



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 4

ISSN 0965-9412

December 2010

Contents

From the Secretariat	3
Chairman's Report	4
OpenTech 2010	4
FLOSS UK Unconference	5
Act on ACTA	6
Book review: SQL Antipatterns	8
Book review: HTML 5: Up and Running	9
Book review: Using SQLite	10
Book review: Being Geek: The Software Developer's Career Handbook	11
Book review: Ubuntu for Non-Geeks (4th edition)	12
Book review: Programming the Mobile Web	12
Book review: Javascript Cookbook	13
Book review: JavaScript Patterns	13
Book review: LPI Certification in a Nutshell	14
Book review: MongoDB: The Definitive Guide	15
Book review: Cooking for Geeks	16
Contributors	17
Contacts	19

From the Secretariat

Jane Morrison

The minutes from the Annual General Meeting held on 23rd September have once again not been circulated to members via hard copy. All members were emailed on 5th October and advised that the minutes and associated documents could be found on the web site at:

<http://www.ukuug.org/events/agm2010/>

Of course any one who wants a printed copy should contact me here at the office.

Current Council members are: Paul Waring, Howard Thomson, Phil Hands, John Pinner, Niall Mansfield, Holger Kraus and Kimball Johnson.

Kimball Johnson has recently joined Council and we look forward to working with him.

John Collins resigned his Council place at the recent Council meeting held on 11th November.

Paul Waring continues in the role of UKUUG Chairman and Howard Thomson continues as Treasurer.

We are currently putting in place a full schedule of events for 2011 and 2012 – to date the following have been agreed:

Perl Tutorials – 2 day Intermediate (including practical sessions) – 15th and 16th February and 2 day Advanced (including practical sessions) – 17th and 18th February – London (see flyer inserted with this Newsletter)

Spring 2011 Tutorial and Conference – 22nd, 23rd and 24th March 2011 – Leeds – (see insert with this Newsletter)

‘Debian Packaging’ Tutorial – by Phil Hands – 19th May 2011 – London

OpenTech – May 2011 – London

Un-conference – Saturday 15th October 2011 – Manchester

Spring 2012 – 20th, 21st and 22nd March, Birmingham

UKUUG is also planning more tutorials for 2011 – if you have any particular topics that you think would be of interest please let us know.

I would like to remind you of a couple of benefits we have in place for members – we have a page on the UKUUG web site – <http://www.ukuug.org/consultants> – which lists UKUUG members who can provide consultancy services. This is free to members – take a look and send me your details if you are a Consultant and get some advertising for free.

Also, I would like to remind you that in addition to the O’Reilly book discounts (40%) we also provide other discounts for members these include: 2nd Quadrant (10% off PostgreSQL courses), GBDirect (10% off various courses) and Bytemark Hosting (10% discount on all virtual machines). Take a look at <http://www.ukuug.org/discounts>.

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

UKUUG would like to thank the continued support of our sponsors. In 2010 these included Bytemark, Google, IBM, Novell and Sun. Their involvement helps us to keep event fees down, while providing a means for sponsoring companies to communicate with some of the brightest people in the industry.

I would like to take this opportunity to wish you all a very Happy Christmas and a Peaceful New Year.

The Secretariat will close on 20th December and re-open to January 4th 2011.

Please note the copy date for the next issue of the Newsletter (March 2011) is 18th February.

We are always looking for interesting submissions from members, so if you have any news, articles etc. please send copy direct to newsletter@ukuug.org.

Chairman's Report

Paul Waring

Recent and Upcoming Conferences

We recently held our first one-day unconference in Birmingham in mid-October under the FLOSS UK banner. The event was well attended, including a number of people who had not been to our events before. Simon Phipps provided the keynote speech and various delegates offered to deliver talks or lead discussions on the day. Plans are already underway to run a similar event in October 2011, this time in Manchester – more details will be available on the website once the venue has been confirmed.

Our Spring conference in Leeds is also approaching fast, and the programme is currently being prepared. There is still time to submit an abstract if you would like to deliver a talk, and a booking form is also enclosed with this newsletter. I hope you will be able to join us for another enjoyable three day tutorial and conference programme.

Changes to Council

We have recently welcomed Kimball Johnson onto Council, following his election at the AGM. Kimball has attended numerous UKUUG events in the past and is already involved in the preparations for the Spring conference.

Unfortunately, John Collins has had to step down from Council for personal reasons, and I would like to take this opportunity to thank him for the work he has done for UKUUG over the past few years. John's departure leaves a vacancy on Council – if you would like to get involved with helping us to run UKUUG and its busy event programme, please do get in touch via office@ukuug.org.

Membership Direct Debits

As a result of feedback from members, Council is currently looking to introduce the ability to pay subscriptions by direct debit. At present we are working on several options for implementing this payment method, and hope to have this in place in time for the annual subscription requests in January.

FLOSS UK Branding

Based on feedback from previous conferences which suggested that the "Unix" part of the UKUUG name no longer covered the wide variety of topics at our events, and the positive response to our unconference branding in October, Council has taken the decision to gradually transition to using the FLOSS UK name for our events in 2011. We will still be running the same mix of conferences and tutorials members are used to, and the same familiar faces will be in attendance, but we are hoping that the broader remit suggested by the new name will encourage people who have not previously been to one of our events to attend.

OpenTech 2010

Paul Waring

On a sunny Saturday in September I headed out for another outing of OpenTech – this time having sensibly opted to stay over the night before instead of risking an early train from Manchester. A regular event which has run for the past few years, OpenTech has a series of talks on a diverse range of subjects, and is held at the University of London Union.

As always, the choice of which session to attend meant a difficult decision had to be made. In the end I opted to start with the session based around two .gov.uk projects, concerning access to data and legislation. Both talks gave an interesting insight into how public sector data can be made available in ways which make it easy to consume in both human and machine readable formats.

The second session involved a discussion group entitled ‘How UK Universities are Using LinkedData’. This was more informal than the talks and seemed to work really well, with lots of useful ideas being thrown into the pot. Hopefully this style of session will be available again at next year’s event.

After lunch the session I attended was focused on scientific topics such as ownership of genome data and OpenGeoScience, which I have to confess went slightly over my head. Following these were talks on Modern Perl (Dave Cross, amusing and informative as always), Modern FreeBSD and Modern Java. These provided some useful information and updates on how these three languages/systems, often seen as somewhat antiquated and behind the times, still have a lot to offer.

The final session which I managed to attend involved a set of talks about newspapers. Phil Gyford’s talk entitled ‘Today’s Guardian’ was particularly interesting – Phil has made a simple website (<http://guardian.gyford.com/>) which fetches all the articles from the Guardian for the current day and displays them like pages in a newspaper. Not only is this an innovative use of the Guardian’s API, it also offers a really simple way to ensure that you have read the day’s news, without needing to keep checking the Guardian’s website for new stories.

As with every OpenTech, the networking over tea in the breaks – or a pint in the bar afterwards – is what makes the event have that ‘buzz’ feeling. Whilst the talks are always top-notch and informative, it’s when you chat to the speakers and other delegates outside the sessions that ideas are discussed and new projects are kick-started.

Overall, the event was a great success and thanks are due to David, Emily and Sam for making it happen, as well as all the speakers for giving up their Saturday and everyone else who chipped in to help. Plans are already underway to run OpenTech again next year, so keep an eye on the website for details. In the meantime, recordings of most of the talks from this year’s event are available online at: <http://www.ukuug.org/events/opentech2010/>

FLOSS UK Unconference

Roger Whittaker

This event, held in Birmingham on the 16th October, was UKUUG’s first unconference, and was held jointly with various other groups, including the PyCon UK Python group and the UK TeX Users group, which held its annual general meeting during the day.

The event took place in the handsome Birmingham and Midland Institute building in central Birmingham: an excellent example of Victorian institutional architecture, with a re-fashioned large lecture hall that was very suitable for the plenary sessions.

The day began (slightly late) with a keynote by Simon Phipps on various aspects of the current state of free and open source software. As so often in such talks, there were some strongly optimistic elements and some warnings and fears of dire threats.

On the positive side, the strength and growth of an “adoption led” approach to software procurement was discussed, and how the pragmatic advantages of such an approach are changing the way that software solutions are chosen and implemented across the board.

On the negative side, Simon was deeply concerned about the threat of US “intellectual property” laws being spread to the rest of the world without any real discussion through the ACTA treaty.

John Pinner then took the chair, and created a list of talks from those offered by people present. The titles were written on “post-it” notes and these were later arranged on a board in rows (represent-

ing time) and columns (representing place) so we instantly had (in theory) something similar to the programme of a traditional conference with three tracks.

One of the first sessions that took place immediately afterwards was in a way a continuation of Simon's talk: a discussion on software patents and ACTA led jointly by Simon Phipps and Russel Winder. Russel very kindly agreed to contribute an article which appears below.

In practice I found that this form of organisation did not quite work out as well in practice as in theory, because I was one of those who wanted to switch between tracks (TeX to Python in my case), and when I arrived back in the Python room for a session that was about to take place according to the board, I discovered that it had already happened.

Despite one or two such glitches, the day went very well, with some interesting talks and a particularly active Python track. There was a good, friendly atmosphere and the venue was very suitable. The event was assisted and sponsored by the Linux Emporium and by Bytemark.

I hope that unconferences of this type will become a regular addition to UKUUG's programme of events, and John Pinner and his team did a great job in organising the event (yes, an unconference requires a considerable amount of organisation!).

Act on ACTA

Russel Winder

It cannot have escaped any software developer's or systems administrator's attention over the last few years that the big companies have turned more strongly than ever to patent cross-licencing or litigation as a core part of their business models – at least in the USA. Hardly a day goes by without news about some high profile patent case involving software, user interface or business method. Apple, Microsoft, HP, IBM, Motorola, Nokia, and many others are all suing each other over various patents on software techniques or user interface techniques. Worse there are now companies whose sole purpose is to own patents and to agree licencing deals or sue – these companies don't actually produce anything concrete at all.

Why should software developers care about these "software patents"? Patents create a state-sponsored monopoly. In order to manufacture a version of something that has a patent on it, you have to obtain a licence from the patent holder, and they will invariably exact a fee. Think Dyson cyclone vacuum cleaners. The Dyson company has a patent on the hardware structure for achieving the cyclone filtering functionality. So any company wishing to manufacture a similar device has to seek permission from Dyson. Fine you say, Dyson put in a lot of hard work to create the invention, and so the organization deserves the protection to make money from it. In the case of inventions with some physical reality this way of working is now an integral part of the business model of the world.

Now consider someone having a patent on linked lists – don't laugh, there are actually a number of US patents on linked lists[1]. The implication is that any software developer that wishes to use a linked list in their code has to seek the permission of the patent holder so to do. At least in the USA – patents are granted in a given legal jurisdiction and so the issue only arises in those jurisdictions where the patents hold. Even if you develop in the UK where there is, to my knowledge, no patent on linked lists, you still have to care about this if your software might be imported into the USA. If the patent holder discovers that you have imported software using linked lists into the USA then they can take action to exact their licence fee.

Hopefully you can extrapolate from this example and see why software patents are such a destructive idea. It means only companies who are big enough to play the patents cross licencing game can develop software. This stifles innovation and creates the cartels that so many of the laws in the world are designed to stop from forming.

Recently I attended the FLOSS UK Unconference 2010 held in Birmingham (UK!) on the 16th October. Although an unconference, there were two planned presentations, one to open the day, the other to

close the day. Simon Phipps [2] gave the opening presentation, and an excellent one it was. His main theme was on the benefits of openness in the software milieu, and he put forward many anecdotes and stories to support his position. Always though, in the background, was this issue of software patents, an issue on which I have done quite a lot of blogging, not to mention this article. Towards the end of his presentation, Simon raised the issue of ACTA [3], [4].

ACTA originally was intended as an international agreement to enforce the illegality of counterfeit goods. Most people can understand the original issue. If a big name brand has a product, it is usually expensive. If such products become fashionable, or at least popular, then unbranded imitations will appear at significantly cheaper prices. Some people will buy the branded product, even at the high price, because they want the cachet of the brand label. Others will buy the cheap imitation because they want the look and don't really care about the label. This is simply business. However, some organizations will attempt to make cheap imitations of the product and try and pass them off as original branded products. This form of counterfeiting is, and probably should always be, illegal. Consumers have a right to the confidence that the brand label is only used by the brand owner. Originally ACTA was an attempt to try and get all governments of the world to agree to this sort of philosophy about brands and brand ownership.

However over time, as Simon so well explained, ACTA has been hijacked by the "intellectual property" trolls, and has been widened so as to more or less impose the USA view of patents and intellectual property on all the countries of the world. Moreover all the negotiations have been in secret, behind closed doors, driven by vested interests and not by elected representatives. Worse it seeks to create yet another international organization independent of WTO (World Trade Organization) or WIPO (World Intellectual Property Organization).

Simon's presentation clearly stimulated many in the audience to care about the issues of software patents and ACTA specifically. As I have had an interest in the danger of software patents being enforced in the UK and EU for many years, when put on the spot to propose a session for the unconference, it was only natural to propose a discussion session to follow on from Simon's presentation. This was accepted, and scheduled for immediately after the coffee break – which seemed entirely reasonable!

The group that met for the session was actually a larger than I expected, I didn't take a head count but I would guess 25 to 30 people. This for me was good news since it means there are people in the software and sys admin communities wishing to know more about the problem, wishing to debate the problem, and wishing to become activists to combat the danger. Moreover, Simon joined us, and even delayed his departure as long as he could to contribute to the session.

It is difficult to try and summarize the debate that was had, particularly from memory as I was not taking notes. However the outcome of the debate was very clear and very memorable: people who care must become vocal about the problem. Inward facing discussion groups are useful but the only constructive step is to become activists. MPs and MEPs need to be informed and educated about the dangers of software patents and ACTA. The general public has to become aware of the danger to UK and EU society of the hijacking of an ostensibly good agreement, to become a tool for "big money" to control technology.

So what to do? Get yourself informed. Write blogs about the issue. Write to your MP and MEP about the issue. To do this, make use of: <http://www.mysociety.org>, <http://www.writetothem.com> and <http://www.theyworkforyou.com>.

Consider joining the Open Rights Group (<http://www.openrightsgroup.org>), and support them in the battle to stop freedom being constrained for profit.

But one caution: make sure you are informed and using the correct arguments. There is nothing quite so self-destructive as an activist with good intentions, using bogus arguments!

Finally, thanks to UKUUG for organizing the FLOSS UK unconference, to all the organizations who supported it, to John Pinner and all his helpers who did the work, and to Simon Phipps for such a good presentation on the issues.

[1] For example,

<http://www.google.com/patents?id=Szh4AAAAEBAJ&printsec=abstract#v=onepage&q&f=false>

[2] About Simon Phipps:

<http://webmink.com/about/>

[3] Wikipedia page on ACTA:

http://en.wikipedia.org/wiki/Anti-Counterfeiting_Trade_Agreement

[4] Simon Phipps on ACTA:

<http://webmink.com/tag/acta/>

SQL Antipatterns

Bill Karwin

O'Reilly Media

ISBN: 978-1-934356-55-5

300pp.

£ 26.99

Published: July 2010

reviewed by Andy Thomas

Keen to find out if it would help me fix a particularly maddening problem I was grappling with at the time with some SQL embedded in a shell script, I volunteered to review this book in the summer of 2010. But after reading a few chapters and deciding it wasn't going to help me fix my problem, I put it back in the envelope it arrived in; back in July, I found this book rather heavy-going and with my hourly script already taking 35 minutes to execute the SQL – increasing by about 5 minutes every week as the database grew - admittedly I was not in the right frame of mind to do the book any justice.

Opening the book again a couple of months later, I still found it a little alien in places and started wondering what it was about the mindset of SQL gurus that makes them so different from other programmers – what exactly were 'metadata tribbles' and 'polymorphic associations', for example? But then I realised how limited my knowledge of SQL was so I really got stuck into the book to try and rectify that.

First off, it is not a book for beginners, for those who want to improve their basic SQL skills or even for those who have a moderate knowledge of SQL and know enough to use it in PHP or Perl web scripts. It is aimed at seasoned software developers who may have developed their own solutions to tackling a particular problem or achieving a given end result, but these solutions may not be the most efficient way of doing things and may even contain serious logical or design flaws – they are antipatterns. And it is during those water-cooler moments when you start picking the brains of your colleagues with questions starting with the phrases "How can I...?" or "Why does this happen/not happen...?" that alarm bells ring inside the heads of the seriously SQL-savvy: "Hey, that guy's got an antipattern!" This book is about helping you recognise antipatterns yourself and how to deal with them effectively.

Following the introductory chapter, the book is divided into four main parts each covering one of the four genres of antipatterns discussed: logical and physical database design respectively, plus query and application development scenarios. Within each category of antipattern there are up to eight chapters devoted to a specific topic and each is laid out identically with five subsections starting with an objective (that is, what the programmer wants to achieve with the SQL code), a description of the anti-pattern, how to recognise it, and then a discussion of the legitimate uses of that antipattern (apparently, they do exist!) rounding off with a solution that will eliminate it. The fifth and final part is headed as Appendixes but contains quite a large chapter on SQL normalisation followed by a brief bibliography.

This organised and consistent structure makes it easy to use the book as a cookbook-style reference but herein lies the book's major weakness – the author tries to make the book a straight read at the

same time and this gives the book a rather clunky feel as if it can't make up its mind whether to be a reference or a textbook. Or both. I realise now that it was this that put me off reviewing the book initially.

Snippets of SQL code from an example database supporting a hypothetical software bug-tracking application are used liberally throughout the book to illustrate each antipattern – this is one of the book's good features with full SQL table definitions included so that the reader can re-create the author's database and try out the solutions for her/himself. Another good point is the book is SQL vendor-neutral and covers all flavours of SQL while at the same time comparing and contrasting the differences in the way some of them treat certain "standard" SQL concepts such as the `NULL` value; I learnt a lot from these comparisons and was rather surprised to find SQL is even more diverse and fragmented than I was already aware. One minor point: although the book itself is vendor agnostic, the example database does show its MySQL heritage and the author has the grace to admit this in the introduction; a reader wanting to create the example database for their own use will need to replace the MySQL-specific statements with the equivalent for their own database product.

In conclusion, this would be a useful book for the experienced SQL database programmer's bookshelf as it explores the common and not-so common traps even gurus can fall into. But it does not teach SQL nor is it that easy to read so I can't see it finding a place on many bedside tables. But it did do one thing for me – it hammered home yet again the fact that SQL is a lousy programming environment and the things I was trying to do with SQL in my broken script were just plain daft; after rewriting the whole thing mostly in bash, calling on our trusty old friends `sed`, `awk`, `cut` and `grep` to do the donkey work and leaving SQL to do what it was best at, script execution time fell to 30 seconds.

HTML 5: Up and Running

Mark Pilgrim

O'Reilly Media

ISBN: 978-0-596-80602-6

224pp.

£ 22.99

Published: September 2010

reviewed by Gavin Inglis

If HTML had a life story, it would be entering its thirties. Conceived in a hurry, it grew quickly, took a few wrong turns, and maybe tried to be something it wasn't for a while. With HTML 5, it is now settling into a leisurely new phase, drawing the benefits of experience.

Mark Pilgrim's book may already be familiar to you, drawn as it is from the website "Dive into HTML5" (<http://diveintohtml5.org/>) and released under a Creative Commons Attribution licence. The differences are minor and in fact the website may have a little more content. It certainly includes extra links and the words "No-Bullshit" in the Appendix title. Both versions are a technical but readable introduction to HTML 5 and its opportunities.

Code monkeys can largely skip chapter one, an enlightening discussion of how the HTML standard evolved. They will, however, want to pick out the reference to serving XHTML as the MIME type `text/html`, and the corresponding implications.

HTML 5 is unlike previous versions as it does not require a full-scale switchover; instead, it is a collection of features which can be used independently where the browser supports them. Chapter two concerns using Javascript to test for these features: the new canvas functionality, video support, local storage, offline web applications and geolocation. It might seem strange to jump straight to "how to" instructions without introducing the new features properly; this reflects the book's general "dive in and get started" approach. Once the concepts here are understood, the reader can refer to the Appendix, the endearing "All-in-One Almost-Alphabetical Guide to Detecting Everything."

Web creators – and who else would read this book? – will flirt with despair at the discussion of “quirks” and the related modes that browsers apply to deal with the toxic HTML out there, especially the “almost standards” mode. Several chapters feature the depressing mantra of “Internet Explorer does not support. . .”. To the author’s credit, at each point he considers fallbacks and alternative ways to implement the same functionality.

“What Does It All Mean?” is a patient deconstruction of a familiar HTML page, exploring each element in turn, considering its function and how it can be optimised for HTML 5. It moves on to new elements and how browsers handle them. These twenty-four pages are excellent; a valuable refresher for seasoned web coders and an accessible introduction for the assisted authoring types who have never understood the superstructure. From there the discussion shifts to new markup such as `<article>`, `<header>` and `<hgroup>`, considering how they will change the web author’s way of thinking.

The closing chapters consider the major areas of new functionality. The pages on `<canvas>` open up the potential of this drawing surface, with shapes, text, colour, gradients and images. The examples may take you back to the fun of your first hand-coded HTML pages. They conclude with an impressively compact implementation of the ancient board game Halma.

The video section is actually a pretty good general introduction to the brain-melting world of video formats and what works on the web. While not neglecting commercial choices, it does tend to lean on the advantages of an open source approach. The tutorial on encoding video using Firefogg, and later HandBrake and ffmpeg, is welcome, but there are probably a couple of pages of website pictures that could be painlessly dropped from the book. Geolocation, local storage, forms and offline usage are shorter chapters with the same approach – enough detail to get started, not enough to get bogged down.

This is not the book to hand over when your boss asks for an executive briefing on HTML 5. But it is a good choice for the web author who wants to get started immediately – and later, to pull them out of a tough spot.

Using SQLite

Jay A Kreibich

O’Reilly Media

ISBN: 978-0-596-52118-9

528pp.

£ 38.50

Published: September 2010

reviewed by Kimball Johnson

This book is described, by the author, as being “primarily written for experienced software developers that have never had a particular need to learn about relational databases. . . [but now need to]”. I do not quite fit this audience, as I have a fair amount of experience with relational databases, however, I have never used SQLite. I have attempted to learn how to use SQLite from a C application in the past, but found it hard to get started, and usually ended up using a different approach.

The book is well structured. It starts with an overview of SQLite and case studies, explaining where it is, or is not, an appropriate solution. It then continues with details on how to build the library and console program. Following this introductory section, chapters four to six provide an excellent tutorial on SQL and database design. For someone with little or no previous experience it provides an invaluable resource for doing this the ‘right way’ and for those with greater experience it is a great refresher.

From chapter seven the book gets into the details of the SQLite library. It begins with a sequence of examples: using the C API to create and use databases; using prepared statements; bound parameters;

and locking issues. This chapter alone was very useful to me, providing the booster step to get going in using it within a C application.

Chapters eight and nine cover the more advanced features, including an overview of the date/time handling and the full text search functions. These are followed by some details on creating extensions to SQLite, that can either be used by your database, or loaded by other users. The book doesn't go into great detail in a lot of the advanced features, instead referring you to the SQLite documentation, this is probably a good approach, as many are quite specialist, and require domain knowledge.

Finally, chapter ten covers creating virtual tables, which allow you to access any arbitrary data through SQLite. This is handled with some depth, including detailed examples, as this is a complex subject, and extremely powerful, this is most useful. Two examples are given, one basic to explain the concepts which produces and SQL interface to some internal data. Secondly a more complex example is given to access Apache log files though SQL. This second example will provide a solid basis for development.

The appendices provide a full reference for the build options, C API and the SQL commands, expressions and functions. Although the information in the appendices is available elsewhere they serve to complete this book as a useful tool for developers rather than merely a tutorial.

Overall this is an extremely useful book, well written and concise but not lacking in the required detail. The author wrote it after becoming frustrated by the lack of tutorial material for the API while he was learning SQLite. I feel he has achieved his aim of producing a useful tutorial and reference work, and I highly recommend it, having learnt much myself. Anyone who needs to learn SQLite, and is not comfortable with API documentation alone, should find this book invaluable.

Being Geek: The Software Developer's Career Handbook

Micheal Lopp

O'Reilly Media

ISBN: 978-0-596-15540-7

336pp.

£ 18.99

Published: August 2010

reviewed by Mike Fowler

Being Geek takes the reader through a modern career as a software developer, from getting a new job to deciding it's time to change jobs again. Each of the forty chapters of this book offer advice for career scenarios delivered in the form of amusing anecdotes. Within these anecdotes is where the power of the book lays. Most of these scenarios you've seen in your own career but if you're like me, you won't have considered how they effect your career.

For example, have you ever devoted any time into thinking about what people's reactions to bad news tells you about them? Lopp has taken the time to categorise the types of reaction you might see along a flight-fight spectrum. At the extremes you have the "I Quit" and the "Raging Bull" with an amusing selection in between. It's this light-hearted but critical approach to the scenarios that quickly get you analysing your own career.

For me the most amusing chapter of the book wasn't even for the geek, it's for their significant other. Lopp asks the reader to hand the book over and what follows is a guide to help them understand their nerd. It helps them to understand the geek's obsession with puzzles, their insatiable appetite for information and the need for a "project". It's a light hearted look into the world of a geek and according to my wife, it's also very accurate.

I'd recommend this book to any geek. The career advice is very different then what we've had in the past from a career advisor, and far more entertaining to boot.

Ubuntu for Non-Geeks (4th edition)

Rickford Grant and Phil Bull

No Starch Press

ISBN: 978-1-59327-257-9

496pp.

£ 27.97

Published: July 2010

reviewed by Mike Fowler

This introductory guide to Linux enters its fourth edition, updated to cover the release of Ubuntu 10.04. Bundled with a live CD, the novice is eased into the world of Linux. The authors quickly persuade you to install Ubuntu using positive and encouraging language. A number of common concerns are highlighted along the way to keep those beginner nerves at bay.

Each of the remaining chapters show you how to use Ubuntu to achieve the tasks you may already be used to performing on other operating systems. The authors introduce each new theme before leading you step-by-step through the common tasks. More complicated, or more involved, tasks are grouped together in projects.

I particularly liked the project concept. I've often found that once you've passed the introduction section of a guide you can quickly lose your way amongst all the new things you can try. When that happens the usefulness of the guide fades as you stumble around searching for answers. With a project you can decide to spend some time learning something, with the added bonus of achieving something at the end of it.

The book succeeds very well at being "a pain-free, get-things-done guide". Next time someone asks me where to start with Linux, I'll be pointing them at this book.

Programming the Mobile Web

Maximiliano Firtman

O'Reilly Media

ISBN: 978-0-596-80778-8

512pp.

£ 38.50

Published: August 2010

reviewed by Lindsay Marshall

At the moment the Mobile Web is probably the new rock and roll (see other review...), with HTML5/CSS3 being the new black, so the books are starting to appear. Some of them focus on single platforms (iOS, Android etc.) and others, like this one, try to cover all bases. The problem is of course, as you will find when you read a few chapters of this book, is that everything is moving so fast that the information in them becomes obsolete and/or irrelevant far too quickly - no jQuery mobile, not much Windows Phone 7. This is not the fault of the author, who has done a great job putting together a summary of devices and platforms, but it is unavoidable.

That being said, I like this book a lot and it will provide a lot of useful material for some courses I am developing (though at close to 40 pounds students are not going to be buying it, sorry.) The most important thing (and the biggest weakness, see above) is the way it pulls together feature implementation summaries for devices, platforms and technologies. As yet I haven't seen such comprehensive summaries on the net, or at least not in a single place, so for the moment this makes the book valuable if you are aiming to support multiple platforms. The book is also strong on the history of the mobile web and on the relationship with desktop technologies.

Definitely a good book that, for a short time at least, would be very useful for anyone developing for the mobile web. If they can justify the price.

Javascript Cookbook

Shelley Powers

O'Reilly Media

ISBN: 978-0-596-80613-2

560pp.

£ 38.50

Published: August 2010

reviewed by Lindsay Marshall

Please see the combined review below.

JavaScript Patterns

Stoyan Stefanov

O'Reilly Media

ISBN: 978-0-596-80675-0

240pp.

£ 22.99

Published: October 2010

reviewed by Lindsay Marshall

With JavaScript having been the new rock and roll a couple of web fashion iterations ago there has been a bit of a flood of books devoted to various aspects of it recently, and, of course, as is inevitable with print things lag a bit behind what is actually happening in the world. The JavaScript Cookbook is a good case in point. Not that there is anything wrong with the content of the book per se – it is filled with useful fragments (and sometimes more than fragments), and anyone using JavaScript seriously will find something in here that will help them. The problem is that quite a large proportion of the solutions and recipes provided are things that jQuery and other packages provide out of the box and with a nice consistent, integrated, and above all multi-browser interface, plus a good support ecosystem wrapped around them. Now, I am sure that there will be times when jQuery is deemed to be too heavyweight for a particular web application and that only a tiny part of its functionality is needed, but I am also sure that these times are few and far between – personally I can't image building anything other than an incredibly basic website without starting with jQuery: it is just too useful. (You must use one of the jQuery CDNs of course!) There is also a slightly tutorial flavour to the book and I wonder whether people who are trying to implement the kinds of things described need this.

So, I have to ask, who is prepared to spend nearly 40 pounds on a book like this? Much of the information in it is freely available on the net even when the functionality is not packaged up for you already so why would you? This may be the going rate for a technical book of 500+ pages, and it may be a way of building the e-book market (though the e-book is only a small amount cheaper), and the production is excellent as always with O'Reilly, but it just doesn't seem to be good value for money.

The big problem I have with the Cookbook and the Pattern book is the layout of the code. This is just a personal thing of course and I seem to be in a minority with respect to this, but I do find the layout style they have to chosen to be quite unreadable. I won't go into details because you are probably one of those people who format their code in this nasty way too, but I just wanted to get it out in to the open.

It isn't the only problem I have with the Pattern book (22 pounds for 200+ pages) though. Since we are getting things out in the open, let me confess that I am not a fan of "patterns". They certainly exist and are even sometimes useful, but they are also a bit of a religion and people who buy into it see patterns everywhere, and, much worse, try and force solutions to fit patterns even when they are not a good fit. Much better to be aware of the general idea and use it flexibly and appropriately rather than carting round some of the extra baggage that you get with rigid patterns. (I can hear people disagreeing with me already). Once again though, the content of the book is pretty good and useful if you want to do this kind of thing and there is good analysis of difficult circumstances where things can be a little tricky in JavaScript (setting `this` inside callbacks for example). If I ignore the pattern-talk in this, I rather like it and I've learned a lot about some of the dark areas of JavaScript's nasty semantics from reviewing it. If this is your thing and you don't know it all already, you could do worse than reading or indeed buying this book.

LPI Certification in a Nutshell

Adam Haeder et al

O'Reilly Media

ISBN: 978-0-596-80487-9

528pp.

£ 38.50

Published: July 2010

reviewed by Roger Whittaker

The Linux Professional Institute (LPI) was founded in 1999 as a non-profit community-based organisation offering a certification programme for Linux professionals.

There are three levels of certification: levels 1 and 2 have been part of the programme throughout most of its existence, while level 3 has been added more recently.

The LPI has had reasonable success in establishing itself as a respectable distribution-neutral certification for Linux, but is up against the fact that the two main commercial 'Enterprise' distributions offer their own (more expensive but practically-based) vendor-specific certifications.

Inevitably the LPI experiences problems trying to offer a distribution neutral certification: it must cover two packaging systems, and it must keep up to date with changes across at least Debian, Ubuntu, Red Hat and SUSE Linux. Perhaps soon it will need to cover two different alternatives to the traditional System V init system.

Despite challenges of this kind, the LPI copes well and, I think, deserves the support of the community. The large pool of multiple choice questions are carefully selected and tested by psychometric methods to ensure as fair a test as possible, and my experience both as a candidate and from looking at the sample questions in this book is that they are a fair test of knowledge at the level for which they are designed.

This book (now in its 4th edition) covers the material required for LPI level 1, which consists of two examinations: 101 and 102. The book is laid out in a very clear style, typical of O'Reilly's "Nutshell" series. It is effectively a Linux command index, but organised according to the objectives and criteria of the LPI syllabus, but with additional exercises and sample questions as well as advice on the exam. But this is not one of those "how to pass the exam" texts: it is full of genuine serious material and if you are properly familiar with the content, you *will* pass the exam.

MongoDB: The Definitive Guide

Michael Dirolf and Kristina Chodorow

O'Reilly Media

ISBN: 978-1-4493-8156-1

216pp.

£ 30.99

Published: October 2010

reviewed by Mike Smith

There's a lot of interest in non-relational "nosql" databases at the moment. I work in enterprise environments where SQL has been (and still is) the answer to most application data storage requirements. The enterprise SQL options are usually phenomenally expensive, not only in licensing, but the consequential maintenance and the big iron one tends to put in place to support a scale-up architecture in these environments. A lot of this is changing with adoption of MySQL, PostgreSQL and with the move to scale-out architectures with the likes of Oracle RAC (which enables a move to more cost-effective x86 servers but does double the licensing cost!) But with massive scalability requirements for worldwide systems we're now taking more radical steps. "nosql" databases are becoming the preferred option.

I'm sure you'll know that the move to Digg v4 was a bit of a trauma. Underpinned by Cassandra, which has reportedly has a few issues itself, but also demonstrates the difficulties in transitioning applications to a nosql architecture. So this background and recent events have more than peaked my interest in MongoDB.

Quite a short book, this one, with 180 pages of content over 11 chapters.

The book introduces us to nosql and MongoDB from the beginning, which will be a relief to the uninitiated. An appendix goes through the very simple steps to install the server – download, unpack and create a directory for the databases to sit in. No complex configuration is necessary here.

What is clear from the book though is that the whole nosql approach is very different (to those who've grown up with SQL) – documents of key-pairs rather than tables; object-oriented; no schema; nested documents even. An application accessing MongoDB (and any other nosql db, I would guess) potentially has to have a very different approach. This may be not too bad if there are standard ACID transactions. Maybe there's scope for another book looking purely at transitions of this type.

One thing I found interesting is that the MongoDB client is a full JavaScript Shell. You can type any JavaScript code in there and it is evaluated. There are non-JavaScript extensions to access the server and select which database to use, for instance, and these elements will be familiar to SQL client shell (such as sqldb) users.

As you'd expect there are chapters on making queries and indexing. Then some more advanced topics including MapReduce, GridFS (a way of storing large files in the database) and server-side scripting (again using JavaScript).

There's a brief chapter on administration – operational activities such as starting up, cleanly shutting down, backup and restore, database recovery after unclean shutdown etc. All good.

Replication is covered, with multiple slaves and various failover scenarios. More advanced configuration, and in particular for high read-intensive scalability, is achieved with replica sets. There's also a discussion on the requirements of the oplog (sizing etc) to ensure slaves don't go out of sync and replication continues. A good chapter.

I do have a slight concern: whilst built for performance there seems to be an acceptance that data may be lost in failure scenarios. Perhaps it's incumbent on the application to sort this type of thing out, but in Larry's world the database guarantees recovery to the last complete transactions, and rolls everything else back tidily. Even crash-consist backup techniques are tolerated with storage arrays that guarantee write-order fidelity.

Sharding. I just like this word. Sounds very cool to me. It's like partitioning in Oracle. In fact sharding is sometimes called horizontal partitioning – splitting up the data based on value ranges or something similar to enable access/processing of these subsets of the data. With horizontal partitioning the data can be split and placed in different tables or even on different nodes enabling write-intensive application scalability. The beauty of MondoDB is that sharding is automated, and the book has a chapter on how to set it up.

The book is rounded off with a few example applications using Java, PHP, Ruby and Python. In each case an additional driver is required to provide the MongoDB API. There are some good tips here of where to look for more documentation and tools, but each example was very light on information and code.

When I got to the end of the book, it just sort of, well, ended. I was left wanting more, and would have liked to have seen a concluding chapter, summarising everything learnt, pointers for more info, perhaps a link to where examples can be downloaded (I'm not sure if there is such a place). Also perhaps some examples of real world production implementations - I'll do that now: Justin.tv, New York Times, bit.ly, SugarCRM, FourSquare ... (and many more)

So definitely recommended for the beginner who wants to get a flavour of what MongoDB is all about and quickly get up to speed with installing, configuring and operating the database. Getting an application up and running may take a little longer.

Cooking for Geeks

Jeff Potter

O'Reilly Media

ISBN: 978-0-596-80588-3

432pp.

£ 26.99

Published: August 2010

reviewed by Ray Miller

My shelves are groaning under the weight of cookery books, but there aren't many I've read from cover to cover. "Cooking for Geeks" is one of those gems that is both an excellent reference manual and a good read. While this book does contain some recipes (100 or so), it's not the one to buy if you're looking for a recipe book. Recipes give you quantities of ingredients and step by step instructions for transforming those ingredients into great meals. This book goes some way to explaining the whys and wherefores of each of those steps. Once you have an understanding of the processes involved in cooking and preparing food – and the science behind them – you can start adapting recipes and inventing new ones. You'll know when an ingredient can be substituted, and what with. You'll know when a step in a recipe can be skipped. You'll start to spot – and correct for – mistakes in published recipes.

The book encourages us to treat our kitchens as our own personal chemistry lab, and it builds the confidence we need to start experimenting. Most important of all, it encourages us to have fun with our cooking. A geeky humour runs throughout the book, with the title of the first chapter, "Hello, kitchen", setting the tone. Of course there's some serious stuff in here too, including an essential section on food safety and foodborne illness.

I'm surprised at just how much information has been packed into the 400 pages of this book. Want to know how to pasteurize an egg? The temperature collagen starts to break down when you cook meat? Where to find enzymes that will do the same job? It's all in here. (The full table of contents is at: <http://oreilly.com/catalog/9780596805890/>.)

There are also plenty of helpful tips along the way. I now cook pancakes without any fat in the pan. You need a good non-stick pan to do this, but they cook more evenly this way. And on the gadget

front, I'm looking out for a compressed gas cream whipper so I can try foamed scrambled eggs and instant chocolate mousse.

As expected, the chemicals used to make the foams, spheres, and heat-stable gels of molecular gastronomy make an appearance, and the book gives a good introduction to sous vide cooking. This is where food is vacuum-sealed in a plastic bag (the "sous vide" part) and cooked in a temperature-controlled water bath, a method of cooking used extensively in top restaurants where consistency is key. Sous vide is gaining popularity in domestic kitchens, with the first water bath aimed at the domestic consumer arriving on the European market just a couple of months ago. My friends thought I was crazy when I spent almost a month's salary on a water bath and vacuum packing machine earlier this year, but once you've tried fillet steak cooked sous vide (a perfect medium rare throughout) and confit pork cheeks (cooked gently in goose fat for 36 hours), there's no turning back.

If you're really interested in the science of cooking, you'll want Harold McGee's "On Food and Cooking" on your shelves and perhaps "The Science of Cooking" by Peter Barham. But you can't have too many cookery books, and "Cooking for Geeks" is a very readable introduction to the subject with plenty of light-hearted diversions to keep it from getting too dry. I particularly enjoyed the interviews and guest appearances by some of my favourite food bloggers – it was like having old friends around.

The author sums up my feelings brilliantly in the afterword:

Curiosity and the joy of discovering how something works are two of a geek's defining characteristics. I can think of very few other things that have brought me as much joy as learning to cook and providing for others. It scratches the same neurons that solving a puzzle or producing a brilliant piece of code does, but tastes better and often takes less time – not to mention that you can do it for other people and make them happy too!

Some of my geek friends might find a copy in their seasonal festive stockings this year. Now, back to the kitchen.

Contributors

Mike Fowler is a Senior Software Engineer at Dotted Eyes. He is a firm open-source advocate and has helped companies move from Microsoft based solutions to complete Linux/PostgreSQL stacks. He is a major contributor to YAWL, an open source BPM/Workflow system. When not melled with a keyboard, he spends his time playing as many different stringed instruments as he can.

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.

Kimball Johnson is a Systems Developer Administrator for LUNS Ltd in Lancaster. He has been programming since a very early age, starting with BBC Micros, then MS DOS and Windows Systems, however was enlightened with a copy of Debian GNU/Linux Woody at university. He is always wanting to learn and is particularly interested in automating as much of his job as possible.

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.

Ray Miller worked as a systems administrator at the University of Oxford for almost a decade, during which time he was actively involved with UKUUG. He served on Council for two years and organised three successful LISA conferences. After a brief interlude in a professional kitchen, he returned to IT and now works as a software developer supporting a mouse genetics project at the

Wellcome Trust Sanger Institute. He is a keen amateur cook and, when he's not chasing down errant DNA sequences, can usually be found in the kitchen. Ray blogs about food and cooking at <http://startagainatzero.blogspot.com/>.

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, Imperial College London and a number of small companies. 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 a director of a wholesale insurance broker. Outside of work he can usually be found filing documentation bugs against various open source and free software projects.

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.

Russel Winder was originally a theoretical particle physicist but decided in 1980 that being a UNIX systems programmer was more fun. However, academia continued to call and he moved to UCL to lecture in programming, software engineering and human-computer interaction, and do research on parallel programming languages and psychological and socio-technical aspects of software development. After 13 years at UCL, Russel moved to KCL to be Professor of Computing Science. Having revamped the teaching programme, continued the research on parallel programming languages, and started research programmes in health informatics, it was time for new challenges so he left KCL to become CTO of OneEighty Software Ltd, a company using novel virtual machine approaches to embedded systems. Unfortunately, the money ran out for this start-up before the orders flooded in and so it had to fold. For the last few years, Russel has been a consultant, analyst, trainer (Python, Java, Groovy, GPar, Python-CSP, Waf, SCons, Gradle, Gant, Bazaar, Mercurial, Git) and author ("Developing Java Software" third edition, and "Python for Rookies"). The rise of multicore processors means that parallelism has finally arrived (after 25 years of being the coming technology), so he has been able to return to being interested in all things concurrent and parallel. Russel is centrally involved with the development of the Groovy programming language, and is author of the Gant build framework. He also works on GPar, Python-CSP, SCons, and Waf, indeed on anything that people will pay money for.

Contacts

Paul Waring
UKUUG Chairman
Manchester

Phil Hands
Council member
London

Kimball Johnson
Council member
Lancaster

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