown over ten years from 10 chapters to 39 chapters (plus appendices) it is still essentially what it was then: a very clear and readable introduction to the Python language which makes no assumptions of prior knowledge, and uses simple very short code examples to explain the concepts as it goes along. It does not (like some impenetrable programming books) try to lead you through some imagined project as a way of teaching the language – that approach in my experience almost always fails. Nor do later chapters depend heavily on the material in earlier chapters, and when they do, the cross references are good. It makes a good general reference book, although that is not its main aim, and is much more readable than the author's other doorstep "Programming Python" which I find suffers slightly from the problem that you sometimes have to follow through the logic of examples you are not interested in to get at the content you really want.

This book is good on the changes in the language between 2.6 and 3.x, and I shall no doubt be referring to it to understand those better in future. It has good sections on new string `format` method in 3.0 and on changes in Unicode handling. For those with brains larger than mine the last two chapters of the book are on Decorators and Metaclasses.

Don't be put off by what I wrote above about the book's size: if you want to own one book on Python, this is still the one.

Head First Programming

David Griffiths and Paul Barry
O'Reilly Media
ISBN: 978-0-596-80237-0
448pp.
£ 38.50
Published: November 2009

reviewed by Roger Whittaker

First, let's be clear: no member of UKUUG is likely to want to buy this book *for themselves.*

I had noticed these "Head First" books before, but not really paid any attention to them: they seemed to be slightly trendy introductory books with lots of pictures and odd graphics.

I have reached (and passed) the age when the children of friends and acquaintances are teenagers, and more than once I have been asked for advice when those teenage children want to learn something about programming.

The most important bit of advice to give in this situation is. . . they should learn Python. One young boy had already bought a frightful book on Java when I saved his enthusiasm and possibly his sanity by giving this advice. He hasn't looked back, and has done some impressive things with Python since.

This book is an introductory *programming* book that uses Python 3. It explicitly states that its purpose is to teach programming, but through the medium of Python, and that it is not "Head First Python". Its introduction also meets head on the questions about what all the graphics and pictures are all about: this series has a philosophy about engaging both sides of the brain, and using pictures and chatty text to draw the reader in.

My initial reaction to the look of the book was negative: all the slightly twee "ironic" stuff and cartoon-like illustrations offended my sense of what's proper. But when I started reading the text, it was clear that the book does do what it says on the tin: it's a first introduction to programming through the medium of Python, and it does it well. I also had to admit from an educational point of view that the book is well laid out and leads the reader through the examples in a well-paced and sensible way.

So this book can certainly be recommended as a gift to a young person who has never done any programming and would like to start.

The Sustainable Network

Sarah Sorensen
O'Reilly Media
ISBN: 978-0-596-15703-6
368pp.
£ 22.99
Published: October 2009

reviewed by Andy Thomas

Standing in the blast of hot air behind a rack of servers, I sometimes wonder if what we are doing in data centres is somehow very bad – bad for the environment, bad for the planet, bad for humanity even. Just think of it, using all this electric power – maybe 4, 8, 10 kilowatts or more in a single rack – simply to push a stream of tiny electrical impulses signifying one and zeros out along a wire seems slightly obscene, especially as elsewhere in the building there will be a cooling plant using almost as much electricity again just to get rid of the heat. Under no law of physics can this be described as efficient conversion of energy from one form into another desired form and I can think of no other industry where a huge amount of energy is expended in creating a nebulous and almost invisible 'end product' that contains no intrinsic energy value. So it was with eager anticipation, tinged with some guilt, that I opened Sarah Sorensen's book, *The Sustainable Network.*

A technical networking treatise it is not – aimed squarely at the thinking individual who is aware that something needs to be done within our lifetimes to secure the future of planet Earth as we know it, the book sets out to persuade the reader that its salvation lies in the global IT network. And here it succeeds very well, looking beyond the technical aspects of data networks at the socio, economic, political and above all, environmental impact.

Admittedly, as a techie I was slightly put off at first by the sprinkling of corporate buzz-words – leverage, growth, empower, strategic for example – and the extensive quoting throughout the book from the kind of business reports from Gartner, McKinsey, et al that only well-funded corporates ever get to see, not ordinary mortals like you and me. Right from chapter one, the author wastes no time in getting the message across that the network enables so many different things, opens up new opportunities and links communities – repeated so often interspersed with a plethora of statistics and references to business intelligence reports that it almost becomes a mantra. By chapter 7 or 8 I was feeling a little dizzy and beginning to wonder whether I'd opened a management 'white paper' by mistake.

But at this point, the book starts to get into its stride (yes, chapter 9 – there are no less than 41 chapters in this 350 page book). The author's tremendous enthusiasm, indeed passion, for her subject begins to shine through the corporate fog of earlier chapters and the book gets down to brass tacks. Ms Sorensen postulates the Sustainable Network Law which states that "the more broadband made available to network users, the faster sustainable network innovation occurs" and she goes on to explore and present the evidence to support this. Now the heavy layering of statistics and reports begins to become clear – without valid factual evidence, the law simply would not stand.

Subsequent chapters explore wired, wireless and mobile networks along with the energy grid, the electricity supply network without which data networks would not exist in their present form. We then look at carbon footprints (something the IT industry is not renowned for reducing, at least not directly but more on that later) and then we move on to environmental issues where she introduces two words new to me – dematerializsation (for example, buying your entertainment as a digital download rather than on CD or DVD bought from a shop or sent through the post along with all the attendant packaging, paper invoices, etc) and detravelization (removing the need to travel to conduct business meetings, etc).

Then we look at the broader picture of where data networks fit into the business, social and political fabric of present day society and the growing issues of cyber-warfare and the need to protect and nurture this valuable resource. This is what the book does well (at least for this reviewer) – at the sharp end of IT, faced with switches, routers and servers in racks you just don't get to see, appreciate or even think about the wider aspects of the network.

But throughout the book runs this common strand – not only can the network become sustainable, it can also be the single most valuable asset for shaping our future on planet Earth. Therein lies the way forward to improving energy efficiency, reducing waste, expanding horizons both economically and socially and most importantly for the 21st century when political, ideological and religious intolerances pose an even more immediate threat to global stability than environmental issues, global data networks bring people, communities, ideas and whole countries together.

This book is not for the technical junkie but is very readable and thought-provoking. For instance, I was surprised to read that there are 100 million mobile phone handsets in Brazil, a country I thought was relatively poor. Surely this exceeds the entire population of Brazil? Then I googled a bit and was astonished to find not only was the population 192m in 2008 but in a space of just 38 years it had doubled from its 1970 level of 96m. The frequent references to reports and statistics serve to show just how thoroughly Ms Sorensen has researched her material for the book.

Reviewing this book has been an enjoyable challenge not least because of its ambitious remit – this is one book that should be on the desk of every CEO (or in Britain, every MD) as only the captains of industry have the necessary funds and clout at the corporate level to make the sustainable network happen. But we all have our own bit to do too by encouraging change. One delicious and memorable moment whilst reading chapter 15 on tele-commuting on my daily commute; the signals failed and I got through 4 more chapters before the London Underground train even moved again. Three chapters later I finally reached my destination over an hour late but for once it was not a waste of my time.

Well laid out with a lot of short and sweet chapters, the book rounds off with a glossary of networking terms, no less than 23 pages of references (having links to a well-known network equipment vendor, the author has far more access to background material than most of us will ever have) and a comprehensive index.

And finally, after reading that the WWF has stated that "a doubling or even tripling of IT sector CO2 emissions could be considered strategic if the overall benefits of ICT are fully realised in other sectors", I no longer feel quite as guilty about the power consumption in rack H4 at Telehouse.

Contributors

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

Damon Hart-Davis is an IT consultant with a long-standing interest in small and low-power devices and various flavours of UNIX, though he only really ‘got it’ about the need for energy efficiency at home and in business in the last few years, which resulted in the SheevaPlug work.

Graham Lee is an independent Mac and iPhone developer and security consultant, based in Oxford. His own book, ”Professional Cocoa Application Security”, will be published by Wrox Press in June.

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

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

Mike Smith works in the Chief Technology Office of a major European listed outsourcing company, setting technical strategy and working with hardware and software vendors to bring innovative solutions to its clients. He has 20 years experience in the industry, including mid-range technical support roles and has experience with AIX, Dynix/ptx, HP-UX, Irix, Reliant UNIX, Solaris and of course Linux.

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 and 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.

Alain Williams is an independent Unix and Linux consultant, running Parliament Hill Computers Ltd.

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`

Roger Whittaker
Newsletter Editor
London

Alain Williams
UKUUG System Administrator
Watford

Sam Smith
Events and Website
Manchester