LINUX JOURNAL™
Since 1994: The Original Magazine of the Linux Community

HOW TO ADD SECURITY TO YOUR SSH KEYS

DECEMBER 2013 | ISSUE 236 | www.linuxjournal.com

BEST of LINUX

2013 READERS' CHOICE AWARDS

A LOOK AT USABILITY
IN OPEN-SOURCE SOFTWARE

INTERVIEW WITH AUSTRALIS
ARTIST, MUSICIAN AND LINUX USER

\+

CONSIDERING WOMEN AND LINUX

USING LVM
THE LOGICAL VOLUME MANAGER

WORKING WITH IMAGEMAGICK
ON THE COMMAND LINE

usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

UPCOMING CONFERENCES

For a complete list of USENIX and USENIX co-sponsored events, see www.usenix.org/conferences

FAST '14: 12th USENIX Conference on File and Storage Technologies

February 17–20, 2014, Santa Clara, CA, USA
www.usenix.org/conference/fast14

2014 USENIX Research in Linux File and Storage Technologies Summit
In conjunction with FAST '14
February 20, 2014, Mountain View, CA, USA
Submissions due: January 17, 2014

NSDI '14: 11th USENIX Symposium on Networked Systems Design and Implementation

April 2–4, 2014, Seattle, WA, USA
www.usenix.org/conference/nsdi14

2014 USENIX Federated Conferences Week

June 17–20, 2014, Philadelphia, PA, USA

USENIX ATC '14: 2014 USENIX Annual Technical Conference
www.usenix.org/conference/atc14
Paper titles and abstracts due January 28, 2014

HotCloud '14: 6th USENIX Workshop on Hot Topics in Cloud Computing

WiAC '14: 2014 USENIX Women in Advanced Computing Summit

HotStorage '14: 6th USENIX Workshop on Hot Topics in Storage and File Systems

UCMS '14: 2014 USENIX Configuration Management Summit

ICAC '14: 11th International Conference on Autonomic Computing

USRE '14: 2014 USENIX Summit on Release Engineering

23rd USENIX Security Symposium

August 20–22, 2014, San Diego, CA, USA
www.usenix.org/conference/usenixsecurity14
Submissions due: Thursday, February 27, 2014

Workshops Co-located with USENIX Security '14

EVT/WOTE '14: 2014 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections

USENIX Journal of Election Technology and Systems (JETS)
Published in conjunction with EVT/WOTE
www.usenix.org/jets
Submissions for Volume 2, Issue 2, due: December 5, 2013
Submissions for Volume 2, Issue 3, due: April 8, 2014

HotSec '14: 2014 USENIX Summit on Hot Topics in Security

FOCI '14: 4th USENIX Workshop on Free and Open Communications on the Internet

HealthTech '14: 2014 USENIX Workshop on Health Information Technologies
Safety, Security, Privacy, and Interoperability of Health Information Technologies

CSET '14: 7th Workshop on Cyber Security Experimentation and Test

WOOT '14: 8th USENIX Workshop on Offensive Technologies

OSDI '14: 11th USENIX Symposium on Operating Systems Design and Implementation

October 6–8, 2014, Broomfield, CO, USA
www.usenix.org/conference/osdi14
Abstract registration due April 24, 2014

Co-located with OSDI '14:

Diversity '14: 2014 Workshop on Diversity in Systems Research

LISA '14: 28th Large Installation System Administration Conference

November 9–14, 2014, Seattle, WA, USA
https://www.usenix.org/conference/lisa14
Submissions due: April 14, 2014

Do you know about the USENIX Open Access Policy?

USENIX is the first computing association to offer free and open access to all of our conferences proceedings and videos. We stand by our mission to foster excellence and innovation while supporting research with a practical bias. Your membership fees play a major role in making this endeavor successful.

Please help us support open access.
Renew your USENIX membership and ask your colleagues to join or renew today!

www.usenix.org/membership

Stay Connected...

twitter.com/usenix
www.usenix.org/youtube
www.usenix.org/gplus
www.usenix.org/facebook
www.usenix.org/linkedin
www.usenix.org/blog

SILICON MECHANICS
SERVERS • STORAGE • HPC

nexenta

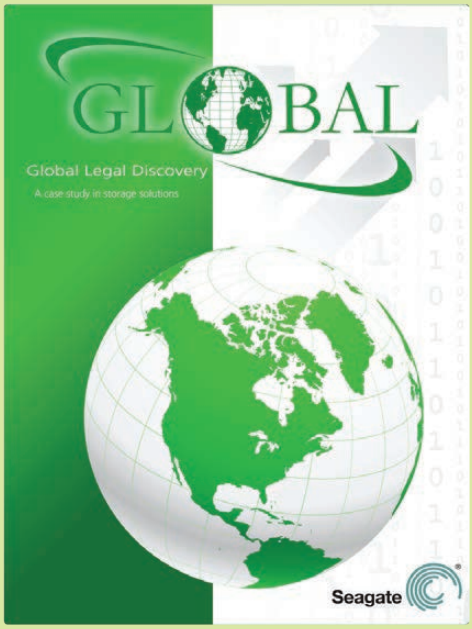
Are you considering software-defined storage?

zStax
ZFS Unified Storage

zStax StorCore ZFS Unified Storage from Silicon Mechanics is truly software defined storage.

From modest data storage needs to a multi-tiered production storage environment, the **zStax StorCore** ZFS unified storage appliances have the right mix of performance, capacity, and reliability to fit your needs.

December Case Study Feature



Unified Storage is Crucial Part of
Search and Discovery for the Cloud
www.siliconmechanics.com/casestudies

zStax StorCore 64

zStax StorCore 104

Talk with an expert today: 866-352-1173
www.siliconmechanics.com/zstax

CONTENTS

DECEMBER 2013
ISSUE 236

2013 READERS' CHOICE AWARDS

FEATURE

56 **Readers' Choice Awards 2013**
The best of Linux.
Shawn Powers

INDEPTH

84 **It's about the User: Applying Usability in Open-Source Software**
How to apply simple usability testing to make your software project easier for new users to learn and use.
Jim Hall

98 **Interview with Australis**
Artist, musician, software developer and Linux user to boot.
Fred Mora

Cover Image © Can Stock Photo Inc. / Andreus

COLUMNS

30 Reuven M. Lerner's At the Forge
2013 Book Roundup

36 Dave Taylor's Work the Shell
Resizing Images with ImageMagick

40 Kyle Rankins' Hack and /
Secret Agent Man

44 Shawn Powers' The Open-Source Classroom
LVM, Demystified

110 Doc Searls' EOF
Mars Needs Women

KNOWLEDGE HUB

82 Webcasts and White Papers

IN EVERY ISSUE

8 Current_Issue.tar.gz

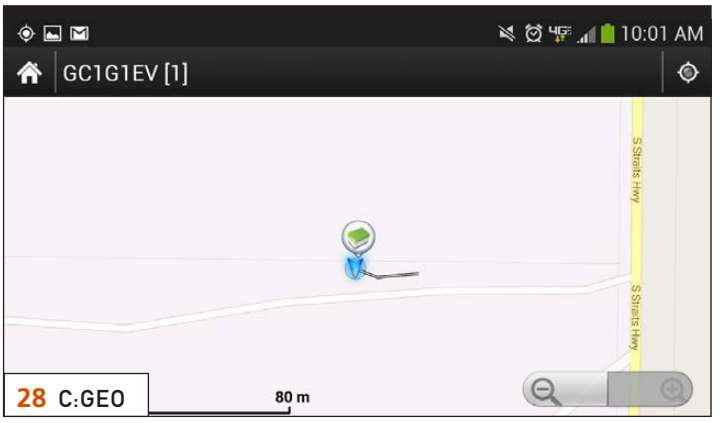
10 Letters

18 UPFRONT

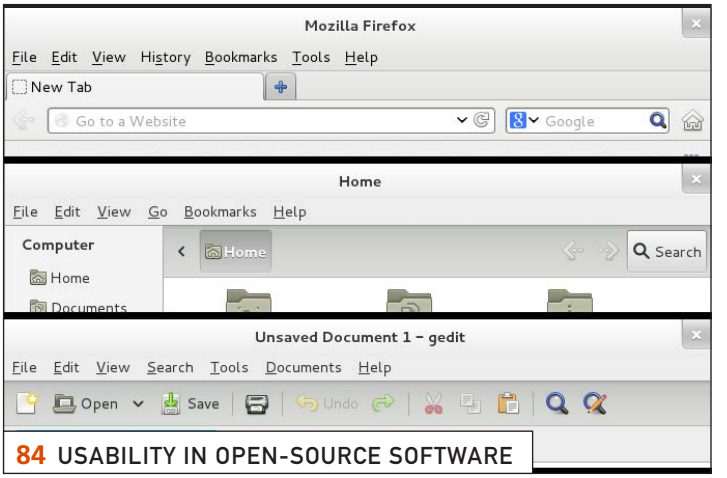
28 Editors' Choice

52 New Products

115 Advertisers Index



28 C:GEO



84 USABILITY IN OPEN-SOURCE SOFTWARE

98 AUSTRALIS

ON THE COVER

- How to Add Security to Your SSH Keys, p. 40
- Best of Linux: 2013 Readers' Choice Awards, p. 56
- A Look at Usability in Open-Source Software, p. 84
- Interview with Australis—Artist, Musician and Linux User, p. 98
- Using LVM the Logical Volume Manager, p. 44
- Working iwth ImageMagick on the Command Line, p. 36
- Considering Women and Linux, p. 110

LINUX JOURNAL (ISSN 1075-3583) is published monthly by Belltown Media, Inc., 2121 Sage Road, Ste. 395, Houston, TX 77056 USA. Subscription rate is $29.50/year. Subscriptions start with the next issue.

LINUX JOURNAL

Executive Editor	Jill Franklin jill@linuxjournal.com
Senior Editor	Doc Searls doc@linuxjournal.com
Associate Editor	Shawn Powers shawn@linuxjournal.com
Art Director	Garrick Antikajian garrick@linuxjournal.com
Products Editor	James Gray newproducts@linuxjournal.com
Editor Emeritus	Don Marti dmarti@linuxjournal.com
Technical Editor	Michael Baxter mab@cruzio.com
Senior Columnist	Reuven Lerner reuven@lerner.co.il
Security Editor	Mick Bauer mick@visi.com
Hack Editor	Kyle Rankin lj@greenfly.net
Virtual Editor	Bill Childers bill.childers@linuxjournal.com

Contributing Editors

Ibrahim Haddad • Robert Love • Zack Brown • Dave Phillips • Marco Fioretti • Ludovic Marcotte
Paul Barry • Paul McKenney • Dave Taylor • Dirk Elmendorf • Justin Ryan • Adam Monsen

Publisher	Carlie Fairchild publisher@linuxjournal.com
Director of Sales	John Grogan john@linuxjournal.com
Associate Publisher	Mark Irgang mark@linuxjournal.com
Webmistress	Katherine Druckman webmistress@linuxjournal.com
Accountant	Candy Beauchamp acct@linuxjournal.com

Linux Journal **is published by, and is a registered trade name of,**
Belltown Media, Inc.
PO Box 980985, Houston, TX 77098 USA

Advertising
E-MAIL: ads@linuxjournal.com
URL: www.linuxjournal.com/advertising
PHONE: +1 713-344-1956 ext. 2

Subscriptions
E-MAIL: subs@linuxjournal.com
URL: www.linuxjournal.com/subscribe
MAIL: PO Box 980985, Houston, TX 77098 USA

SHAWN POWERS

Hear Ye, Hear Ye

As we wave a fond farewell to 2013, we close out the year with one of our favorite issues. I know, I often tease about being lazy and having the readers write the Readers' Choice issue, but I only do that because it's absolutely true! Seriously though, this is one of our favorite issues because we get to hear from you. Some of your feedback was expected, some was a little surprising, but it was all appreciated. If you want to see how you line up with your fellow readers, you can skip ahead to the Readers' Choice article, but if you do, you'll be missing out on tons of great content!

Reuven M. Lerner, for example, gives us his annual summary of the most interesting books he's read. Whether you're looking for a nice holiday read or want to make sure you haven't missed any gems this past year, Reuven shares his insight from his library. Dave Taylor follows up with another column on the library of tools available in the ImageMagick suite. If you need to edit photos, especially in an automated or bulk way, the command-line image editing tools Dave describes are invaluable.

SSH is a tool just about every geek knows well. There are countless tutorials for creating SSH key pairs for adding security (and convenience) to server logins, but that same layer of security comes with the danger of stolen key files. Kyle addresses the issue of SSH security while keeping as much convenience as possible. SSH Agent may be the best of both worlds: convenience and security. I follow Kyle's article with a how-to for implementing LVM. Logical Volume Manager adds convenience and expandability to your system with far less complexity than it might seem. If you've ever been scared of LVM, be sure to check out my column.

If you're tired of hearing how hard the GIMP is to use because of its interface, Jim's article will interest you.

Jim Hall helps point out the warts in open-source software this month with his article on usability testing. If you're tired of hearing how hard the GIMP is to use because of its interface, Jim's article will interest you. As longtime geeks, it's often easy to overlook an interface's shortcomings. This article helps us take off our rose-colored glasses and see our programs for what they are.

We also had the opportunity to interview a Linux user, musician and developer: Australis. I always love to meet folks who use Linux on a daily basis for their livelihoods, and Fred Mora had the fortune of interviewing the indie artist. If you like to hear how Linux and open source can influence and empower artists, you'll really enjoy the interview. I know we sure did.

I don't normally mention Doc Searls' closing column in my Current_Issue column, but this month, he brings up an interesting discussion about the tech world, and *Linux Journal* specifically. In a male-dominated industry, *Linux Journal* is owned and managed by women. What does that mean in the bigger picture? Doc starts a great discussion on an important topic.

Of course, the big story this month is the Readers' Choice article, or as we're tempted to call it, "The Raspberry Pi Award Ceremony!" I probably should have said "spoiler alert", but I'm sure it's no surprise that the Raspberry Pi is still very popular among readers. Thankfully, there's a boatload of other categories, all of which were chosen by you. This year, not only did readers cast the votes, but there was an entire nomination round as well. Thank you to everyone who participated. You made this issue an awesome one.■

Shawn Powers is the Associate Editor for *Linux Journal*. He's also the Gadget Guy for LinuxJournal.com, and he has an interesting collection of vintage Garfield coffee mugs. Don't let his silly hairdo fool you, he's a pretty ordinary guy and can be reached via e-mail at shawn@linuxjournal.com. Or, swing by the #linuxjournal IRC channel on Freenode.net.

letters

Say Goodbye to Windows XP—StartUbuntu Project

I'm amjjawad from Ubuntu Community, and I am the founder and leader of the StartUbuntu Project (**https://wiki.ubuntu.com/StartUbuntu**). You can find all the information about the project from that link.

I'm approaching you to help me so we can spread the word of Linux worldwide. We don't have much time left. We are trying to reach as many users as possible. I am in charge of many teams and projects, but I am trying to keep a low profile in all these other areas in order to focus on StartUbuntu, so we can reach as many Windows XP users as possible. With your help and support, we surely can achieve that in no time.

I appreciate your time reading this, and I am looking forward to hearing what you think about it.
—amjjawad

I didn't realize Windows XP was still being supported. You probably are correct that computers running XP will not be able to run Windows 7 or Windows 8 very well, but they'll still be able to run Linux. I hope the expiring support opens doors that might otherwise have been closed. Good luck!—Shawn Powers

Supercomputers Run Linux

This is not your everyday Linux computer, but it is currently the world's fastest supercomputer. It's the Tianhe-2 (**http://en.wikipedia.org/wiki/Tianhe-2**), and it runs Kylin Linux, which is a version of Ubuntu.

But, that's not all! The top ten ranking supercomputers all run Linux (**http://en.wikipedia.org/wiki/TOP500**).

There's even more. From the top 500 fastest computers on Earth, 476 run

Linux. That's more than 95%.

Keep up with the good work at *LJ*. I'm glad to be part of the community.
—jschiavon

We're glad to have you. It does make me wonder about that other 5% though, you know? Most of those are running UNIX of some sort, and three of them are running Windows. I wonder if those are hosted by Microsoft itself—Hotmail servers?
—Shawn Powers

Integrating Linux into a Windows Network

I am a system administrator for a large secondary school in England. I've always been fine with Windows, and I have used Linux for desktop environment use or for just researching different technologies. However, as part of an expansion from our ICT office, I am having to manage and help other local primary schools in the near vicinity. We look after a total of five primary schools now, and I am encountering a problem. Most of these primary schools have very limited ICT budgets, and they have a range of equipment with some of it dating to ten years ago. Suffice it to say that money for licenses is also almost nonexistent, which brings me to the point of Linux. How can I use a flavour of Linux, say CentOS or Ubuntu, to sit along with the other Windows servers and offer extra services to enhance functionality? I was thinking to use them as a Backup to HDD solution, Clonezilla to dish out Windows or Linux images, and maybe even using Samba to blend it

with AD and host user accounts or shares. I know some of the basics, but I just don't know how to bring everything together. The boxes I'm thinking of using were running on Windows 2003.

Any help would be appreciated.
—Rene Duranona

I'll give you two really quick suggestions. If you're mixing platforms, definitely use Windows AD for user authentication. Linux is much better at playing with AD than Windows is at playing with, well, anything other than Windows. Second, include your teachers in the discussion. They will frustrate you, and they won't always understand you, but if they are a part of the planning, their ownership of the project will make your life infinitely easier. And a third bonus suggestion: find other schools that already have implemented Linux, and learn from their successes and failures. Good luck. I've been where you are, and it's both awesome and terrifying!—Shawn Powers

Article of Interest

I've lived in Grand Rapids, Michigan, for 34 years, and I am trying to visualize the 15-level parking deck in downtown Grand Rapids mentioned in "Dude, Where's My Car?" (in the UpFront section of the October 2013 issue). I have no real knowledge of Freenode, so I've written my thought here. Thanks in advance.
—Tad Gilliam

Hmm...well I didn't really count the floors, so 15 might have been a high estimate. I was referring to one of the Ellis Parking complexes. I know it was really tall, and it took a million years to walk down. (Oops, I did it again!)—Shawn Powers

Parallella Review?

Since the boards are to be available by December 2013, will *LJ* cover the board, its Ubuntu release and possible applications any time soon? That would be great.

I'd write an article myself, but I know for sure I would not be writing any parallel code to some meaningful level.
—Joris

If we get a developer who wants to share, that's the sort of thing we love to publish. At the very least, this letter will get people thinking about it. Thanks for the heads up!—Shawn Powers

HealthCare.gov

I know nothing about the technology behind, or the reasons for the problems with, HeatlhCare.gov, the Obamacare interface that has been in the news so much lately. But, a quick look at the Web suggests that this was an open-source project, largely. I would love to see *Linux Journal* examine this technology, the problems, what went wrong, what went right and why. This even seems like a reporting, analysis and writing project that could be distributed across a few of your usual excellent contributors.

—Greg Laden

I'm not sure we'll be privy to the inner workings of the Web site, but I'll be interested to see what sort of information (if any) comes out of the situation. There's no doubt more planning should have gone into the project, regardless of the software involved. I know there was some of the front-end code on GitHub at one point, but it was pulled, etc., etc. I doubt there was a single issue with the launch, but rather many things that went wrong, compounded by enormous amounts of traffic. I'm just glad it wasn't my project!—Shawn Powers

Games Section Needed

My name is António Casqueiro and I've been a *Linux Journal* subscriber since 2011.

I've been a regular Linux user for about eight years now. I came from the Windows world, frustrated with having to be concerned about viruses and having to pay for operating system licenses for each PC I have (I've got three). And that was not enough. There was, and still is, the MS Office issue. Even when I was a regular Windows user, I started to use OpenOffice.org, and I was glad with it, but my coworkers insisted on sending me MS Word files. I could open them to read, but if I changed the document and saved it again, the formatting was affected. So I was being coerced into buying MS Office licenses too!

I decided to shout and say, *that's enough*, no more Windows for me. Linux has all I need! Since I'm a Java programmer and that programming language is cross-platform, I can do my job in Windows, Mac or Linux.

When I'm not programming, most of the time I'm surfing the Web. Since FireFox is multiplatform and even faster than Internet Explorer, the

transition was painless. If I'm doing other stuff like listening to music, Amarok and Clementine are perfect for that. As far as watching a movie, MPlayer made it simple because it had all the required codecs, and there was no need to search and install custom codecs like I had to do in Windows. Currently, I'm using VLC and I'm quite happy with it. And for editing images, GIMP is a must.

So do I miss or need Windows anymore? Well unfortunately, that's not quite true yet. Why, you may ask, what's missing in Linux? *The games*!

Sure there are some games, but almost all of them are available only in Windows. So something was missing in my life, since I detached myself from Windows, but that constraint changed last year when the Unity 3D engine started to support Linux as a deployment platform. This year, another great engine is also coming with Linux support, it's LeadWerks.

Okay, I've shared my Windows-to-Linux conversion story with you, but what can *Linux Journal* do besides publish it? Actually, you can do something to fill the hearts of the Linux users like myself, because there is a relevant section missing from *Linux Journal*, the games section. If every month you tell us about cool software, hardware and books, why not games?

What I'm asking is for you to consider adding a game section to the journal, telling us about new cool Linux games being developed or that have been published. Since Steam is betting big on Linux now with SteamOS in the near future, the lack of Linux games will not be an issue anymore.

Also keep an eye on Kickstarter, because many indie developers are there trying to get their cool games funded. One noteworthy amazing space simulation game is fighting for its chance to become a reality there. It's called Skyjacker. Make sure you check it out, and maybe you could write about it so that the word among Linux gamers is spread (I would really appreciate that). Just think of this game as the *Star Citizen* (*SC*) for Linux in terms of quality. Because *SC* is being made using a game engine that doesn't support Linux, Linux gamers will not be able to play it any time soon. But *Skyjacker* (**http://www.skyjackergame.com**) also has AAA graphics, the spaceships are awesome and the

gameplay will be so much fun, you can't miss that one!
—António Casqueiro

All I heard is "Readers demand Shawn plays more games!" Seriously though, when I do post game-related information (like mentioning The Humble Bundle stuff, and my constant discussion of Steam), I wonder if readers are more annoyed than informed. It's nice to hear there is some demand for gaming information. I'll do my best!—Shawn Powers

ip6tables/YouTube

I have seen that you have a "Mastering iptables" three-part video series on your YouTube channel. May I ask if you would consider making a video about IPv6 firewalling with ip6tables?

It definitely would be helpful to someone, for sure for me.
—Stefan

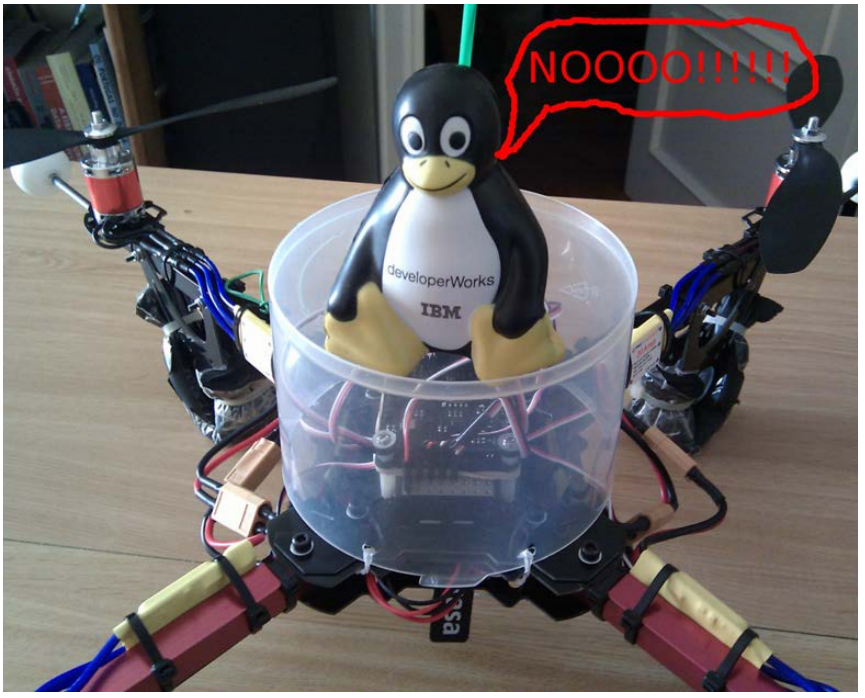
Thank you for the excellent idea. We'll continue to have IPv6 discussions here at LJ *and*

see what we can come up with.
—Shawn Powers

Photo of the Month

I have many interests besides Linux and open source, and one of them is multicopters using some excellent software like the Arduino IDE, which uses GNU tools. Here's one photo of Tux having second thoughts about flying in my Xcopter.

—Luis Sismeiro



WRITE *LJ* A LETTER We love hearing from our readers. Please send us your comments and feedback via **http://www.linuxjournal.com/contact**.

PHOTO OF THE MONTH

Remember, send your Linux-related photos to ljeditor@linuxjournal.com!

OVH.COM
Dedicated Infrastructure for your Business

OVH.com: The Widest Range of Operating Systems on the Market!

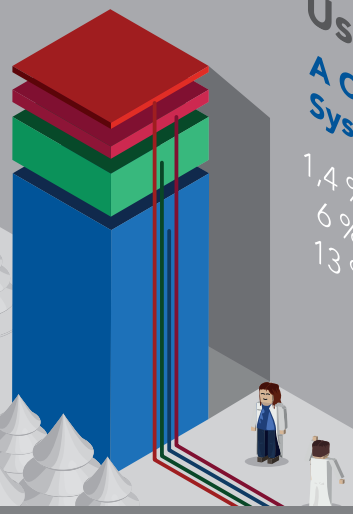
Figures based on an analysis of operating systems installed on OVH.com dedicated servers leased in North America. Study carried out in September 2013.

User Choice

A Crushing Majority of Operating Systems are Based on Linux

1,4 % FreeBSD
6 % Windows
13 % VMware
78 % Linux

- 36 % CentOS
- 23 % Ubuntu
- 18 % Debian
- 7.5 % ESXI (VMware)
- 4.5 % Proxmox



OVH Release: an expertly designed hosting operating system

Nearly 8% of the servers leased from OVH are equipped with the distribution specially developed in-house. Offered as a beta version on dedicated servers and VPS, Release #3 is based on CentOS 6 and equipped with Apache 2, PHP5, OVHm and Webmin. OVH Release can be managed via an English or a French interface, enabling you to get a hosting server up very quickly, and to configure domains and services in just a few clicks. Ideal for people that use their machine for web hosting and/or email hosting.

diff -u
WHAT'S NEW IN KERNEL DEVELOPMENT

There have been a number of attempts to make Linux able to isolate **CPUs** and other hardware resources, effectively guaranteeing access to those resources to the particular processes that need them.

Christopher Lameter recently posted his own attempt. His code operated very early in the boot cycle, so it could prevent any startup dæmons from getting onto the CPUs in question.

There immediately was a discussion of whether to redo these patches as enhancements to things like **isolcpus** and **cpusets**, which provided similar features. Everyone seemed to be in favor of the feature set, but no one seemed satisfied with the existing solutions or with Christopher's version.

According to **Mike Galbraith**, isolcpus, for example, apparently is being taken out of the kernel at some point, although like Christopher's code, it operates at a very early stage of the boot cycle. But, as **Gilad Ben-Yossef** said, cpusets, on the other hand, started later in the boot cycle and wasn't able to handle certain types of process migration, but it was more elegantly written, which counts for a lot, in Linux.

One possibility was to keep isolcpus around and enhance it with Christopher's code, but take out most of the rest of its code, leaving it just a configuration tool for cpusets. But Christopher nixed that idea, saying that isolcpus was actually broken and insane, and simply had to go.

As it turned out, even cpusets was not immune and was slated to be replaced by cgroups, a **Google** project to provide similar features not just for CPUs, but for memory and all other resources on the system.

The real difficulty with any of these solutions seems to be correctly handling all the various cases that may arise. Migrating threads may leave child threads behind that also need to be migrated. There are potential race conditions. And some groups of users, like banks and financial institutions, want these features to co-exist with

nearly bare-metal control over the system as possible. How can all this be arranged?

It remains unclear. But ideas keep bubbling up, code keeps getting written, and at some point, something is bound to strike a chord with everybody.

In case you're wondering, **Linus Torvalds** doesn't use backups. He had a hard disk crash recently and talked about it, so we got to see a bit of how he deals with such things.

Apparently, the crash did cost him a few days of work. But he remarked, "I long ago gave up on doing backups. I have actively moved to a model where I use replaceable machines instead. I've got the stuff I care about generally on a couple of different machines, and then keys etc backed up on a separate encrypted USB key."

And, **H. Peter Anvin** said he did a similar thing, because disk drives just weren't reliable enough. He said he always mirrored his main system disk onto other computers.

It's unusual for a whole hardware architecture to be taken out of the kernel tree, but it can happen. **Guenter Roeck** recently pointed out that the **H8/300 architecture** hadn't worked for years and wouldn't even compile.

Guenter posted a patch to gut the code, but he also invited discussion to make sure no one was working on resuscitating it. Through a fluke, he left the H8/300 maintainer, **Yoshinori Sato**, off the cc list, but **Joe Perches** caught that and added him back in.

Barring any objections, **Greg Kroah-Hartman** said he fully supported the patch and pointed out that they always could undo the git commit if it turned out someone wanted it back. **David S. Miller** also saw no need to keep dead code alive, as did **Wim Van Sebroeck**.

Linus Torvalds said he was fine with taking out the code. It was a big patch, but because it was a whole architecture, it was relatively isolated from the rest of the kernel and posed little threat to anything else.

It's possible there may be a delay, as **Geert Uytterhoeven** wanted to give Yoshinori a chance to discuss the issue in person at the **Kernel Summit**.

Miklos Szeredi has introduced a new system call to swap the names of two files. His **rename2()** system call made certain kinds of filesystem operations atomic, where they hadn't been before. He gave the example of replacing a directory tree with a symlink.

Another value of the patch,

Miklos said, was its ability to handle whiteouts in **union filesystems**. In a union filesystem, where multiple filesystems are overlayed, a whiteout allows the user to delete a file in the visible union, even if the file itself happens to be on a read-only filesystem. The file is "whited out" of the overlay, so the user no longer sees it.

His rename2() system call would make that type of situation much cleaner, although Miklos acknowledged that there were some cases that still would have problems.

There was some talk of extending rename2() to allow a more complex, yet more flexible behavior, but Linus Torvalds said that Miklos's patch was actually a simplified version of an earlier effort that had been too complicated for Linus' liking. Linus said, "I was actually very relieved to see this much simpler and cleaner model, because the alternative really was nasty." **—ZACK BROWN**

A Plexible Pi

If, like me, you've jumped onto the Plex bandwagon with both feet, you've probably discovered how difficult it is to make a standalone Plex player. Sure, you can install an entire OS, then auto-start the Plex program in full screen, but it's not as simple as installing the XBMC distro, or even OpenELEC. If you own a Raspberry Pi, that has all changed.

RasPlex is a custom Linux distribution based on the popular (and awesome) OpenELEC Raspberry Pi port. Rather than installing XBMC on an RPi, however, RasPlex installs the Plex Home Theater application. Granted, the Raspberry Pi does struggle with menu speed in Plex until the cache of thumbnails is built, but with a developer focusing strictly on making Plex work for the RPi, those caching issues will be solved soon!

If you have Plex on your phone, tablet, computer, browser and Roku, but really wish you could make a standalone Plex Home Theater with your Raspberry Pi, check out RasPlex today: http://www.rasplex.com.

—SHAWN POWERS

Non-Linux FOSS: Let's Make Music Together

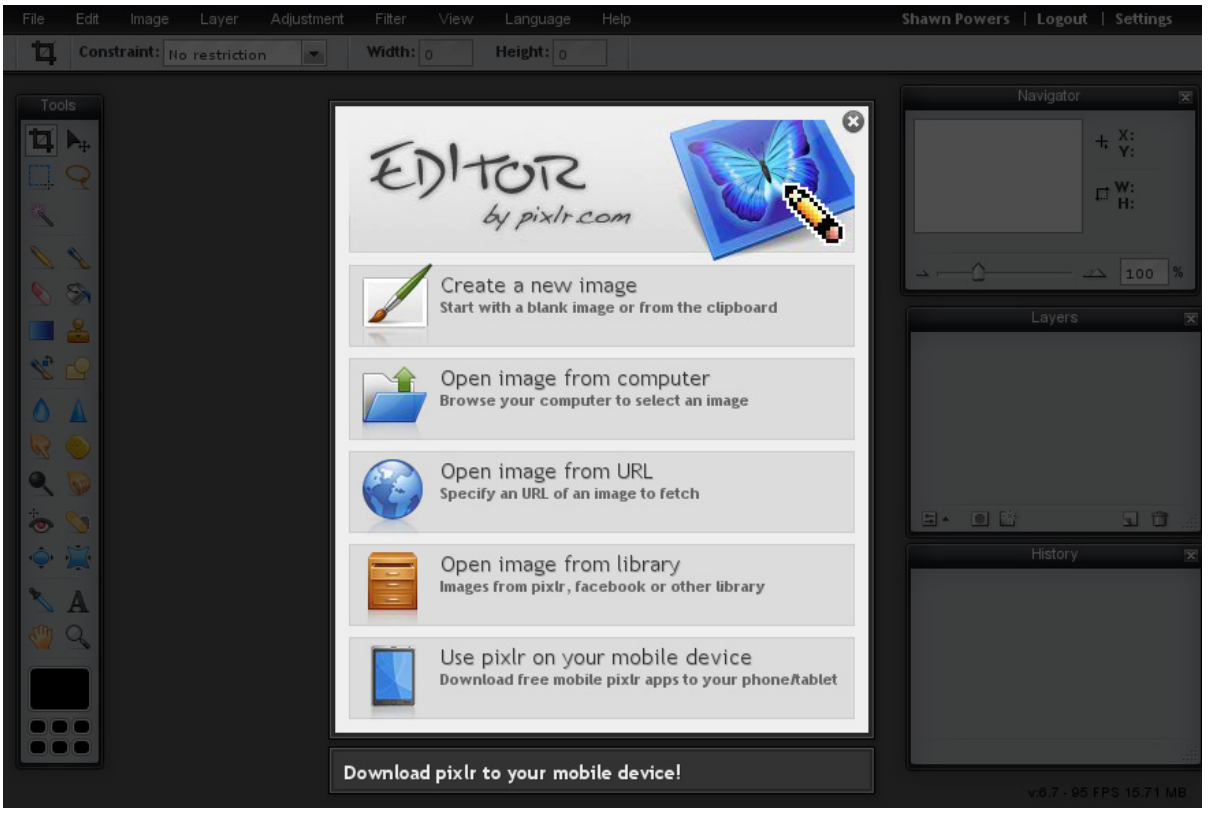
Just because you're not on Linux doesn't mean you can't have awesome open-source tools. I was having a conversation with a friend and reader (Don Crowder: @eldergeek) on Twitter the other day about music theory. Yes, I'm not just a computer nerd, but a music/math nerd too. Anyway after our conversation, I started looking for an open-source program for creating sheet music. Not only was I able to find one, but it happens to work for those folks on Windows as well as Linux.

Mind you, I'm a neophyte when it comes to music theory, but thankfully, MuseScore is useful for experts and n00bs alike. Not only can you create sheet music, but you also can download thousands of pieces others have created and shared on the Web site.

If you're a Windows user who wants to dabble in sheet music, but can't afford something like "Finale", MuseScore is right up your alley. If you're a musician who wants to give back, please join the community of users and contribute some of your music. To see how MuseScore is helping blind musicians, check out Katherine Druckman's article on our Web site: **http://www.linuxjournal.com/content/music-all-open-source-software**.

If you want to download MuseScore for yourself, you can download it from your repositories if you're on Linux, or download the installer from the Web site for Windows or OS X: **http://www.musescore.com**.

—SHAWN POWERS



GIMP Shmimp, Give Me a Browser

Don't get me wrong, I love The GIMP. We all love The GIMP, as our Readers' Choice awards show this month. If I'm being completely honest, however, I rarely have the need for such a powerful application. Usually, regardless of what computer system I'm on, I pick Pixlr as my image editing program.

Pixlr is a Web-based image editing tool that rivals native applications in speed, and more important, in functionality. Powerful tools like "spot heal" that aren't found in most simple image editors are essential for folks like me who still get teenager-like pimples in their late 30s. It also integrates with on-line storage (Flickr, Picasa, Facebook) and allows simple uploads/downloads to your local computer. In fact, you have to look really hard in order to realize Pixlr isn't a native application.

Regardless of what operating system you're on, you can check out Pixlr right now by heading to **http://pixlr.com/editor**. It's not GIMP, but it certainly isn't gimpy either.—**SHAWN POWERS**

Tinker with Molecular Dynamics for Fun and Profit

Molecular dynamics computations make up a very large proportion of the computer cycles being used in science today. For those of you who remember chemistry and/or thermodynamics, you should recall that all of the calculations you made were based on treating the material in question as a homogeneous mass where each part of the mass simply has the average value of the relevant properties. Under average conditions, this tends be adequate most times. But, more and more scientists were running into conditions that would be on the fringes of where they could apply those types of generalizations.

Enter molecular dynamics, or MD. With MD, you have to move down almost to the lowest level of matter that we know of, the level of atoms and molecules. At this level, most of the forces you are dealing with are electrical in nature. Atoms and molecules interact with each other through their electron clouds. Several packages are available for doing this type of work, such as GROMACS and GAMESS. In this article though, I take look at TINKER.

Unlike most of the software I've covered in this space, TINKER isn't available in the package systems of most distributions. This means you will have to go out and download it from the main Web site. There are binary files for Linux (32-bit and 64-bit), Mac OS X and Windows (32-bit and 64-bit). Although these should work in many cases, you probably will want to download the source code and build it with the exact options you want. You can download either a tarball or a zip file containing the source code for TINKER.

Once it is unpacked, change directory to the tinker subdirectory. There are a number of subdirectories named after the various operating system options available. Because you're using Linux, you will want to move to the linux subdirectory.

You will find a series of subdirectories for each of a number of possible compilers. For this article, I chose to use the gfortran compiler. Inside the gfortran subdirectory, you will find a number of scripts to handle each of the build steps. The first step is to run `compile.make` to build all of the required objects. These scripts need to be run from the location where the source code resides, so once you know which set of scripts you are going to use, move over to the subdirectory tinker/source. From here, I ran `../linux/gfortran/compile.make` to compile all of the source code I needed into object files.

The next step is to combine these into a single library file by running `../linux/gfortran/library.make`. The last step is to do the linking with the system libraries to create a final executable. This is done by running `../linux/gfortran/link.make`.

You now will have a full set of executable files, recognizable by filenames that end with .x. These executable files then can be moved to any other location to make them easier to use.

You should find that 61 different executable files have been created. Each of these executables handles some separate task in the analyses that TINKER is designed to do. I look at only a few different executables here to give you a flavor of the types of tasks that you can do.

The first is `analyze.x`. This executable will ask for a structure file (in the TINKER .xyz file format) and the type of analysis to run. The output you get back includes the following items: the total potential energy of the system; the breakdown of the energy by potential function type or over individual atoms; the computation of the total dipole moment and its components, moments of inertia and radius of gyration; the listing of the parameters used to compute selected interaction energies; and the energies associated with specified individual interactions.

The next executable, `dynamic.x`, performs a molecular dynamic or stochastic dynamic computation. On an initial computation, it will take a .xyz structure file as input. If a previous computation was check-pointed, you can use the resultant dynamics trajectory file (or restart file) as input too. These two programs are both deterministic in their methods.

The executable `monte.x` provides a way to apply Monte Carlo minimization methods to molecular dynamics. It takes a random step for either a single atom or a single

torsional angle, then applies the Metropolis sampling method.

The `scan.x` executable takes a .xyz structure file as input and finds an initial local minimum. From this first local minimum, the program starts searching out along normal modes to try to find other minima. Once it has searched along each of these modes, it then will terminate.

A number of these 61 executables are support utility programs that do non-computational work. For example, the executables `xyzint.x` and `intxyz.x` convert back and forth between the .xyz structure file format and the .int internal coordinates formatted file.

For all of these programs, the specific details of how they work is determined by a keyword file (with a filename ending with .key). TINKER uses a huge number of keywords to decide the specifics of any particular run. For example, you could set a single bond stretching parameter with the keyword BOND. The keyword CHARGE will set a single atomic partial charge electrostatic parameter. A full listing

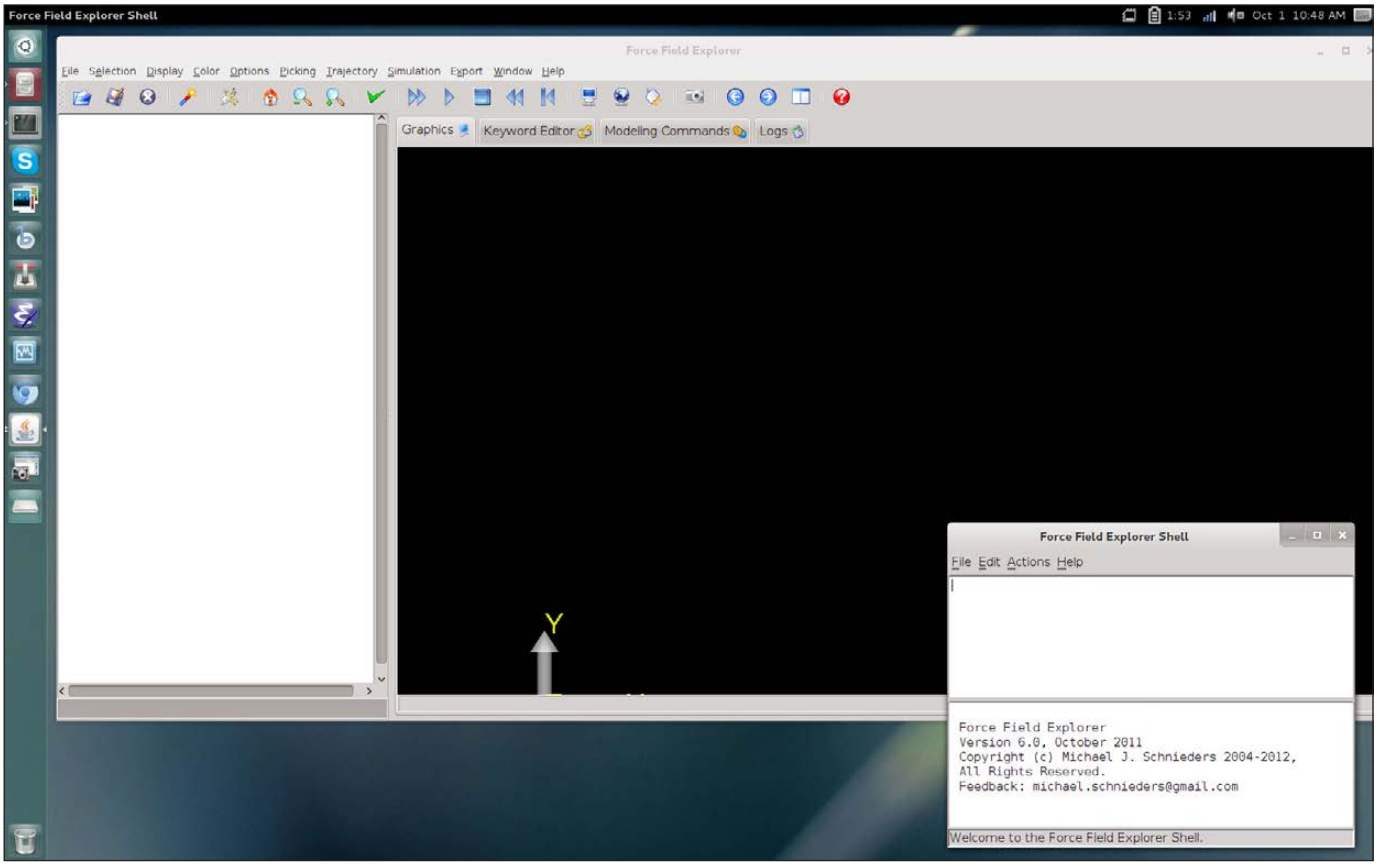


Figure 1. On initial startup, you will get an empty project window and a TINKER console.

of the keywords is available in the TINKER documentation.

All of these executables are designed to run as command-line programs. The output tends to be files of numbers, which are hard for a human to evaluate. The group who created TINKER also created a program called Force Field Explorer (FFE).

The executables built above are not compiled to interface with FFE as is. If you want to compile your own copy and have it interact with FFE, it requires changing a number of source files. In this case, I would suggest that you go ahead and download one of the installation packages that include FFE. These come as a gzipped shell archive. After gunzipping it, you can run the shell script to start up the Java-based installer. It will let you select which portions to install along with FFE. Once it is all done, go ahead and start up FFE. It will open up the main window and a console window. From within FFE, you can load up structure files and start various TINKER analyses.

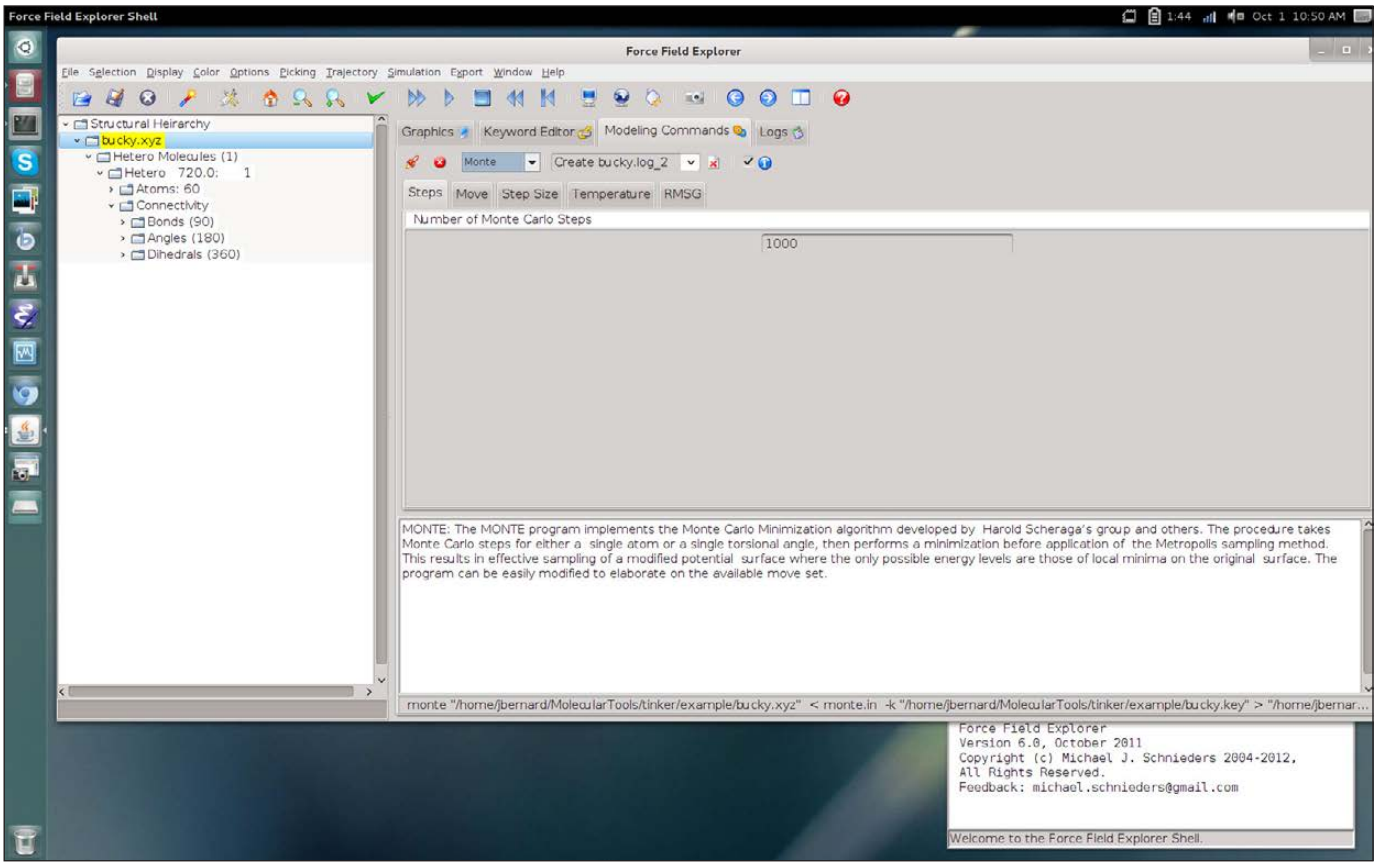


Figure 2. You have access to all of the TINKER analysis routines, directly from FFE.

When you first open an .xyz file, the structure is rendered and displayed in the main window. You then can select the Modeling Commands tab to select which specific TINKER analysis to run. By default, these TINKER runs happen locally on the same machine, but it doesn't have to be this way. FFE gives you the option of connecting to a remote machine, likely more powerful than your desktop, and getting the actual TINKER programs to run over there.

Once you have results, you can change the visual details like colors and whether to use wireframe or tube and so on. You also have the option of exporting a visual as an image file in one of several file formats.

I easily could fill the entire contents of *Linux Journal* just covering the most basic functionality TINKER provides. Hopefully, you will have seen enough to get an idea of whether this software might be of use to you. If so, a rather large amount of detailed documentation is available at the main TINKER Web site.

Resources

- Main TINKER Web Site: **http://dasher.wustl.edu/tinker**
- An Introduction to Molecular Dynamics: **http://en.wikipedia.org/wiki/Molecular_dynamics**

—JOEY BERNARD

They Said It

The love of learning, the sequestered nooks, / And all the sweet serenity of books....
—Henry Wadsworth Longfellow

All that counts in life is intention.
—Andrea Bocelli

He that climbs the tall tree has won right to the fruit.
—Sir Walter Scott

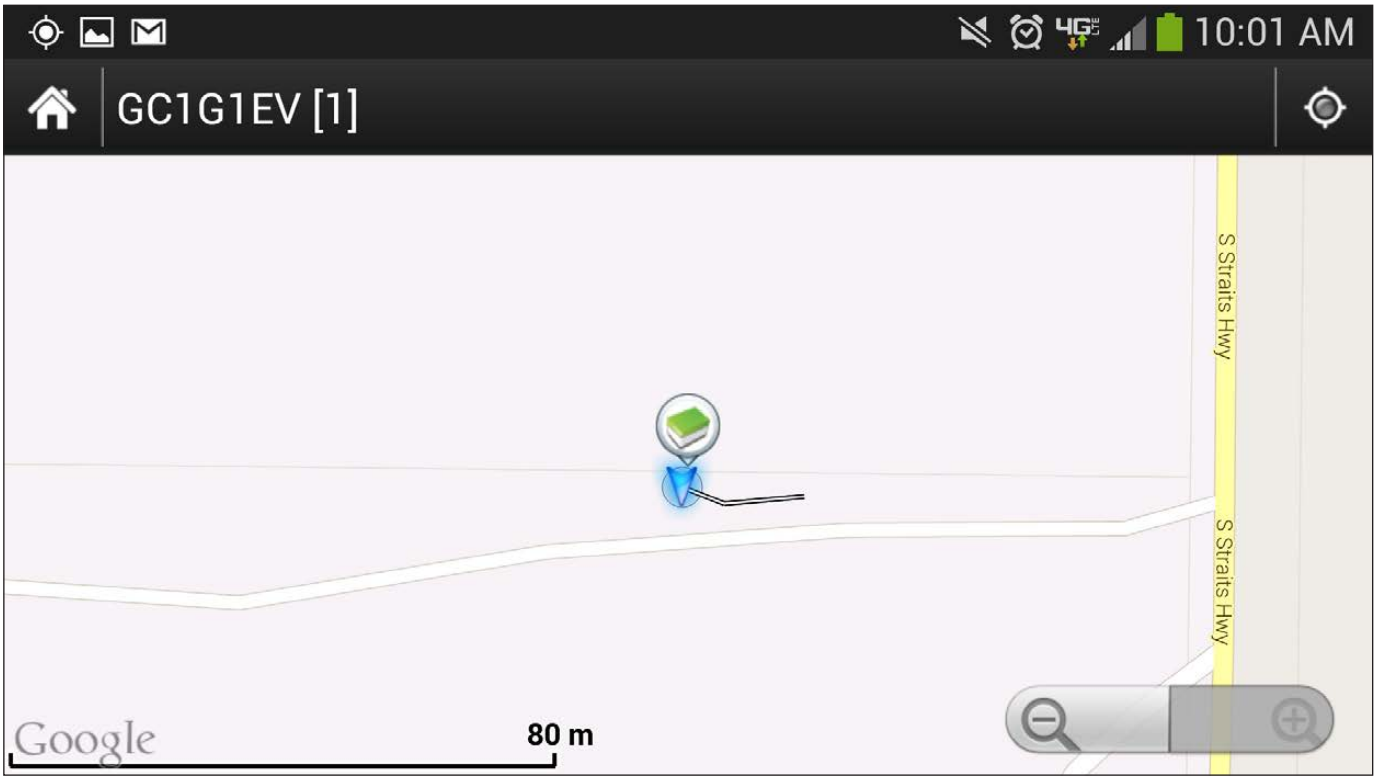
Wear the old coat and buy the new book.
—Austin Phelps

We can't take any credit for our talents. It's how we use them that counts.
—Madeleine L'Engle

Android Candy: Free, Family, Fun—Fantastic

I've mentioned geocaching before, but if you've never taken the time to go out and do it, you're really missing out. Whether you're dragging your family through two feet of snow in the middle of the woods (yeah, I did that last year, I'm still not sure they've forgiven me) or following your GPS around a parking lot looking for a tiny micro-cache, geocaching is *fun*. You need only a few things to go geocaching:

1. A sense of adventure.
2. Friends or family (not required, but more fun).
3. A sharp mind (there often are brain puzzles involved).



4. A GPS or smartphone app to guide you.

That's where c:geo comes in. There are several geocaching apps for Android, and they vary in price from free to very not free. The c:geo app is one of the free ones, and it also happens to be one of the best. It will show you clues, help you find local geocaches and guide you on a map to the GPS location you need. Whether you're a hard-core geocacher or just want to go out for a little fun with the family, c:geo is a great tool for your Android device that will make geocaching easier and more enjoyable. You can find it in the Google Play Store.

Because it's such an incredible application, and because it relates to our three favorite F words (Free, Family, Fun), c:geo gets our Editors' Choice award this month. Download it now, and get out there and find stuff: http://www.cgeo.org.

—SHAWN POWERS

REUVEN M. LERNER

2013 Book Roundup

Reuven's annual summary of interesting books.

I'm always amazed to hear about the death of the publishing industry. True, books and (gulp) magazines are often fighting for their lives, and the state of journalism is in tatters. But at the same time, we continue to see a large number of high-quality books being published. This past year was no exception; I read many books that really enlightened me, giving me new ideas in areas of technology, business and life in general.

So as I do at roughly this time every year, here's a roundup of the most interesting books I've read during the last year. This is not a representative sample; it reflects a combination of books that I bought and received for review, generally because I saw or noticed them. I expect there are many good books I haven't read, but that just means they'll likely be on the list for next year. This also means there almost certainly are some books on this list that I saw for the first time in the past year, but which were published before then.

Programming Languages

As someone who develops software on a regular basis, and also teaches programming to a large number of people, programming languages still are a subject that I enjoy reading and learning about. The languages I use most often—Ruby, Python and JavaScript—are the subjects of a constant stream of books, many of which are no longer simple tutorials, but explorations of specific topics that will be of interest to many developers working with the language.

Python for Data Analysis by Wes McKinney (O'Reilly, ISBN: 978-1-449-31979-3) is an introduction to manipulating data with two well-known Python libraries, NumPy (for numeric analysis and some highly efficient data structures) and Pandas (for data analysis). Reading through the description of these libraries reminded me greatly of the excellent R language for data analysis and manipulation, as well as the relational

algebra that we know (and love!) in SQL. The advantage of Pandas is that it allows you to integrate the analysis into a language you are already using, rather than having to learn a new one. The book is full of examples and practical hints; if you ever have wanted to learn how to analyze, manipulate and plot your data, this is a great way to get started, in an excellent and readable language.

Python is remarkable in numerous ways, among them the fact that it is an excellent language for beginning programmers, as well as for experienced professionals. Jason Briggs has written *Python for Kids* (No Starch Press, ISBN: 978-1593274078), which introduces programming to children aged ten and up through playful examples, including GUI-based programs using the Tk library. The use of child-friendly examples and humor (for example, "Want to hear a dirty joke? A pig fell in the mud!") makes me want to re-start the programming lessons that I've given my own children.

Ruby, the language I use most often in my day-to-day work, has been the focus of several excellent books through the last year. Perhaps the most celebrated book is *Practical Object-Oriented Design in Ruby* by Sandi Metz (Prentice Hall, ISBN: 978-0321721334). If you are familiar with object-oriented programming principles, the ones Metz mentions will not be new. However, the examples that Metz provides are so clear, and the practical guidelines and suggestions that stem from these ideas so compelling, that this book is a must-read for everyone in the Ruby community, and even for those using other object-oriented languages.

In a similar vein is *Confident Ruby* (**http://confidentruby.com**), a self-published e-book by Avdi Grimm. Grimm starts off with the assertion that Ruby makes software so easy to write, and to write quickly, that we do so too quickly, creating impossible-to-understand libraries. Grimm's writing is always clear and interesting, and in this book, he breaks down the types of code we are likely to write most often—classes and methods—and gives us guidelines for doing so faster and better.

My last choice on the Ruby front is another self-published book, *Working with Ruby Threads* by Jessie Storimer (**http://www.jstorimer.com/products/working-with-ruby-threads**). I must admit that I'm one of those programmers who hasn't worked much with threads over the years (since Web applications tend not to use them much), so I've gained a bit of an aversion to threads. This book really

opened my eyes to issues regarding threading, the differences between Ruby implementations in this area and alternative solutions to the problems.

As a Web developer, the third language that I tend to use a fair amount is JavaScript. JavaScript is an increasingly popular language, both because it is ubiquitous and implementations have been increasingly speedy. JavaScript, for all of its flaws, is here to stay, and all Web developers need to understand how it works, rather than treat it as a language they're forced to work with.

Two books that came out in the last year from O'Reilly provide examples and information that you can use to improve your understanding and methodology of working with JavaScript. *Testable JavaScript* by Mark Ethan Trostler (ISBN: 978-1449323394) shows a number of different techniques for testing JavaScript code and describes how to put them in action, using such tools as PhantomJS. *Learning from jQuery* by Callum Macrae (ISBN: 978-1-4493-3519-9) is aimed at those developers who know how to do things from within jQuery, but don't understand how to do those same things from the underlying JavaScript. For someone who has been working in JavaScript for a while, this book probably will be unnecessary, but if you are a "jQuery developer" rather than a "JavaScript developer", it might well help make the transition and deepen your understanding of JavaScript.

JavaScript has a number of quirks that make it difficult for programmers to move, conceptually, from other languages. As such, a number of books have been aimed at helping programmers make the transition. A great book on this front is *JavaScript Allonge* by Reginald Braithwaite (**https://leanpub.com/javascript-allonge**), available both as a purchased PDF and as a free HTML version for on-line reading. The book is funny, interesting and helps programmers really understand why JavaScript works the way it does. I have been using JavaScript since it was invented, in a number of different ways, and I still enjoyed the style and content of Braithwaite's book.

You might also want to consider *JavaScript Enlightenment* by Cody Lindley (ISBN: 978-1-4493-4288-3), which is organized more as a cookbook of programming paradigms in JavaScript, as opposed to *JavaScript Allonge*, which has more of a narrative style.

Of course, other programming languages exist, and I've been trying to dip my toes into those waters. I've

played a little bit with Erlang in the past few years, and although I'm not convinced I'll do a lot of work with it, I do like many of its ideas and find it useful and enlightening to try it out. *Learn You Some Erlang for Great Good* by Fred Hebert (No Starch Press, ISBN: 978-1-59327-435-1) is a large book (600+ pages), which introduces the Erlang language and its many facets, including a great emphasis on testing.

Several months ago, I covered the basics of Web development using the Clojure language, a modern Lisp that runs on the JVM. *Web Development with Clojure* by Dmitri Sotnikov (Pragmatic Programmers, ISBN: 978-1-937785-64-2) is a gentle introduction to creating Web applications in Clojure, using such libraries as Compojure. The book is aimed at beginners, and it seems to do a solid job of introducing the features of the language that will be useful for Web development. If you are interested in this area and haven't found the online tutorials to be sufficiently detailed or helpful, this book probably will serve you well.

Concepts and Techniques

A number of books aim not at discussing a particular language, but rather ideas and techniques for working with them. Most intriguing among them is *Seven Concurrency Patterns in Seven Weeks* by Paul Butcher (Pragmatic Programmers, ISBN: 978-1-93778-565-9). It assumes that threading is difficult and bad, and looks at alternatives to threading to enable safe, concurrent execution. I would say that this book is a good followup to Jessie Storimer's book on Ruby and threads, but this is likely a good read for anyone who is working with threads and wants to find a better way for programs to remove the dangers and frustrations of thread-based programs.

Avdi Grimm's second book of the year, *Much Ado about Naught* (**https://shiprise.dpdcart.com/cart/view?product_id=64334**), is a surprisingly interesting read about a subject I quite frankly never expected to think or talk about. The fact that the book is written using TDD techniques also is a good introduction to the subject for people who are new to testing.

Finally, Facebook and LinkedIn might be well known social networks, but the world of social-network analysis has existed for many years and provides techniques that can help you understand your users better. *Social Network Analysis for Startups* by Maksim Tsvetovat and Alexander Kouznetsov (ISBN: 978-1449306465) introduces the core ideas of SNA

and then demonstrates how to apply them using Python code. If your Web applications involve groups of people, you might well be able to benefit from this book to see how they are connected.

Frameworks

Increasingly, Web developers don't create applications with just a programming language, but in a framework as well. Nowadays, Web developers typically use two different frameworks, one for the server (such as Ruby on Rails or Django) and another for the browser, written in JavaScript (such as AngularJS or Backbone).

The Pragmatic Programmers, continuing its *Seven ___ in Seven Weeks* series (including the book on concurrency patterns mentioned above), have come out with *Seven Web Frameworks in Seven Weeks* by Jack Moffitt and Fred Daoud (ISBN: 978-1-93778-563-5), which covers some server-side frameworks and some client-side ones. The idea, as with their other books, is to give you experience and understanding of the frameworks—not to become an expert in them, but rather to gain an appreciation for the ways they work. Just as learning a new programming language can be a useful and enlightening experience, so too can learning a new Web application framework. Experienced Web developers who are eager to learn new paradigms definitely should take a look at this book.

Die-hard Rails developers will be delighted to see that the Pragmatic Programmers have released an update to *Crafting Rails 4 Applications* by Jose Valim (ISBN: 978-1-937785-55-0). This book doesn't even pretend to teach MVC, Rails conventions or anything else that nearly every Rails book starts with. Instead, the first chapter starts with the creation of a Rails plugin, and things get hairier and more interesting from there. If you are working on complex Rails applications, or just want to gain a better appreciation and understanding of the framework, you likely will want to read this book.

Client-side frameworks, written in JavaScript, continue to be popular. But in the last year or two, we have seen growing interest in full-scale MVC frameworks, beyond the bare-bones capabilities that Backbone and its ilk offer. The two titans here are Ember.js and AngularJS, both of which I intend to review and discuss in this column in coming months. There doesn't yet seem to be a book about Ember, but there is a short book called *AngularJS* by Brad Green and Shyam Seshadi (O'Reilly, ISBN: 978-1-449-34485-6).

This book gives a good, quick introduction that is easier (for me, at least) to understand than the official documentation and tutorials.

Other Reads

Although I do read (and listen, and watch) a great deal about technology, I also try to find time to read on other subjects as well.

My first recommendation isn't for a book, but rather a magazine. I grew up hearing about *The New Yorker* as a source of witty cartoons but not much else. I can now tell you, having subscribed for a bit more than a year, that the magazine itself is fun, interesting and thought-provoking, with some of the best writing I've ever had the chance to enjoy. If you want to read about all sorts of interesting topics—as well as tickle your funny bone with the cartoons—I heartily suggest reading *The New Yorker*.

My favorite nontechnical book of the year is *Thank You For Arguing* by Jay Heinrichs (Three Rivers Press, ISBN: 978-0385347754). I found this book to be amazingly insightful and interesting, as well as funny, and it taught me the basics of rhetoric—the art of persuasion. The author breaks rhetoric into a number of basic tools and then uses those building blocks to demonstrate how to make a convincing argument.

I continue to be fascinated by China, now that I've been there twice. Three interesting China-related books I read during the summer were: *Factory Girls* by Leslie Chang (ISBN: 978-0385520188), which describes the reality of the young women who leave the countryside in search of a better future; *China Airborne* by James Fallows (ISBN: 978-0375422119), which uses aviation as a way to understand modern Chinese decision-making; and *Last Days of Old Beijing* by Michael Meyer (ISBN: 978-0802717504), a bittersweet book about how the Chinese government is removing historic, old buildings and putting new ones in their place.

I hope that you enjoyed this year's book roundup. I'm always on the lookout for new book suggestions, so if you have any titles to offer, I'd be delighted to hear about them.■

Web developer, trainer and consultant Reuven M. Lerner is finishing his PhD in Learning Sciences at Northwestern University. He lives in Modi'in, Israel, with his wife and three children. You can read more about him at http://lerner.co.il, or contact him at reuven@lerner.co.il.

Send comments or feedback via http://www.linuxjournal.com/contact or to ljeditor@linuxjournal.com.

DAVE TAYLOR

Resizing Images with ImageMagick

Sure, you can open up a graphics program like GIMP and resize an image, but what if you want to resize 10, 50 or 200 images? ImageMagick's `convert` program is just what you need.

In the October 2013 issue, I started a series on working with ImageMagick on the command line, but then I had to stop and deal with the massive migration project of moving my AskDaveTaylor.com site from one server to another while simultaneously dropping it into a completely different back-end software system—madness. I'm still fixing things and cleaning up the insane sprawl of it all.

So, my last article detoured into a discussion of scripts that helped with the migration process. I'm still working on these fast, short scripts, including one I wrote this morning:

```
for entry in blog/*
do
  new=$(echo $entry | sed 's/blog\///')
  echo "Redirect 301 $entry $new"
done
```

Can you track what this loop does? The only tricky part is the `new=` statement that removes `blog/` from the filename matched in the `for` statement; otherwise, it's quite straightforward.

Seriously though, let's return to ImageMagick. There are a ton of things you can do with the command-line utilities. But first, let me look at where I left off.

I'd just shown a simple example of ImageMagick command-line tools to identify the dimensions of an image and use that as the basis of coming up with a scaled HTML img tag. Here's the script:

```
#!/bin/sh

identify=/usr/bin/identify

scale=$1

image=$2   # needs input validation code
```

```
height=$($identify $image | cut -d\   -f3 | cut -dx -f1)
width=$($identify $image | cut -d\   -f3 | cut -dx -f2)
newwidth="$(echo $width \* $scale | bc | cut -d. -f1)"
newheight="$(echo $height \* $scale | bc | cut -d. -f1)"
echo "<img src=$image height=$newheight width=$newwidth>"
exit 0
```

(Actually, I couldn't resist tweaking it slightly if you are keeping track, but I'm still being lazy and not validating the input as of yet. You can add that code easily enough.)

In use:

```
$ scaledown.sh 0.5 pvp.jpg
<img src=pvp.jpg height=152 width=485>
```

Okay, that's one way to make the display of the image be reduced on a Web page, but anyone who has done any work trying to speed up a Web site knows the huge problem here: reducing the container that displays an image doesn't reduce the image. The Web site visitor still has to download the original image, which is a huge waste of bandwidth and a performance hit.

So let's update the script to create a new, smaller version of the image as part of its output.

Enter the convert Command

The `identify` command is a great way to learn specific information about a graphical image file, but to manipulate it, you need to switch to `convert`.

There are a million command-line options to `convert`, but the one I use here is `-resize`, like this:

```
$ convert pvp-big.jpg -resize 0.5 pvp-0.5.jpg
$ identify pvp-big.jpg pvp-0.5.jpg
pvp-big.jpg JPEG 970x305 970x305+0+0 8-bit DirectClass 127kb
pvp-0.5.jpg JPEG 1x1 1x1+0+0 8-bit DirectClass 1.1kb
```

Hmmm...you can see what's happened, right? The image went from 970x305 to 1x1. Yikes.

How did that happen? The problem is that I'm handing the wrong kind of parameter to the `-resize` option. In fact, it wants a percentage (weirdly enough), so `-resize 50%` or `-resize 50` both work:

```
$ convert pvp-big.jpg -resize 50 pvp-50.jpg
$ convert pvp-big.jpg -resize 50% pvp-50%.jpg
$ identify pvp*
pvp-50.jpg[1] JPEG 50x16 50x16+0+0 8-bit DirectClass 2.01kb
pvp-50%.jpg[2] JPEG 485x153 485x153+0+0 8-bit DirectClass 44.7kb
pvp-big.jpg[3] JPEG 970x305 970x305+0+0 8-bit DirectClass 127kb
```

A bit of mathematics reveals that `-resize 50` meant that the width was scaled to 50 pixels, with the height proportionally scaled down to a tiny 16 pixels. `-resize 50%`, on

So the script will need users to enter a proper percentage amount or otherwise compensate.

the other hand, accomplished the goal, scaling the image down to a reasonable 485x153.

So the script will need users to enter a proper percentage amount or otherwise compensate. To make it more interesting, let's make the output filename gain a suffix that denotes the new geometry (as ImageMagick likes to refer to the height x width values). In this instance, the goal is to have pvp-big.jpg shrink 50% and be copied as pvp-big.285x153.jpg.

Rather than use the `bc` statements from the original script, let's make ImageMagick do the work by having this workflow:

1. Convert image to resized image and save as temp file.
2. Use `identify` to get new dimensions of temp file.
3. Create new filename based on geometry.
4. Rename temp file to new filename with geometry specified.

It turns out it's a lot less work, because mathematics are no longer required, which is a good thing!

The hardest part is to create the new filename, which involves more lines of code than the conversion itself. It involves figuring out the filename suffix, chopping the filename up and building a new one that inserts the new image geometry in the middle.

Here's the result (it's long):

```
#!/bin/sh

convert=/usr/bin/convert

identify=/usr/bin/identify

resize=$1

source=$2

if [ -z "$resize" -o -z "$source" ] ; then

  echo "Usage: $0 resize sourcefile"; ;exit 1

fi

if [ ! -r $source ] ; then

  echo "Error: can't read source file $source" ; exit 1

fi

# let's grab the filename suffix

filetype=$(echo $source | rev | cut -d. -f1 | rev)


tempfile="resize.$filetype" # temp file name


# create the newly sized temp version of the image

$convert $source -resize $resize $tempfile
```

```
# figure out geometry, the assemble new filename

geometry=$($identify $tempfile | cut -d\   -f3 )


newfilebase=$(echo $source | sed "s/$filetype//")

newfilename=$newfilebase$geometry.$filetype


# rename temp file and we're done

mv $tempfile $newfilename


echo \*\* resized $source to new size $resize. result = $newfilename


exit 0
```

That's it. It's not incredibly complicated if you go through it step by step. In fact, go back to the four-step algorithm I presented earlier. That's almost exactly duplicated in the comments within the script.

The only nuance is the sequence for `newfilename` assembly, which just strings together a series of variables to have their values tucked together.

Let's give it a whirl and see what happens:

```
sh resize.sh 50% pvp.jpg

** resized pvp.jpg to new size 50%. result = pvp.485x153.jpg
```

I'm skeptical, so let's test the new image file by using `identify` to get its dimensions:

```
$ identify pvp.485x153.jpg

pvp.485x153.jpg JPEG 485x153 485x153+0+0 8-bit DirectClass 44.7kb
```

Perfect. More important, look at how the image size has shrunk as a result of it being scaled down 50%:

```
$ ls -l pvp.jpg pvp.485x153.jpg

-rw-rw-r-- 1 taylor taylor  45751 Oct  9 04:14 pvp.485x153.jpg

-rw-r--r-- 1 taylor taylor 130347 Sep  5 08:20 pvp.jpg
```

A definite win and a pretty handy script to keep around.

Of course, better positional parameter checking and a quick check to ensure that the resize parameter isn't something crazy would be good coding, but it's not a bad script that serves a very useful purpose.

So that's it. In my next article, I plan to take a look at adding embossing—text that's superimposed over a graphic—as an easy way to watermark sets of photos from the command line. Until then, cheerio!■

Dave Taylor has been hacking shell scripts for more than 30 years. Really. He's the author of the popular *Wicked Cool Shell Scripts* and can be found on Twitter as @DaveTaylor and more generally at http://www.DaveTaylorOnline.com.

Send comments or feedback via http://www.linuxjournal.com/contact or to ljeditor@linuxjournal.com.

KYLE RANKIN

Secret Agent Man

Can you have your security and your convenience too? It turns out with SSH the answer is yes.

It used to be that only the paranoid among us focused on strict security practices, yet these days, it seems like people are stepping up their games with respect to encryption, password policy and how they approach their computers in general. Although I always have considered myself more inside that paranoid camp than outside of it, I even have found myself stepping up my game lately. Security is often at odds with convenience, yet whenever I need a good example of better security practices that are *more* convenient than the alternative, I turn to SSH keys.

With SSH keys, you generate a private and public key pair with the `ssh-keygen` command and distribute the public key to servers to which you want to connect. SSH keys use your private key to authenticate yourself instead of a password on the remote server, so if you are one of those people who are worried about SSH brute-forcing, if you use SSH keys, you can disable password SSH authentication altogether and not care about those SSH brute-force attempts you see in your logs. When I used to set up SSH key pairs, I wouldn't provide a passphrase to unlock the key. Without a passphrase, I could just `ssh` in to a machine without typing any sort of password—a case where you can increase security against brute-force SSH attacks while also increasing your convenience.

Of course, the problem with skipping the passphrase when you generate SSH keys is that all of your security relies on keeping your private key (usually found at ~/.ssh/id_rsa or ~/.ssh/id_dsa) secret. If others were able to get a copy of that file, they could log in to any machine to which you distributed the public key. Lately I decided I didn't like that kind of risk, so when I generate SSH keys, I now use a passphrase. This means

if others got my private key, they couldn't immediately use it, but it also means I now have to type in a passphrase to use my SSH key. This is less convenient, but I've found that by using SSH agent, I can get back to a similar level of convenience but with a few added bonuses that I discuss in this column.

SSH Agent

On most systems that use sudo, after you type in your sudo password, it is cached for some period of time, so if you run a few sudo commands in a row, you don't have to keep typing in your password. SSH agent works in a similar way for SSH passphrases, caching your unlocked key in memory for a defined period of time. This is particularly useful if, like me, you use Git on a routine basis with SSH—it would be a pain to have to type in your passphrase every time you do a git push or git pull. So for instance, if I wanted to cache my passphrase for 15 minutes, I could type:

```
$ ssh-add -t 15m
```

Then after I provide my password a single time, it would be cached for the remainder of SSH commands I run within that 15 minutes, after which it would expire.

SSH Alarm Clock

Because you are prompted for a password after the timeout you set expires, one of the first uses that came to mind for the `ssh-add` command was an alarm clock of sorts. Sometimes when you are deep in your work, you can forget to do things like eat lunch. What I like to do when I start work for the day is calculate how long until I'd like to break for lunch and set `ssh-add` to that. For instance, if I start work at 9am, and I want to break for lunch at noon, I would just type:

```
$ ssh-add -t 3h
```

Then when noon rolls around, I'll notice, because my next git push or pull, or my next SSH session, will prompt me for a password. Currently I take a ferry into work, and the ferry has a fixed time that it leaves. I know I need to leave the office around 5:30pm to catch that ferry, so once I get back from lunch, I calculate how many hours (or minutes if I want to be that fine-grained) until then and run a new `ssh-add` command. This alarm clock even has a sort of snooze feature where I can run another `ssh-add` command to add an extra nine minutes if I want to finish up something before I leave.

SSH Agent Forwarding

Of course, the traditional nice feature SSH agents give you is the ability to forward on your credentials to a server you have logged in to. When you are a sysadmin, you often run into an issue where you'd like to `scp` a file between servers, but if you have disabled password authentication for SSH (and you should), that could mean putting your private key on your servers, which you may not want to risk. With SSH agent forwarding, your SSH credentials from the private key on your local machine are forwarded to a machine you `ssh` in to, in RAM, and if you `ssh` to another machine from there, it will use those credentials.

There is a potential risk with agent forwarding. I think the ssh_config man page says it best:

> Agent forwarding should be enabled with caution. Users with the ability to bypass file permissions on the remote host (for the agent's Unix-domain socket) can access the local agent through the forwarded connection. An attacker cannot obtain key material from the agent, however they can perform operations on the keys that enable them to authenticate using the identities loaded into the agent.

All that said, to use agent forwarding, just add `-A` to any SSH command you normally would run:

```
$ ssh -A user@remotehost
```

Alternatively, you also can set the ForwardAgent setting in a local SSH config file, so you can control which hosts automatically get agent forwarding and which don't.

I love it when adding security can add convenience. While adding a passphrase to my SSH key potentially could have added a big inconvenience in the name of security, I think the benefit of an alarm clock, plus the general ability of ssh-agent to allow me to forward credentials to remote servers without having to risk compromising my private key far outweighs any inconveniences of managing a passphrase or SSH keys in general.■

Kyle Rankin is a Sr. Systems Administrator in the San Francisco Bay Area and the author of a number of books, including *The Official Ubuntu Server Book*, *Knoppix Hacks* and *Ubuntu Hacks*. He is currently the president of the North Bay Linux Users' Group.

Send comments or feedback via http://www.linuxjournal.com/contact or to ljeditor@linuxjournal.com.

SHAWN POWERS

LVM, Demystified

It's complicated and time consuming, but totally worth it.

I've been a sysadmin for a long time, and part of being a sysadmin is doing more than is humanly possible. Sometimes that means writing wicked cool scripts, sometimes it means working late, and sometimes it means learning to say no. Unfortunately, it also sometimes means cutting corners. I confess, I've been "that guy" more than once. A good example is SELinux. On more than a few (hundred!) occasions, I've simply disabled SELinux, because getting things to work right is often really frustrating and time consuming. The same is true with LVM (Logical Volume Manager). I didn't get it. I thought it added an unnecessary layer of complexity. I thought it meant another potential point of failure. I thought it was stupid.

I was wrong.

LVM is an incredibly flexible, ridiculously useful and not terribly complicated to use system. It makes life easier. It makes future storage upgrades and migrations simple. Quite simply, I love it. So in this article, I cover the concepts and usage of LVM. By the time I'm done, hopefully you'll love it as much as I do!

What LVM Is

The best analogy I can come up with for explaining LVM is a SAN. If you've ever used a SAN (Storage Area Network) in your server environment, you know it abstracts the idea of individual hard drives and allows you to carve out "chunks" of space to use as drives. Rather than worrying about how big your hard drives might be, a SAN lets you throw all your hard drives into a big chassis and then allocate space to individual clients without being concerned about how many or how few physical drives are being used. LVM is sort of like that, but for an individual system rather than an entire network.

Figure 1 shows my poor attempt at drawing the concept of an LVM system. At first glance, it might

seem like using an LVM is silly. Why combine a bunch of drives together, only to carve them up into virtual drives, right? Thankfully, that simple concept gives incredible flexibility down the road. Need a great big partition, but have only a bunch of smaller disks? No problem. Have only a couple disks now, but want to add more later without reformatting? No problem. Need to take snapshots, like with virtual servers, but you're using actual bare metal? No problem. LVM makes dealing with storage far better than partitioning drives or using a simple RAID setup (which, incidentally, brings me to the next issue).

What LVM Isn't

With all the flexibility and expandability I mentioned in the previous paragraph, it seems like LVM would be a perfect replacement for hardware- or software-based RAID. After all, one of the big advantages of RAID is that multiple smaller drives can be used as a single, larger drive. For that particular feature, LVM is indeed ideal. Unfortunately, however, LVM doesn't provide any options for redundancy or parity. That means if you have a drive fail in LVM, you lose data. There's no such thing as striped LVM or mirrored LVM; it's simply not designed to do that.

LVM also isn't designed to increase

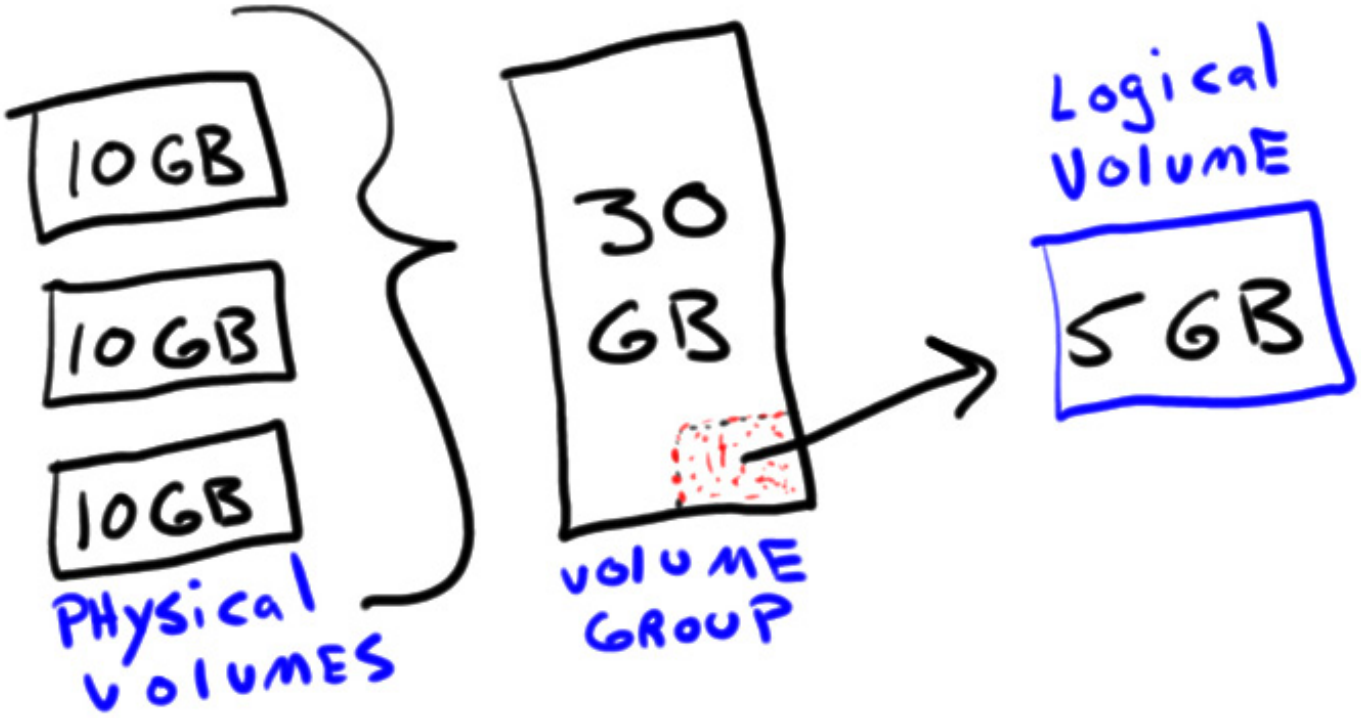


Figure 1. It's important to think of my drawings as art—possibly second-grade art.

speed by striping reads and writes across multiple disks. As block devices in the volume group fill up, such simultaneous read/writes may occur, but it's not by design and certainly not to gain speed. Hopefully, it's clear: LVM is really cool, but it is not in any way a replacement for RAID. Thankfully, it doesn't need to be.

(Note: recent versions of LVM actually do provide striping and mirroring features. In some cases, it can take the place of RAID completely. I still think understanding them as separate concepts is important. If you want to learn more about LVM and utilize the RAID features, I'll leave that as an exercise for you.)

Two Peas in a Pod

If you look at the first "stage" in my drawing (Figure 1), you'll notice that I didn't call the 10GB chunks "drives"; I called them physical volumes. That's because although it's certainly possible to use a physical drive as a physical volume in LVM, it's not a requirement. In fact, it's not even the most common scenario. In most production environments, LVM is used in combination with RAID. Whether that's hardware-based RAID or software-based RAID, having your underlying physical volumes exist as RAID devices is ideal.

As someone who has had problems with hardware-based RAID arrays, I tend to lean toward software-based RAID in my systems. That's certainly a matter of personal preference, but it's good to know that since software-based RAID and LVM both operate at the kernel level, both are extremely efficient. Software-based RAID admittedly uses some CPU, especially when rebuilding arrays, but LVM uses very little. If I/O performance is of utmost importance for your purposes, it's worth doing some research and possibly testing before committing to any solution.

Getting Started

Although it's certainly possible to transition to an LVM system after Linux is already installed, it's far more preferable to do so during the initial setup. Most distributions allow for LVM setup to take place during the installation process, and in the case of CentOS and RHEL, LVM is used by default. Even if you're installing only onto a single, non-RAID hard drive, setting up LVM allows you flexibility and expansion opportunity later. Heck, it's possible to add RAID to a server later on, then simply migrate the data from your original physical volume to the RAID physical volume. That's far easier than using `dd`,

especially when you'd like to keep your server running!

Because this is an introduction, let me start with a simplistic setup. Let's say you have two hard drives, /dev/sdb and /dev/sdc. With LVM, any block device can be used as a physical volume (PV), which means you can use either partitions or entire drives. If you need to have a "traditional" partition (in some cases, the /boot partition might need to be on a regular, non-LVM device), be sure to partition the drive before adding the physical volumes to your volume group. In this example, let's use the raw disks themselves.

Step 1: Create Physical Volumes

Once you have the block devices you want to add to your volume group (again, keep referring to my drawing if the terms get confusing), you need to establish them as LVM physical volumes. To do that, use the `pvcreate` command:

```
pvcreate /dev/sdb
pvcreate /dev/sdc
```

These commands configure the drives as potential candidates to be added to a volume group. If you want to make sure it worked correctly, you can type `pvdisplay` or `pvscan` to show the status of any existing LVM Physical Volumes:

```
$ sudo pvdisplay
  --- Physical volume ---
  PV Name               /dev/sdb
  VG Name
  PV Size               10.4 GiB / not usable 3.00 MiB
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              4994
  Free PE               4994
  Allocated PE          0
  PV UUID               SRKAXh-EpYr-r2td-g0gA-31RA-fnfz-3qqGrO

  --- Physical volume ---
  PV Name               /dev/sdc
  VG Name
  PV Size               10.4 GiB / not usable 3.00 MiB
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              4994
  Free PE               4994
  Allocated PE          0
  PV UUID               t2cKru-IwMy-I8re-ADp2-vzFF-Tvh5-O4zMhI
```

And, the simpler `pvscan`:

```
$ sudo pvscan
  PV /dev/sdb         lvm2 [10.4 GiB]
  PV /dev/sdc         lvm2 [10.4 GiB]
  Total: 2 [20.8 GiB] / in use: 0 [0 ] / in no VG: 2 [20.8 GiB]
```

Once you create the volume group and logical volumes, go ahead and

run these commands again to see how the information changes. The differences should be obvious and should make sense.

Step 2: the Volume Group

You don't currently have any volume groups, so create one using the two physical volumes you just made:

```
vgcreate my_volume_group /dev/sdb /dev/sdc
```

Hopefully the command is clear. You've created a volume group named my_volume_group using the physical volumes /dev/sdb and /dev/sdc. As with the physical volumes, if you want to check the current state of LVM Volume Groups on your system, type `vgdisplay` to get a listing:

```
$ sudo vgdisplay
  --- Volume group ---
  VG Name               my_volume_group
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               20.8 GiB
  PE Size               4.00 MiB
  Total PE              9988
  Alloc PE / Size       0 / 0 GiB
  Free  PE / Size       9988 / 20.8 GiB
  VG UUID               oVYiY6-bQp9-4CVO-QgrN-LGgB-1umR-ebJQo4
```

As you can see in the output, you've combined the available space of the two physical volumes (10.4GB each) into a total pool of 20.8GB. You could add more drives to the volume group or mix and match entire drives with partitions from other drives. LVM is very flexible. The large pool of available data does no good, however, until you create Logical Volumes to act as your usable disks.

Step 3: Logical Volumes

When you add a hard drive to your system, you don't really get to pick its name. You get /dev/sda, /dev/sdb and so on. When you create logical volumes, however, you decide what you want the devices to be called. You also get to decide how large each "drive" is as you carve it out of the larger volume group. It's good to note here that if you make your logical volumes too small, it's very easy to expand them later, so don't

worry too much about planning for long-term potential needs. If you need more space later, you can just add it. To create your logical volumes, type:

```
$ sudo lvcreate -L 5G -n 5gig my_volume_group
  Logical volume "5gig" created
```

Then to see what happened behind the scenes, type:

```
$ sudo lvdisplay
  --- Logical volume ---
  LV Path                /dev/my_volume_group/5gig
  LV Name                5gig
  VG Name                my_volume_group
  LV UUID                3MxOB0-ce5o-yvBD-YORT-52qV-j8HJ-oDru2G
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                5.0 GiB
  Current LE             5753
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           252:0
```

Notice how nice and clean the device-naming system is with LVM. It's important to run the `lvdisplay` command, however, to make sure you know the mapped device name. Many systems use symbolic linking in an attempt to make the device's virtual locations easier to find, but I think that adds a layer of confusion for folks trying to understand what's really going on.

Look, a New (Virtual) Hard Drive!

Once you've successfully created your logical volumes, it's just a matter of using them as block devices. If you need a filesystem to mount as your /home directory, just do this:

```
$ sudo mkfs.ext4 /dev/my_volume_group/5gig
$ sudo mount -t ext4 /dev/my_volume_group/5gig /home
```

And, your /home directory will be a whopping 5GB in size, but fully expandable, thanks to LVM. (Obviously, if you really want to mount your logical volume as your home directory, you should add an entry to /etc/fstab so it mounts on boot.) From the standpoint of your Linux system, however, /dev/my_volume/5gig is a block device similar to any hard drive you might plug in. You can use it as swap, format it like you did above, or even encrypt it and mount it somewhere as an encrypted partition.

That Was a Lot of Work, Why Again?

I know, in this little example, you've

done nothing but create a JBOD (Just a Bunch Of Disks) type system, which will completely fail if you lose even one drive. The power of LVM isn't fully realized until down the road when you want to expand your logical volumes without migrating data. Or, when you want to take an LVM snapshot of your drive so you can roll back to an instantaneous backup when an upgrade fails. Or, when you replace a small drive with a fast RAID array and want to migrate the data quietly to your new PV.

The Logical Volume Manager is a system that abstracts storage devices. It does add a layer of complexity to your system, I won't lie, but the trade-off is significant. It may complicate your system a bit more, but it also simplifies your work a great deal when you have to deal with storage in the future.

You Keep Talking about the Future...

Hopefully at this point, you see LVM isn't a complete waste of time. When the time comes, what sorts of advantages will LVM provide? Here's a quick off-the-top-of-my-head list you might want to check out:

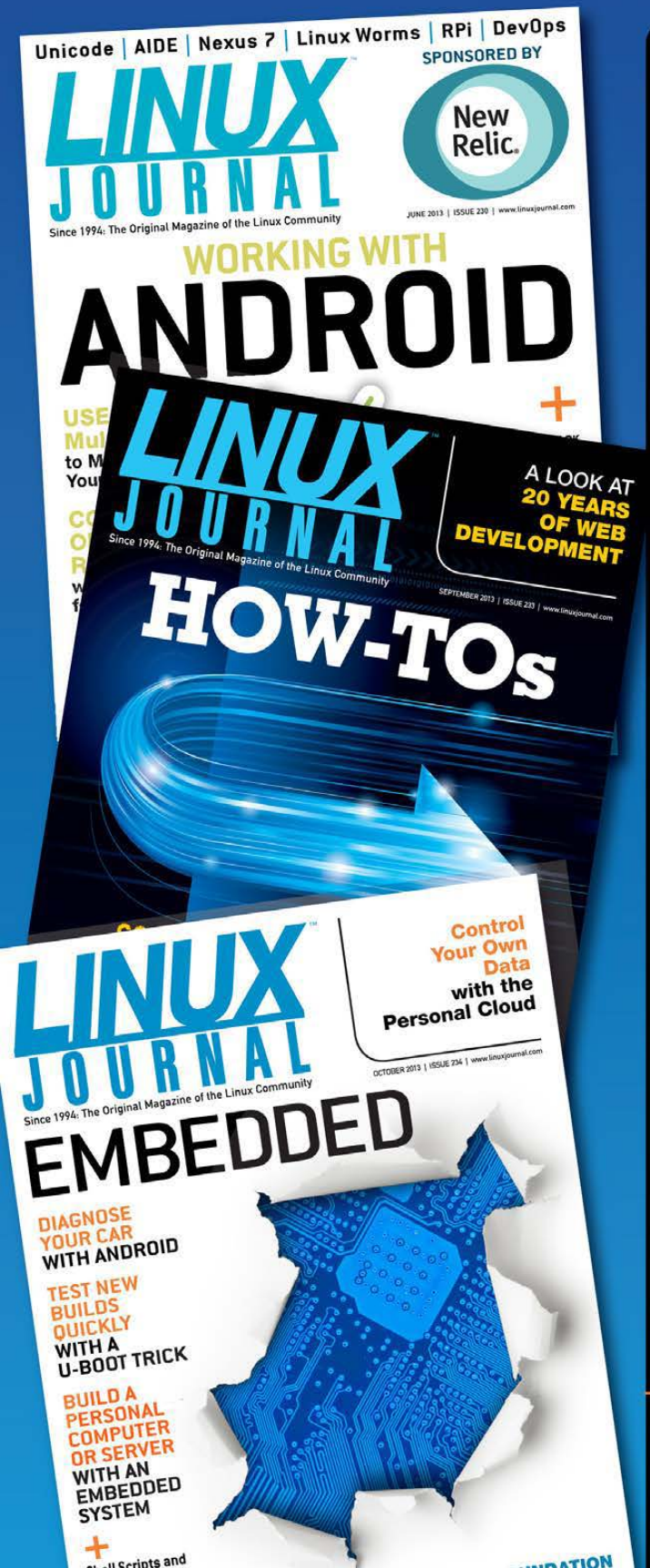
- Move Logical Volumes from old, slow PVs to new, fast PVs, on the fly.
- Resize Logical Volumes, filling more space in the Volume Group.
- Stripe data across PVs in a VG for increased performance.
- Resize Volume Groups by adding or subtracting physical volumes.
- Take a snapshot of any Logical Volume, which can be restored later.

One of my favorite uses for LVM in production is to take an LVM snapshot before an upgrade. If something goes wrong, I can just revert back to the snapshot. Once you start thinking about all the possibilities LVM offers, you'll wonder why you waited so long! ■

Shawn Powers is the Associate Editor for *Linux Journal*. He's also the Gadget Guy for LinuxJournal.com, and he has an interesting collection of vintage Garfield coffee mugs. Don't let his silly hairdo fool you, he's a pretty ordinary guy and can be reached via e-mail at shawn@linuxjournal.com. Or, swing by the #linuxjournal IRC channel on Freenode.net.

Send comments or feedback via http://www.linuxjournal.com/contact or to ljeditor@linuxjournal.com.

If You Use Linux, You Should Be Reading LINUX JOURNAL™



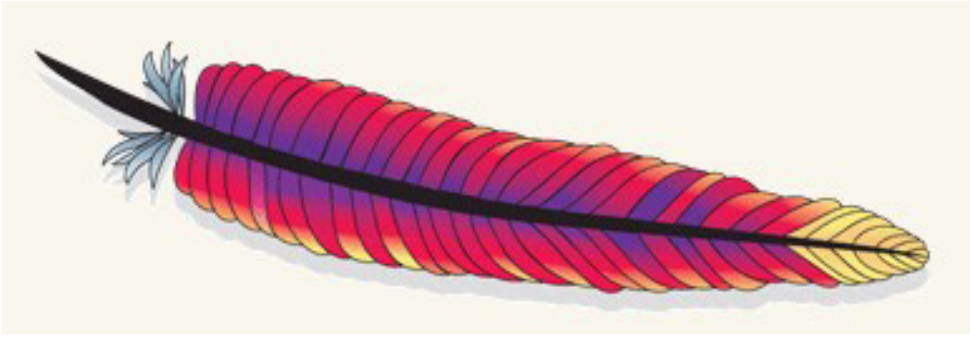
- In-depth information providing a full 360-degree look at featured topics relating to Linux
- Tools, tips and tricks you will use today as well as relevant information for the future
- Advice and inspiration for getting the most out of your Linux system
- Instructional how-tos will save you time and money

Subscribe now for instant access! For only $29.50 per year—less than $2.50 per issue—you'll have access to *Linux Journal* each month as a PDF, in ePub format, in Mobi format, on-line and through our Android and iOS apps. Wherever you go, *Linux Journal* goes with you.

SUBSCRIBE NOW AT:

WWW.LINUXJOURNAL.COM/SUBSCRIBE

Apache CloudStack



Apache CloudStack is an integrated Infrastructure-as-a-Service software platform that allows users to build feature-rich public and private cloud environments. Our contacts over at the Apache Software Foundation informed us that the new version 4.2, sporting 57 new and 29 improved features, and 160+ fixes, is available for immediate download. This release represents more than six months of work from the Apache CloudStack community. New integrated support of the Cisco UCS compute chassis, SolidFire storage arrays and the S3 storage protocol are just a few of the features available in this release. Both the official source code release and convenience binaries are available on the Apache CloudStack download page.

http://www.apache.org

Red Hat Enterprise Linux

Confirming its commitment to the Red Hat Enterprise Linux 5 platform's ten-year lifecycle, Red Hat ratcheted up RHEL's version number to 5.10, signaling a batch of noteworthy new features. With an emphasis on providing greater stability for critical applications, RHEL 5.10 offers enhanced features for reliability and security, including an updated version of OpenSCAP—the open-source Security Content Automation Protocol (SCAP) configuration scanner, which meets the National Institute of Standards and Technology's (NIST) SCAP 1.2 standard. Beyond OpenSCAP, the new release includes MySQL 5.5, enhanced subscription management tools, the new Red Hat Access support tools and Red Hat Developer Toolset 2.0.

http://www.redhat.com

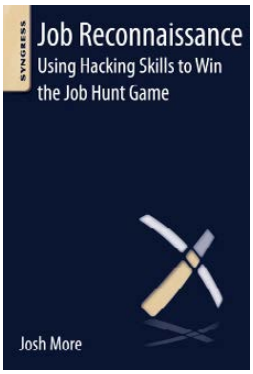
EnerPlex's Surfr Series Smartphone Cases

Convert your smartphone into a solar-electric hybrid model with the new solar-powered Surfr Series smartphone cases from EnerPlex. With a Surfr case on your Samsung Galaxy S III or Apple iPhone 4 or 4S, you can set yourself free from the outlet and enjoy hours of extra talk/text/surf time thanks to an on-board internal battery and a series of copper indium gallium selenide (CIGS) solar cells. The cases are thin (15mm thick for the Galaxy S III model) and light (69 grams), and their hardened plastic construction protects your smartphone from scratches and dings. A wall charger is included for indoor charging. EnerPlex products are the consumer side of Ascent Solar, a developer of thin-film CIGS solar technology.
http://www.enerplex.biz

Percona Server

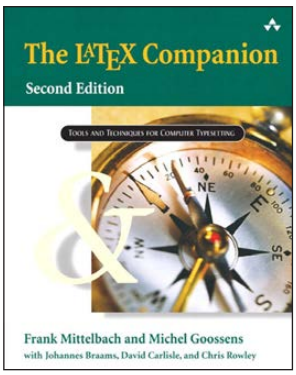
The value proposition of the new Percona Server 5.6—an open-source drop-in replacement for MySQL—says Percona, is that it includes much of the same functionality of MySQL 5.6 Enterprise Edition yet comes as a free download. The new 5.6 release, adds Percona, offers all the improvements found in MySQL 5.6 Community Edition plus scalability, availability, backup and security features found only in MySQL 5.6 Enterprise Edition, which requires an Oracle support contract to access. Superior diagnostics and improved integration with other Percona software also are included—for example, Percona Server 5.6 can be migrated to Percona XtraDB Cluster to create a high-availability MySQL cluster. Percona notes that by removing many resource contentions that slow MySQL 5.6 Community Edition, Percona Server 5.6 delivers better performance at more than 150 concurrent threads and up to 4x better performance at 1,000 or more concurrent threads. Percona's stated open "secret" to remaining more advanced and current than the latest release of MySQL, yet being drop-in-compatible, is its penchant for pushing out bug fixes and performance enhancements faster than Oracle.
http://www.percona.com

Josh More's *Job Reconnaissance* (Syngress)



Most people's limiting factor in getting a better job, says book publisher Syngress, is not technical skills or even the soft skills necessary to do well in a new job. But rather it is that getting a job is a completely different skill set and one that most people practice only periodically. Master job-search skills and get the job you deserve with Syngress' new Josh More book *Job Reconnaissance: Using Hacking Skills to Win the Job Hunt Game*. As the subtitle suggests, the book seeks to inform infosec and IT job seekers about leveraging the same skills they use in penetration testing and recon toward job-hunting success. These skills include targeting, reconnaissance and profiling combined with a technical look at skills other career search books commonly miss. The book covers the entire job-hunt process from deciding when to leave your current job to the departure of your current job, suggests how to research new possible job opportunities and illustrates how to target your new boss, from controlling the job interview process to negotiating compensation.

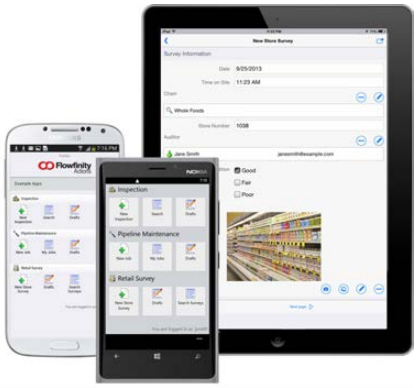
http://store.elsevier.com/Syngress



Frank Mittelbach and Michel Goossens' *LaTeX Companion* (with Johannes Braams, David Carlisle and Chris Rowley) 2nd ed. (Addison-Wesley Professional)

It was 1993 when the first edition of Frank Mittelbach et al.'s *LaTeX Companion* came out. The new second edition of *LaTeX Companion: Tools and Techniques for Computer Typesetting* recently was released and is part of a new boomlet of LaTeX book titles. A technology with impressive staying power, LaTeX is a high-quality typesetting system used in the publication of scientific documents. This completely updated edition contains the latest information about LaTeX and the vast range of add-on packages now available, with more than 200 of them treated therein. Full of new tips and tricks for using LaTeX in both traditional and modern typesetting, this book also illustrates how to customize layout features, from phrases and paragraphs to headings, lists and pages. Nearly 1,000 fully tested, ready-to-run examples that illustrate the text and solve typographical and technical problems are included. The accompanying CD-ROM contains a complete plug-and-play LaTeX installation, including all the packages and examples featured in the book.

http://www.informit.com



Flowfinity Wireless Inc.'s Flowfinity Actions

Empowering employees to be more productive everywhere, whether in or out of network coverage, is Flowfinity Wireless Inc.'s "mission possible" with Flowfinity Actions. Flowfinity Actions 7.5 is the latest version of the firm's flexible application software that supports cross-platform enterprise app creation without programming. The premier improvement in this latest release is the ability for users to save current progress and switch between multiple Flowfinity apps with or without network connectivity. This is especially useful when performing data collection tasks, such as pipeline inspections, agricultural management and remote construction site reporting, which require diverse information to be gathered or accessed when no Internet connection is available. Another new off-line feature is parameter or filter-based search, enabling users to filter through thousands of records quickly to find the information they need at any time. Finally, lookups that allow information to be stored in one app and copied into another app also are fully supported off-line.
http://www.flowfinity.com

ASUS RT-AC68U Wireless Router

Boasting dual-core CPU and TurboQAM technology, the new ASUS RT-AC68U Dual-band Wireless-AC1900 Gigabit Router is touted by its maker as the world's fastest dual-band wireless router. The technological innovations allow the RT-AC68U to deliver speeds up to 1.3Gbps over 802.11ac and 600Mbps over 802.11n. The RT-AC68U further features dual USB ports for file, printer and 3G/4G modem sharing that includes a USB 3.0 port for high-speed data transfers. ASUS AiCloud technology transforms home networking into one's personal cloud for easy streaming and sharing to smartphones, PCs and tablets. AiRadar intelligently strengthens the signal for extended and enhanced coverage while dual-band 2.4GHz and 5GHz technology ensure full backward-compatibility with current wireless devices.
http://www.asus.com/Networking/RTAC68U

Please send information about releases of Linux-related products to newproducts@linuxjournal.com or New Products c/o *Linux Journal*, PO Box 980985, Houston, TX 77098. Submissions are edited for length and content.

READERS' CHOICE AWARDS 2013

This year's awards feature reader-nominated categories and choices. See who won!

SHAWN POWERS

This year's Reader's Choice issue was truly fun to put together. No, not just because you do all the work (voting), but because it's great to get a feel for what our community is buzzing about. Based on your feedback, we've given you all the data again this year, with percentages and rankings, plus we tried to include as many of your less-popular responses as possible. It wasn't that long ago Linux itself was less popular, so we have a soft spot for such things.

We also had an extra round of voting this year specifically for nominations. Everything you see below is reader-generated, including some new categories suggested by readers. Well, okay, my comments aren't reader-generated, but hey, I do read every issue, so that counts, right? We hope you enjoy this year's Readers' Choice Awards.

BEST LINUX DISTRIBUTION

This year's Best Linux Distribution is a testament to choice. Although it's not a surprise that Ubuntu is in the top spot, it's fun to see how close those top spots are to each other. Just a handful of percentage points separates a dozen or so distributions, and the most popular option only has 16% of the total votes! When it comes to their distros, Linux users love choice.

16% Ubuntu

14.1% Debian
10.8% Arch Linux
10.5% Linux Mint
6.9% Fedora
5.2% openSUSE
4.7% SolydK
4.1% CentOS
3.8% Kubuntu
3.7% PCLinuxOS
3.2% Gentoo
2.1% Other
2.1% Slackware
2% elementary OS
2% Xubuntu
1.8% Manjaro
1.6% Red Hat Enterprise Linux
1.6% Ubuntu Server
1.1% CrunchBang
.8% Mageia
.6% Chakra
.5% Lubuntu
.2% Puppy
.1% Bodhi Linux
.1% Kali Linux
.1% Mandriva
.1% NixOS
.1% Oracle Linux
.1% SolusOS
.1% Zorin OS

BEST DISTRIBUTION FOR NETBOOKS/LIMITED HARDWARE

What used to be a category full of very specific distributions designed for very specific screen sizes has become far more generic. That doesn't mean small screens aren't loved, it just means that standard Linux distributions consider the smaller screens in their design philosophies. Seeing Debian at the top is pretty cool, because it's more often seen as the "foundation" for another spin-off. It's also great to see Android on the list, as a large number of smaller devices are indeed tablets or their ilk.

13.1% Debian

12% Arch Linux
10.6% Ubuntu/Unity
10.4% Android
10.1% Other*
8.2% Xubuntu
6.5% Lubuntu
5.9% Puppy
5.5% Chrome OS
5.1% Fedora
3.3% PCLinuxOS
2.6% Gentoo
2.2% Slackware
2.1% Manjaro
1.3% Peppermint OS
1% LXLE
.1% Wolvix

**Popular write-ins: CrunchBang, SolydXK, Bodhi Linux, Linux Mint and elementary OS.*

BEST DISTRIBUTION FOR HIGH-PERFORMANCE COMPUTING

When it comes to raw number-crunching, we need a distribution that is easy to manage, easy to scale and easy to trust. Stability and predictability trump all the glitz and glamor of the desktop-focused distributions. Debian and Ubuntu top the results this year, with RHEL and CentOS right behind. These are the names we've come to trust, and with high-performance computing, that's what we need.

22.5% Debian

15% Ubuntu
13% Red Hat Enterprise Linux
11.7% CentOS
8.6% Other*
7.3% Gentoo
7.2% Linux Mint
5.4% openSUSE
5% Fedora
3.3% SUSE Linux Enterprise Server
1.1% Rocks Cluster

**Popular write-ins: Arch Linux and PCLinuxOS.*

BEST DESKTOP DISTRIBUTION

It's no surprise to see Ubuntu in the top spot for Desktop Distributions. It took all around favorite distro, and because it excels on the desktop, this makes sense. The results show that Arch Linux is really turning some heads as a viable alternative to the "traditional" desktop system. Linux Mint, designed with the philosophy of making things easy for the end user, logically grabs a huge percentage as well.

23.2% Ubuntu

16% Linux Mint
8.7% Arch Linux
8.6% Fedora
8.1% Debian
6.1% openSUSE
5.6% Kubuntu
4.9% SolydK
3.9% PCLinuxOS
3.8% Other*
2.9% Xubuntu
2.4% Gentoo
2% Manjaro
1.7% Slackware
.8% Chakra
.6% Red Hat
.4% Lubuntu
.3% Xfce
.1% NixOS

**Popular write-ins: elementary OS and CentOS.*

BEST DESKTOP ENVIRONMENT

Oh GNOME…the Best Desktop Environment category will never quite be the same. KDE easily takes top spot this year over Unity, and GNOME (in any of its forms) is down below. Even XFCE, one of my personal favorites, comes in ahead of GNOME. That's not to say the GNOME way of doing things is gone. Several other alternatives with some decent percentages of the votes are re-creating the old GNOME concept. If "GNOME-like" were an option, we might see closer numbers, but as it is, KDE is the king.

17.9% KDE

12.9% Ubuntu/Unity
12.7% KDE Plasma
12.1% Xfce
14.1% GNOME 3
8.6% Cinnamon
4.5% GNOME 2
4.5% Other*
3.6% MATE
2.6% Openbox
1.7% Enlightenment
1.7% LXDE
1.3% PCLinuxOS
.9% Fluxbox
.7% DWM
.1% KWin

**Popular write-ins: awesome window manager, i3 window manager and Pantheon desktop environment.*

BEST MOBILE LINUX OS

It came as a shock to approximately zero people that Android is the most popular mobile OS. The cool part of the survey is that alternatives are available, and they got a not-insignificant number of votes. Yes, CyanogenMod is Android, but it was different enough (and got enough votes) that we thought separating it out was interesting. Plus, it's not like Android needed those votes to win! Between them, Sailfish and FirefoxOS took more than 20% of the vote, which both surprised and excited us. The mobile world is where the action currently is happening, and it's great to see the options haven't stagnated.

46.6% Android

17.5%	Sailfish OS	2%	Other	.9%	Mer
14.2%	CyanogenMod	1.5%	Maemo	.9%	Replicant
6.3%	FirefoxOS	1.4%	Ubuntu for Phones	.7%	Tizen
2.6%	MeeGo	1%	Ubuntu for Android	.6%	Gentoo
2.6%	Ubuntu Touch	1.3%	PCLinuxOS		

BEST LINUX SMARTPHONE MANUFACTURER

As someone with a Samsung phone in his pocket, it's not a surprise to see the Korean company take more than a third of the votes. Samsung has created some truly beautiful hardware of late, and it doesn't look like it plans to stop. Will the new wave of smartwatch accessories cement Samsung's lead? Time will tell, but for now, it's certainly your favorite. Jolla took a surprisingly large number of votes this year, especially considering the lack of large numbers of units. Still, being small has never bothered the Linux community before. If Jolla makes great hardware, we'll let it know with our pocketbooks.

34.4% Samsung

20.1%	Jolla	3.4%	Other	1.4%	Huawei
14%	Nexus	3.1%	Sony	1%	Golden Delicious Computers
12.7%	HTC	2.8%	GeeksPhone		
4%	Nokia	2.6%	LG	.3%	Winko Cink

BEST LINUX TABLET

Google takes the top two spots this year, with the same tablet in different sizes. The Nexus 7 received a massive upgrade this year, so those votes are likely split between the two models, but more than half our votes went to a Google-branded tablet. I suspect part of their success is the early access to Android upgrades, but there's a lot to be said for delivering the stock system and not adding custom interfaces. The specialized Kindle and Nook tablets are near the bottom of the list, probably due to most of us being power users and wanting the most out of our tablets.

37.9% Google Nexus 7

15.4% Google Nexus 10
11% Samsung Galaxy Tab
9.3% KDE Vivaldi Tablet
8.7% Samsung Galaxy Note
6.4% Ubuntu Edge
5.8% Other*
3.3% Kindle Fire HD
1% Nook HD
.7% PengPot
.6% Ekoore Python S3

**Popular write-in: Asus Transformer.*

BEST OTHER LINUX-BASED GADGET

If its "Reign of Awesome" continues much longer, we're going to have to rename this category "Best Linux Gadget That Isn't a Raspberry Pi". The tiny little ARM system blows everything else out of the water again this year. And really, no one is surprised. Heck, Kyle Rankin and I are still writing about the various uses we have for our Raspberry Pi units, and we were doing that more than a year ago! The Google Chromecast took a fair number of non-RPi votes in this category, making it clear that if you haven't given the Chromecast a try, perhaps you should.

68.3% Raspberry Pi

5.6% Google Chromecast
5.4% Amazon Kindle DX
3.3% BeagleBone/BeagleBoard
3.1% Roku 3
2.9% mintBox
2.6% Other
2.5% TomTom Navigation System
2.2% Parallella Supercomputer
1.4% GTA04 Upgrade Board for Openmoko
1.1% Cubieboard
1% Sony PRS T2 eReader
.3% RaZberry
.2% Gumstix

BEST LINUX LAPTOP VENDOR

With just a fraction of a percentage point, the Linux-specializing System76 takes the top spot away from Lenovo this year. It says a lot about Lenovo, however, that a company who doesn't specialize in Linux is almost tied in popularity to one that does. There's a lot of big names near the top of the list this year, which is good news for everyone, because it means more and more laptops are Linux-friendly.

25.6% System76

25.3% Lenovo
17.2% ASUS
15.4% Dell
5.9% Acer
5.6% Other*
2.6% EmperorLinux
2.2% LinuxCertified
.2% CyberPower
.1% Eurocom

**Popular write-in: ZaReason.*

BEST LINUX-FRIENDLY HARDWARE VENDOR

Not satisfied with simply taking Top Gadget, the Raspberry Pi Foundation takes the second-place spot as a hardware vendor this year behind Intel. It seems like Intel has made huge strides in efficiency and speed this past year, and it hasn't left Linux users out along the way. The other big names in computer hardware ranked well on the list too, which again shows that the big companies continue to take Linux seriously.

25% Intel

19% Raspberry Pi Foundation
12.7% System76
9.5% AMD
8.1% Lenovo
7.3% Dell
4.7% NVIDIA
4.5% Hewlett-Packard
4.1% IBM
2.3% Other
1.1% Supermicro
.7% Element14
.3% Microway
.3% MSi
.2% Eurocom
.2% Huawei
.2% Silicon Mechanics

BEST LINUX DESKTOP WORKSTATION VENDOR

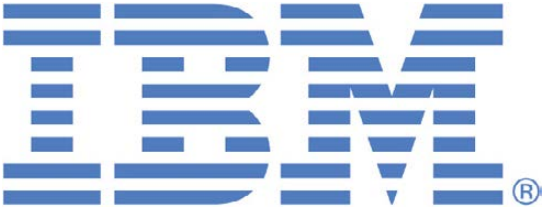
It's nice to see various companies excel at certain things. Dell, for instance, takes our top spot as Desktop Workstation Vendor, while it didn't make top three for laptops. This is one of the reasons I love the Readers' Choice Awards so much, because without purchasing lots of computers from lots of companies, I'd never know who is the best. You had some strong opinions on desktop computers, and the votes were split between only a handful of vendors—valuable information for any potential buyer.

37% Dell

35.1% System76
16.7% Hewlett-Packard
6.9 Other*
2.2% CyberPower
2.1% Microway

**Popular write-ins: "build your own" or "custom made".*

BEST LINUX SERVER VENDOR



Server hardware, much like our High-Performance Computing category, specifically targets trustworthy, rock-solid computers. Even making our lists implies total Linux compatibility as well. IBM takes the top spot in a very competitive category. It's great to see big companies on our list, proof that Linux is serious business.

33.4% IBM

31.1% Dell
18.9% Hewlett-Packard
8.7% Supermicro
6.1% Other*
1.9% Microway

**Popular write-ins: System76 and "build your own".*

BEST ANDROID APP

Google Maps is indeed an incredible application on Android. I was personally surprised to see the Firefox Mobile browser take second place, because it's not a Google project. Don't get me wrong. I was very pleasantly surprised. If you're looking for a list of great apps to put on your phone or tablet, look no further than these results.

15.5% Google Maps

13.1%	Firefox Browser for Android
10.3%	Other*
8.4%	Chrome Browser for Android
7.4%	Dropbox
5%	K-9 Mail
3.9%	F-Droid
3.8%	SwiftKey
3.7%	Waze
3.5%	ConnectBot
3.4%	Dolphin
2.9%	Linux Journal
2.3%	Aldiko
2.3%	LibreOffice Impress Remote
2.2%	Kingsoft Office
2.2%	Nova Launcher
1.8%	Plants vs. Zombies
1.3%	Duolingo
1.1%	TTRSS-Reader
1%	Google Currents
1%	Moon+ Reader
1%	Sudoku
.7%	CamScanner
.5%	My Tracks
.5%	Out of Milk
.4%	PC Monitor
.3%	ElectroDroid
.3%	Subsonic
.2%	TouchDown
.2%	Tweet Lanes
.1%	imo

**Popular write-ins: AirDroid, whatsapp, OsmAnd and Opera Mobile.*

BEST CONTENT MANAGEMENT SYSTEM

Content management systems are doing their best when they get out of our way and let us publish content. WordPress claims victory this year as your favorite CMS, and it's followed by Drupal, Joomla! and MediaWiki. If you're looking to try something new, our results show a handful of lesser known, but obviously still worthwhile projects to check out. If you're looking for proven results, however, it's hard to beat WordPress.

35.6% WordPress

20.6%	Drupal
14.3%	Joomla!
10.1%	MediaWiki
9.3%	Other*
3.7%	MATE
3.1%	Alfresco
1.1%	WebGUI
.9%	ikiwiki
.5%	Blosxom
.3%	eZ publish
.2%	Conary
.2%	Elgg
.1%	Wolf CMS

**Popular write-ins: Django, Plone and dokuwiki.*

BEST LINUX-FRIENDLY WEB HOSTING COMPANY

I found it interesting that Amazon took the number-one place in our survey this year for Linux-Friendly Web Hosting Company. Don't get me wrong, Amazon is definitely Linux-friendly, but I've never considered it a standard Web hosting company. Still, with the incredible array of Web- and cloud-based tools available, I guess it makes sense it would take the gold medal. And really, if you haven't used Amazon's services, you should!

20.2% Amazon

16.6% Other*
12.1% GoDaddy.com
11.3% Rackspace
10.7% Linode
5.2% OVH
4.8% DreamHost
4.5% 1&1
4.5% LAMP Host
4.4% HostGator
1.7% Hurricane Electric
1.1% Liquid Web
.9% Sawis
.7% Blacknight Solutions
.4% Netfirms
.4% Prgmr
.4% RimuHosting
.2% Host Media

**Popular write-ins: Digital Ocean, Hetzner, Blue Host, Bytemark and Gandi.*

BEST WEB BROWSER

Wow! Way to go Firefox! Beating Google's Chrome browser in both the mobile app category and this one, Firefox is proving it's not going away any time soon! More than half of you picked Firefox as best browser, so if you're one of the 47% who didn't, perhaps it's time to revisit the trusty old Firefox, you might be surprised!

52.8% Firefox

35.5% Chrome/Chromium
4% Opera
3.1% Iceweasel
1.5% Other
1.3% rekonq
.7% QupZilla
.6% SeaMonkey
.4% dwb
.1% Dillo

BEST RSS READER

The RSS category was difficult for many of us this year. Google Reader going away created a huge vacuum in the newsreader world. Thankfully, there were many alternatives that arrived to fill the void. Feedly, with its unique interface, took your top spot, followed by the more traditional RSS readers, Thunderbird and Akregator.

24.9% Feedly

19.6%	Thunderbird	5.7%	gReader	**Popular write-ins: Digg Reader, NewsBlur, ownCloud news, rssOwl and this old reader.*
14.9%	Akregator	3.7%	SeaMonkey	
10.4%	Other*	1.8%	Newsbeuter	
9.9%	Tiny Tiny RSS	1.5%	InoReader	
6.9%	Liferea	.6%	Miniflux	

BEST BOOKMARK-SYNCING TOOL

Although it might be due to Firefox's popularity with readers, perhaps Firefox's syncing ability is what *gave* Firefox its edge this year. Chrome certainly got a lot of votes for Best Bookmark-Syncing Tool, but Firefox once again takes the top spot. Great job, Firefox team!

36.4% Firefox Sync

32.5%	Chrome/Chromium	4.8%	Other*	1.6% Tiny Tiny RSS
9.5%	Xmarks	3.8%	Delicious	.2% SemanticScuttle
9.1%	Google Sync	2.1%	Feedly	**Popular write-in: Opera Link.*

BEST E-MAIL CLIENT

The thought that 5% of our readers are like Kyle Rankin and prefer their e-mail in a terminal window is both scary and fascinating. For most of us, GUI e-mail is where it's at, and Thunderbird grabbed almost half the votes as Best E-mail Client. Granted, the Web-based Gmail came in with 29% of the vote, but it's impressive to see how many people still prefer an actual e-mail program versus a Web site.

41.1% Mozilla Thunderbird

29%	Gmail	2.8%	Geary	.7%	SeaMonkey
7.7%	KMail	1.8%	Claws Mail	.6%	Gnus
4.8%	Mutt	1.3%	Opera Mail	.5%	Sylpheed
4.2%	Evolution	1.2%	Alpine	.4%	IBM Notes 9
3%	Other	1.1%	Zimbra	.1%	Rediffmail

BEST IM CLIENT

Pidgin is still top—well, top bird I guess, in our IM category. It's one of my personal favorite ways to handle IRC, so it doesn't surprise me to see Pidgin on top again. When it comes to sheer number of supported protocols, Pidgin is amazing. Skype takes a surprising second position, not surprising due to lack of ability, but surprising because it's owned by Microsoft! Thankfully, Skype is still available for Linux, so we won't sling mud.

39.5% Pidgin

12.9%	Skype	2.4%	Jitsi
11.9%	Google Chat	2.2%	Konversation
7.7%	Empathy	1.3%	Psi
7.2%	Kopete	1.1%	Gajim
5.6%	Other*	1.1%	Miranda IM
4.6%	Facebook		
2.6%	BitlBee		

**Popular write-ins: KDE Telepathy, Google Hangouts and Irssi.*

BEST IRC CLIENT

Back in my day, we didn't need audio or video to chat with our friends. We had green text on a black screen, and we liked it. Thankfully, for many of us, we're still living in my day. XChat takes top spot as the IRC-specific application, and Pidgin follows close on its heels offering incredible IRC support for a multiprotocol client.

24.6% XChat

21.6% Pidgin
15.9% Irssi
9.5% Konversation
7.3% Chatzilla
4.9% Other*
4.7% Quassel
3.7% WeeChat
2.6% HexChat
2.6% Opera IRC
1.4% Jitsi
1.2% ERC (emacs)

**Popular write-in: KVirc.*

BEST MICROBLOGGING CLIENT

Those of you who know me know that I tweet a lot—probably more than my employers like! I've tried just about every Twitter client available, and I agree with the vote here. TweetDeck is my number-one client as well. Gwibber comes in a close second, but for me, it's the centralized login that works between computers that tips the scale.

31.7% TweetDeck

30.9% Gwibber
14.3% Choqok
13.7% Other
6.6% Hotot
2.8% bti

BEST OFFICE SUITE

Remember when there was only one main office suite Linux users loved? Oh, right, that's now! LibreOffice grabbed almost 75% of the votes this year, and keeps its spot as your favorite office application.

71.8% LibreOffice

11.8% Google Drive
6.7% Apache OpenOffice
5.2% Calligra Suite
4.2% Other*
.3% WebODF

**Popular write-in: Kingsoft Office.*

BEST SINGLE OFFICE PROGRAM

49.4% LibreOffice Writer

15.7% LibreOffice Calc
9.4% AbiWord
4.9% Gnumeric
4.8% Apache OpenOffice Writer
4.6% Other
4.1% Calligra Words
1.7% SciTE
1.6% AUCTex
1.5% Apache OpenOffice Calc
.8% Calligra Plan
.8% Calligra Stage
.6% Apache OpenOffice Impress

BEST GRAPHICS/DESIGN TOOL

When it comes to editing photos, Linux has so many options. GIMP is your number-one choice for image manipulation, and with the new single-window option, it's even easier for new folks to use. Inkscape and Blender also are high on your list of graphics/design options, so if the GIMP doesn't quite do it for you, check them out.

58.7% GIMP

15.1% Inkscape
13.1% Blender
6.3% Krita
1.9% Other
1.1% Pinta
1% Apache OpenOffice Draw
.9% Tux Paint
.8% RRDtool
.6% Xfig
.5% DraftSite

BEST DIGITAL PHOTO MANAGEMENT TOOL

I'll be honest, seeing the GIMP on the top of the photo management list did surprise me a bit. I still haven't found what I consider the perfect photo management tool, but thankfully you've provided a few I haven't looked into. Now if you'll excuse me, I have some tools to try.

28.5% GIMP

19.1% digiKam
14.3% Shotwell
9.5% Picasa
6.7% Gwenview
5.7% darktable
5.4% ImageMagick
3.9% gThumb
2.7% Other
1.3% Bibble/Corel AfterShot Pro
1.3% Geeqie
.8% LightZone
.8% RawTherapee

BEST AUDIO TOOL

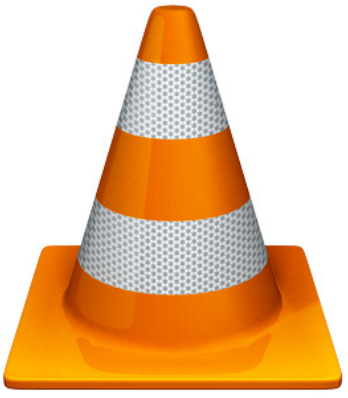
What do you do with audio? You create, convert and listen. Those three needs define the top three spots in our survey. Audacity takes top spot, as an incredible audio creation and editing tool. Then VLC converts and plays, while Amarok is an incredible playback tool. Linux people know their audio, and we have the tools to prove it.

34.7% Audacity

21.1%	VLC	4.3%	Other	.9%	Mixxx
12.2%	Amarok	4.1%	Ardour	.7%	LMMS
10.6%	FFmpeg	4.1%	XBMC	.2%	Format Junkie
6.1%	Audacious	1%	SoX		

BEST AUDIO PLAYER

Much like the "Top Audio Tool" category showed, VLC and Amarok are incredible audio players. This category includes a plethora of other options, in case the top slots don't fit your needs or desires.



21.7% VLC

18.2%	Amarok	2.8%	foobar2000	.6%	MPC-HC
11.3%	Clementine	2.3%	XBMC	.5%	cmus
9.6%	Rhythmbox	1.7%	Xmms	.5%	Mixxx
5.7%	Audacious	1.6%	DeaDBeeF	.5%	Subsonic
5.7%	Banshee	1.3%	Ncmpcpp	.2%	Decibel Audio Player
5.1%	Other	1%	MOC		
4.3%	Spotify	.9%	Nightingale		
3.9%	MPlayer	.7%	Guayadeque		

BEST MEDIA PLAYER

60.3% VLC

13.1%	MPlayer	2.4%	Clementine	.6%	Xmms
8.4%	XBMC	1.5%	Kaffeine	.3%	Daum Potplayer
3.8%	Amarok	1.4%	Plex		
3.4%	Other*	1.2%	MythTV		
3%	Totem	.8%	mpv		

*Popular write-in: SMPlayer.

BEST VIDEO EDITOR

Kdenlive has come a long way through the years, and your votes are a testament to just how awesome it's become. It's interesting that VLC took the second spot; apparently we don't all think "non-linear editor" when we think of video editing.

21% Kdenlive

20.9%	VLC	16.4%	Avidemux	6.1%	PiTiVi
19.4%	OpenShot	11.3%	LightWorks	4.9%	Other

BEST ON-LINE COLLABORATION TOOL

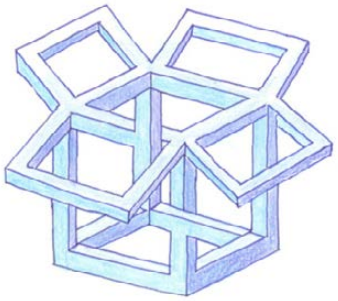
Whether you love or hate Google, it's hard to deny it's really done an amazing job with document collaboration. Multiple people editing the same file at the same time is... well, you have to see it to believe it. Of course, not all collaboration is editing a document, and your votes show there are other ways to collaborate with Linux.

50.5% Google Docs

16.3%	Google Hangout	5.7%	Other*	1.4%	Feng Office
10.1%	MediaWiki	5.3%	Redmine	.1%	Norton Zone
8.8%	WordPress	1.7%	WebODF		

*Popular write-in: Etherpad.

BEST CLOUD-BASED FILE STORAGE



Dropbox is still the favorite cloud-based storage option, but it's great to see ownCloud nipping at its heels. No, not because I have anything against Dropbox (I use it myself), but because I love to see open-source alternatives whenever possible.

35.5% Dropbox

16.3% ownCloud
16.2% Google Drive
7.1% Ubuntu One
6.8% rsync
5.2% Other*
4.8% Amazon S3
4.6% SpiderOak
1.7% Copy
1.4% Box
.4% AjaXplorer
.1% Norton Zone

**Popular write-ins: BitTorrent Sync, MEGA, Skydrive and Wuala.*

BEST LINUX GAME

Okay, true confession, it's been a couple years now, and I still don't understand the popularity of *Minecraft*. I just don't get it. That's okay, however, because most of you obviously do! *Minecraft* takes more than 20% of the vote this year. Will Steam's Linux support change things up next year? We'll have to wait and see.

21.2% *Minecraft*

20.9% Other*
18.5% *Half-Life*
11.1% *Frozen Bubble*
7.1% *Trine 2*
5.2% *OpenTTD*
4.2% *Battle Field*
3.3% *Warzone 2100*
1.8% *FreeOrion*
1.6% *Hedgewars*
1.4% *Scorched 3D*
1.2% *Darwinia*
.9% *KGoldrunner*
.8% *Glest*
.6% *Oolite*
.4% *BurgerSpace*

**Popular write-ins:* 0 AD, Battle for Wesnoth, Dota 2 *and* FTL: Faster Than Light.

BEST BRAND OF VIDEO CHIPSET

51.6% NVIDIA

26.3% Intel
21.3% AMD
.8% Other

BEST SQL DATABASE

Databases may not be the most exciting topic of discussion, but as someone who spent the past year working in the database department of a university, I can assure you, they are important. The numbers are very close this year, but it's neat to see MariaDB topple PostgreSQL.

29.7% MySQL

28.5% MariaDB
26.8% PostgreSQL
8.8% SQLite
4.4% Oracle
1.8% Other

BEST NOSQL DATABASE

43.6% MongoDB

15.3% Apache HBase
13.2% Cassandra
13.2% CouchDB
7.7% Other
4.5% Redis RethinkDB
2.6% Neo4j

BEST BACKUP SOLUTION

The short version of our results: I don't care which option you use, just back up! Now! Seriously though, it's interesting to see Dropbox as a backup solution. Yes, it does versioning, but I guess I've never considered it a backup. Perhaps I'm too old.

19.6% Clonezilla

19.3% Dropbox
19.1% Other*
8.8% Bacula
7.7% rdiff-backup
7.5% CrashPlan
5.4% Back In Time
4.5% Amanda
4.1% luckyBackup
1.8% Tivoli Storage Manager
1.4% Symantec Backup Exec
.5% Storix
.4% Areca-Backup

**Popular write-ins: BackupPC, Deja Dup, SpiderOak, duplicity and rsync + tar/btrfs/ftp/cron/and so on.*

BEST VIRTUALIZATION SOLUTION

37% Oracle VM VirtualBox

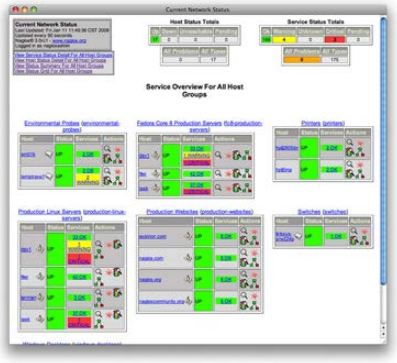
22.2% KVM
21.2% VMware
4.9% XEN
4.8% OpenStack
4.1% QEMU
2.7% Other*
1.7% OpenVZ
1.3% Linux-VServer
.2% Symantec Workspace Virtualization

**Popular write-in: LXC.*

BEST MONITORING APPLICATION

21.9% Nagios

20% Wireshark
17.4% htop
7.7% Other*
7.1% Zabbix
5.1% Zenoss
4.8% PC Monitor
3.4% Munin
2.9% New Relic
2.7% SaltStack
1.9% Monit
1.8% NTM (Network Traffic Monitor)
.8% FlowViewer
.8% Opsview
.6% Manage Engine
.6% SysPeek
.4% Circonus
.3% xosview



**Popular write-in: Icinga.*

BEST OPEN-SOURCE CONFIGURATION MANAGEMENT TOOL

Our results here are admittedly a little different than we expected. Although I love apt as much as the next guy (moo), I never considered it a configuration management tool. Whether it was meant as a non-answer (we don't need no stinking configuration management!) or a desire to include configuration into deployable packages, apt was an unexpected winner. It also was cool to see Subversion, as I have heard of people using it for managing configuration revisions, and it's neat to see evidence of that reflected here.

38.8% apt

21.1% Puppet
11.6% Yast
9.6% Other*
9.2% Subversion
6.6% SaltStack
3.1% CFEngine

**Popular write-ins: Ansible, Chef and git.*

BEST PACKAGE MANAGEMENT APPLICATION

While apt-get and Synaptic are both tools for managing the apt system, pacman is its own beast entirely. More than 50% of you chose either apt-get or Synaptic, but that 13% vote for pacman is proof that Arch Linux is popular.

38.5% apt-get

13.7% Synaptic
12.9% pacman
11.1% RPM
6.9% Aptitude
6.2% Other*
4.3% Gentoo Portage
1.8% Yaourt
1.5% dpkg
1.4% SaltStack
1.1% pkgtool
.5% Nix
.3% Conary

**Popular write-ins: yum, yast and zypper.*

BEST REVISION CONTROL SYSTEM

78.3% Git

11.8% Subversion
4.9% Mercurial
2.6% Bazaar
1.8% Other
.7% Plastic SCM

BEST DEBUGGER (SERIAL OR PARALLEL)

75.8% GDB (The Gnu Project Debugger)

13.9% PuDB (Python Debugger)
6.5% Other
3.8% Edebug

BEST OPEN-SOURCE SECURITY TOOL

23.2% Nmap

17.3% BackTrack Linux
15% Kali Linux
14.6% KeePass
7.8% Netfilter
6.9% pfSense
6.5% Metasploit
5% Other
2.3% AIDE
1.5% OSSEC

BEST OPEN-SOURCE FORENSICS TOOL

This one obviously was answered by folks who have done forensics work. All the fancy tools in the world are the second step to the venerable dd.

29.2% dd

27.7% Kali Linux
15.2% TestDisk
11.7% HexEdit
10.4% Sleuth Kit
5.8% Other

BEST OPEN-SOURCE PEN TESTING TOOL

37.1% Nmap

32.3% Kali Linux
25.1% Metasploit
5.6% Other

BEST FILE ENCRYPTION

34.1% TrueCrypt

29.4% GnuPG
14.9% LUKS (Linux Unified Key Setup)
6.9% dm-crypt
5.5% eCryptfs
4.3% EncFS
3.1% Other
.9% Gringotts
.9% Symantec Endpoint Encryption Full Disk Edition

BEST PROGRAMMING LANGUAGE

It's interesting that Python takes the spot as best programming language—not just "best beginner's language", but best language. I still recommend people start with Python, but based on this category, perhaps they never have to leave.

28.3% Python

21.3% C++
14.9% C
7.4% Java
5.3% Other
4.9% Perl
4.4% Ruby
4.3% JavaScript
3.6% QML
1.8% Go
1.2% Haskell
1% Lisp
.7% Rust
.6% Fortran
.4% Erlang

BEST SCRIPTING LANGUAGE

37.1% Python

20.2% Bash
10.8% JavaScript
10.5% PHP
8.6% Perl
4.9% Ruby
4% Shell Script
2.2% Other
1.7% Lua

BEST TEXT EDITOR

Yes! My fellow nerds united and voted vi/vim the best text editor. My challenge to you is to explain vim to a high-school student who grew up with Microsoft Word. That's a tough crowd. Still, as my personal go-to for all text editing, it's nice to see vim get the love.

38% vi/vim

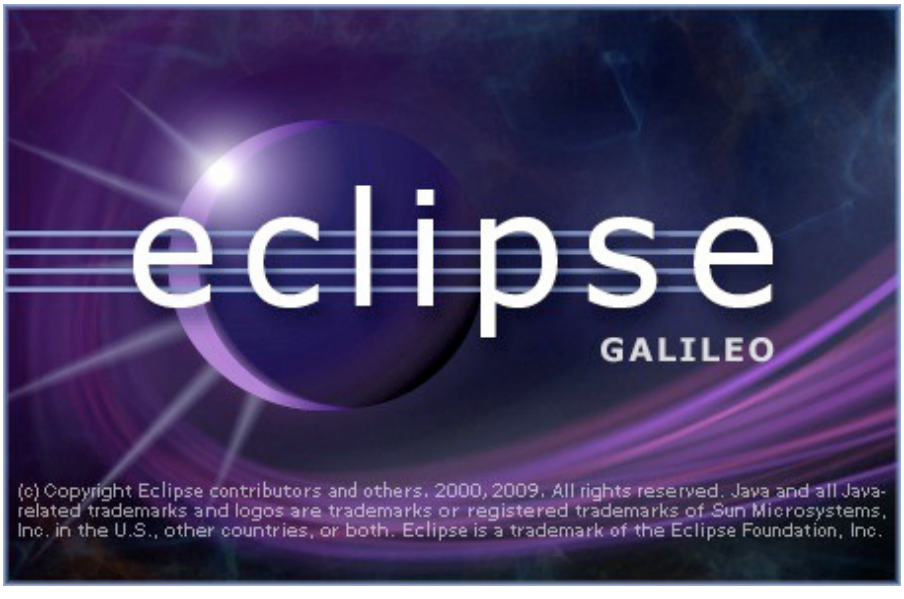
16.4% gedit
14.3% Kate
9.6% Emacs
8.2% Nano
7.3% Other*
5.3% Geany
.9% joe

**Popular write-in: Sublime Text.*

BEST IDE

19.4% Eclipse

16.3% vim
11.9% Qt Creator
8.3% Sublime Text
7.2% Emacs
6.3% KDevelop
5.2% Kate
5% NetBeans
4.9% Other
4.4% Brackets
3% Geary
3% IntelliJ IDEA
2.2% Komodo IDE
2% Code::Blocks
1% JetBrains PhpStorm



BEST PLATFORM FOR DEVELOPING RICH INTERNET APPS

34.8% Qt

23.4% Django
18.9% Ruby on Rails
11.6% Other*
4.8% OpenShift
2.9% Catalyst
2.1% Dojo
1.5% Vaadin

**Popular write-ins: AngularJS and node.js.*

BEST JAVA JRE

Based on your feedback, we're considering renaming this category "best way to get kicked in the shins", but nevertheless, OpenJDK takes more than twice the votes of Oracle's Java this year. Now if those pesky few applications that require Oracle's Java environment would just get with the times.

65.3% Openjdk

30.5% Oracle
3.4% Other*
.9% WSO2

**Popular write-in: "all of them suck".*

BEST JAVA APP SERVER

Much like the JRE question, perhaps we should rename this category "best Java-related cuss word", as we clearly have few Java fans in our community.

57.9% Tomcat

22.5% jboss
9.6% jetty
7.9% Other*
2.2% WSO2

**Popular write-ins: "NONE!", "LOL" and "you're kidding, right?"*

BEST JOURNALING FILESYSTEM

72.5% ext4

13% btrfs
6.5% xfs
4.8% ext3
3.3% Other*

**Popular write-in: ZFS.*

BEST FILE MANAGER

Although these results line up fairly well with the distribution category, it's cool to see third-party file managers on the list. My favorite answer? Bash.

25% Dolphin

17.4% Nautilus
10.4% Bash
10.3% Command line
7.9% Midnight Commander
7.5% Thunar
5.9% Nemo
3.1% PCManFM
2.7% Total Commander
2.8% Zsh
2.8% Other*
2.1% Krusader
1.4% Emacs
.9% ranger

**Popular write-in: Caja.*

BEST *LJ* COLUMN

Um. I. Um. Wow. (And with that eloquent acceptance, Shawn loses the Best Column award.) Seriously, I'm stunned and honored to get voted as the best columnist this year. Thank you so much. But really, with our columnists, it's like picking a favorite ice cream—unless you drop it on the floor, it's hard to go wrong!

18.7% Shawn Powers' The Open-Source Classroom

16% Kyle Rankin's Hack and /
13.2% Dave Taylor's Work the Shell
14.1% Zack Brown's diff -u
10.8% Other
8.5% Doc Searls' EOF
7.3% Joey Bernard's Science Column
6.4% Reuven M. Lerner's At the Forge
5.1% James Gray's New Products

BEST LINUX/OSS ADVOCATE/EVANGELIST

This list reads pretty much like the "autographs I want to collect" list, but regardless of the order, what an incredible group of people. (And yeah, it feels hinky to say that with my name on the list of nominations here, but even if I'm the margin of error, it's an incredible list of incredible people.)

23.4% Linus Torvalds

14.9% GitHub
13.6% Richard Stallman
10.1% Jupiter Broadcasting
7.3% Dodeimedo
5.2% Jon "Maddog" Hall
4.3% Mark Shuttleworth
3.8% Other
2.9% Cory Doctorow
2.6% Jono Bacon
.9% Freecode
2.2% Jonathan Corbet
1.9% Shawn Powers
1.5% Doc Searls
1.3% Carla Schroder
1.2% Sarah Sharp
.8% Jim Zemlin
.7% Thomas S. Hatch
.5% Zack Brown
.3% Roy Schestovitz
.2% Eric Christensen
.2% Don Marti

BEST "WORST" LINUX/OPEN-SOURCE IDEA

Now now, stop laughing. We've all made mistakes. Remember how we dressed in the 1980s? (And if you're too young to remember the 1980s, get off my lawn.)

19.9% GNOME 3

19.5% "Creating a new distro instead of creating a new application"
17.8% Mir (Ubuntu's next-generation display server)
15.9% "Ubuntu's going it alone"
5.9% Liberator (3-D printed handgun)
5.7% "Putting GNU in front of Linux"
4.5% Ubuntu
4.3% LibreOffice fork
3.7% "Poetterings' ideas"
2.9% Other*

**Write-in comments: "all of the above", "pointless Ubuntu bashing", "so should I vote for the one I like or the one I don't?", "this is the year of the Linux Desktop" and "this question".*

BEST NEW OPEN-SOURCE PROJECT (FROM 2012–2013)

Much like the gadget category, this may have to become "Best Open-Source Project That Isn't Raspberry Pi". I agree with the number two spot. I'm anxious to see what happens with FirefoxOS.

46.5% Raspberry Pi

19.4% FirefoxOS
7.6% Brackets
7.5% Nemo Mobile
6.7% Other*
4.7% Manjaro Linux
2.5% The Parallella Supercomputer
2% Rust Programming Language
1.2% Everpad
.6% Pump.IO
.6% TeXnicCenter
.5% MultiSystem
.3% WSO2 Stratos 2.0
.1% OxForMongo

**Popular write-ins: SolydXK and Sailfish/Jolla.*

BEST NEW COMMERCIAL APPLICATION (FROM 2012–2013)

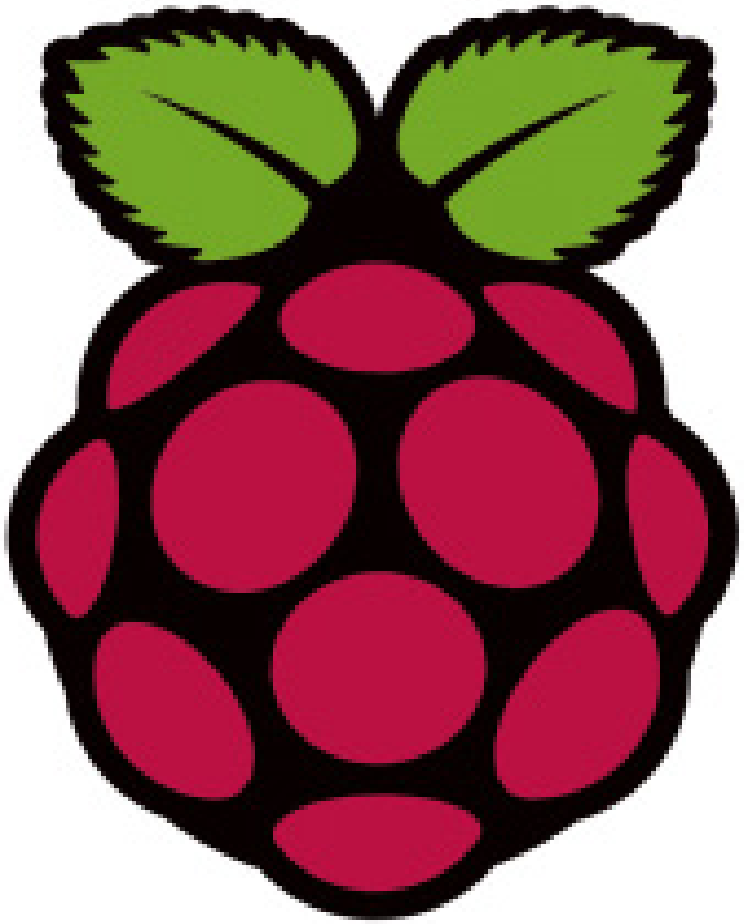
Finally! Steam for Linux is real! No longer the vaporware that I keep writing about, it's a real thing, with real games. This doesn't bode well for my productivity.

74.6% Steam for Linux

5.6% Krita Studio
4.6% LightWorks Pro
4.5% SaltStack
3.6% VueScan Scanner Software for Linux
3% IBM Notes 9.0 Social Edition
2.5% Other
1.6% Clipperz (on-line vault and password manager)

LINUX PRODUCT OF THE YEAR

That's it. Maybe we just need to call this "The Issue Where The Raspberry Pi Wins Everything". Still, the RPi is so awesome, we couldn't eliminate it from the running. For one thing, you'd have flogged us. For another, it's just that cool! Don't let that gold medal take your focus off the rest of the list, however, because it's chock full of awesome. It's been a great year for Linux, and it's only getting better. We can't wait to see what happens next!



34.6% Raspberry Pi

19.9%	Jolla/Sailfish	3.7%	LXDE-Qt	2.3%	SaltStack
11.7%	FirefoxOS	3.2%	Manjaro	2.1%	Plex
8%	Ubuntu Edge	2.7%	Other	2%	Krita Studio
4.8%	Google Chromecast	3.2%	OpenShift	1.8%	Roku 3

Shawn Powers is the Associate Editor for *Linux Journal*. He's also the Gadget Guy for LinuxJournal.com, and he has an interesting collection of vintage Garfield coffee mugs. Don't let his silly hairdo fool you, he's a pretty ordinary guy and can be reached via e-mail at shawn@linuxjournal.com. Or, swing by the #linuxjournal IRC channel on Freenode.net.

We'd like to make 2014's Readers' Choice Awards even better. Please send ideas for new categories, requests for categories to be deleted next time around and any comments or feedback via http://www.linuxjournal.com/contact or to ljeditor@linuxjournal.com.

WEBCASTS

A Call to Arms for Private Cloud Builders

Sponsor: ActiveState | Topic: Cloud Computing **ON DEMAND**

The era of elastic IT is here. Businesses are realizing that the cloud not only allows cost reduction, but provides opportunities for innovation and growth. Elastic clouds enable next-generation applications that drive revenue opportunities, increase agility, and make IT teams competitive with public cloud systems.

In this presentation, Randy and John talk about the forces driving this change, and outline an action plan for building an elastic cloud infrastructure and dynamic applications using DevOps and Platform-as-a-Service.

> **http://lnxjr.nl/CTACloud**

Private PaaS for the Agile Enterprise

Sponsor: ActiveState | Topic: Virtualization

If you already use virtualized infrastructure, you are well on your way to leveraging the power of the cloud. Virtualization offers the promise of limitless resources, but how do you manage that scalability when your DevOps team doesn't scale? In today's hypercompetitive markets, fast results can make a difference between leading the pack vs. obsolescence. Organizations need more benefits from cloud computing than just raw resources. They need agility, flexibility, convenience, ROI, and control.

Stackato private Platform-as-a-Service technology from ActiveState extends your private cloud infrastructure by creating a private PaaS to provide on-demand availability, flexibility, control, and ultimately, faster time-to-market for your enterprise.

> **http://lnxjr.nl/privatepaasAE**

Learn the 5 Critical Success Factors to Accelerate IT Service Delivery in a Cloud-Enabled Data Center

Today's organizations face an unparalleled rate of change. Cloud-enabled data centers are increasingly seen as a way to accelerate IT service delivery and increase utilization of resources while reducing operating expenses. Building a cloud starts with virtualizing your IT environment, but an end-to-end cloud orchestration solution is key to optimizing the cloud to drive real productivity gains.

> **http://lnxjr.nl/IBM5factors**

Linux Backup and Recovery Webinar

Sponsor: Storix | Topic: Backup and Recovery

Most companies incorporate backup procedures for critical data, which can be restored quickly if a loss occurs. However, fewer companies are prepared for catastrophic system failures, in which they lose all data, the entire operating system, applications, settings, patches and more, reducing their system(s) to "bare metal." After all, before data can be restored to a system, there must be a system to restore it to.

In this one hour webinar, learn how to enhance your existing backup strategies for better disaster recovery preparedness using Storix System Backup Administrator (SBAdmin), a highly flexible bare-metal recovery solution for UNIX and Linux systems.

> **http://lnxjr.nl/StorixWebinar**

WHITE PAPERS

Linux Management with Red Hat Satellite: Measuring Business Impact and ROI

Sponsor: Red Hat | **Topic:** Linux Management

Linux has become a key foundation for supporting today's rapidly growing IT environments. Linux is being used to deploy business applications and databases, trading on its reputation as a low-cost operating environment. For many IT organizations, Linux is a mainstay for deploying Web servers and has evolved from handling basic file, print, and utility workloads to running mission-critical applications and databases, physically, virtually, and in the cloud. As Linux grows in importance in terms of value to the business, managing Linux environments to high standards of service quality — availability, security, and performance — becomes an essential requirement for business success.

> **http://lnxjr.nl/RHS-ROI**

Standardized Operating Environments for IT Efficiency

Sponsor: Red Hat

The Red Hat® Standard Operating Environment SOE helps you define, deploy, and maintain Red Hat Enterprise Linux® and third-party applications as an SOE. The SOE is fully aligned with your requirements as an effective and managed process, and fully integrated with your IT environment and processes.

Benefits of an SOE:

SOE is a specification for a tested, standard selection of computer hardware, software, and their configuration for use on computers within an organization. The modular nature of the Red Hat SOE lets you select the most appropriate solutions to address your business' IT needs.

SOE leads to:

- Dramatically reduced deployment time.
- Software deployed and configured in a standardized manner.
- Simplified maintenance due to standardization.
- Increased stability and reduced support and management costs.
- There are many benefits to having an SOE within larger environments, such as:
  - Less total cost of ownership (TCO) for the IT environment.
  - More effective support.
  - Faster deployment times.
  - Standardization.

> **http://lnxjr.nl/RH-SOE**

It's about the User: Applying Usability in Open-Source Software

You don't have to be an expert to apply usability tests in open-source software. Anyone can do it. And with good usability, everyone wins. JIM HALL

Open-source software developers have created an array of amazing programs that provide a great working environment with rich functionality. At work and at home, I routinely run Linux on my desktop, using Firefox and LibreOffice for most of my daily tasks. I prefer to run open-source software tools, and I think most *Linux Journal* readers do too. But as comfortable as the open-source software ecosystem can be, we've all shared or heard the same comments about some of our favorite Linux programs:

- "___ is a great program, once you figure out how to use it."
- "You can do a lot in ___, after you get past the awkward menus."
- "You'll like using ___, if you can learn the user interface."

That's the problem. No matter how powerful the program, that functionality is lost if people have to figure out how to use the program in order to unlock its secrets. Typical

users with average knowledge should be able to operate a general-purpose program. If a program is hard to use, that suggests the problem is with the program, not with the user.

Usability and Open-Source Software

"Usability" refers to how easily users can learn and start using software, or any similar "information product". Usability is separate from the functionality of the program, and so usability testing is different from unit testing. Instead, usability testing allows us to uncover issues that prevent users from using our programs.

Most open-source software programs are written by developers for other developers. Although some large open-source programs, such as GNOME and Drupal, have undergone usability testing, most projects lack the resources or interest to pursue a usability evaluation. As a result, open-source software programs often are utilitarian, focused on the functionality and features, with little attention paid to how people will use it. Applying usability practices tends to be antithetical to how open-source software is created. Open-source developers prefer functionality over appearance. Although some projects may have a maintainer who dictates a particular design aesthetic, many more do not. In an interview for this article, open-source advocate Eric Raymond commented to me that most programmers view menus and icons "like the frosting on a cake after you've baked it", which is an apt metaphor. Open-source software developers tend to prefer assembling the ingredients and baking the cake, not applying frosting to make it look nice.

So how can open-source developers easily apply usability to their own programs? There are many ways to implement usability practices. Alice Preston described 11 different techniques to evaluate usability in the STC Usability SIG newsletter. These methods run the gamut from interviews and focus groups to heuristic reviews and formal usability tests:

1. Interviews and observations: one-on-one sessions with users.

2. Focus groups: often used in marketing well before there is any kind of prototype or product to test, a facilitated meeting with multiple attendees from the target user audience.

3. Group review or walk-through: a

facilitator presents planned work flow to multiple attendees, who comment on it.

4. Heuristic review: using a predefined set of standards, a professional usability expert reviews someone else's product or design and shares comments with the designer.

5. Walk-around review: copies of the design or prototype are tacked to the walls of a conference room, and testers are invited to examine them and make comments.

6. Do-it-yourself walk-through: make mock-ups of the design, and use realistic scenarios to walk through the design yourself.

7. Paper prototype test: use realistic scenarios of a fake product that is still in design.

8. Prototype test: a step up from the paper prototype, test an animated mock-up against realistic scenarios.

9. Formal usability test: using a stable product, an animated prototype or even a paper prototype, test a reasonably large number of subjects against a controlled variety of scenarios.

10. Controlled experiment: a comparison of two products, with careful statistical balancing.

11. Questionnaires: ask testers to complete a formal questionnaire about how they would use a design.

However, such formal usability practices tend to clash with the open-source developer community and are like "swimming against a strong cultural headwind", to mix metaphors from Eric Raymond. With that in mind, developers should consider a subset of usability methods that apply well to the culture of open-source software. I propose this list:

1. Heuristic review.

2. Prototype test.

3. Formal usability test.

4. Questionnaires.

You don't need years of usability experience to apply good usability practices in open-source software development. As suggested by usability expert Janice (Ginny) Redish in numerous articles, you can learn a lot just by sitting down with

a few users and watching them use the software.

Whatever method you choose, the value of usability testing is in practicing it during development, not after the fact. Apply usability testing iteratively using prototypes, even paper-based mock-ups. At each round of testing, you will identify a number of issues that you can resolve for the next version. Successive usability tests will uncover additional issues and further improve your project.

Applying Usability Tests to Your Own Programs

Let's walk through a usability test as an example. Remember, the purpose of a usability test is to uncover issues that general users might have in utilizing the program. It is not a functional test of the program's features, but a practical test of its ease of operation, and as such, it differs from quality assurance or unit testing.

To start, define a set of written scenarios that represent how typical users with average knowledge would use the program. Don't look at every feature of the program, just describe the tasks that most users would want to do with the software. Make your scenarios realistic; provide a short description of each scenario, and ask the tester to perform tasks within that context. Use simple language; don't lead the tester by using the product's own words to describe menu items or actions, especially if typical users of average knowledge are unlikely to use those words every day. For example, if you want to evaluate an editor, your scenarios might ask the tester to type a short text document, save it and make basic copy edits to the file. For a Web browser, your scenarios could ask the user to search for a Web site, bookmark it and save a copy of the Web page for off-line use.

Invite testers to join you for a usability test. Although you might think you need a lot of users to evaluate a program's usability thoroughly, you really need only about five testers to get useful results, as usability expert Jakob Nielsen asserts in his research. Present the testers with the scenarios, one at a time, each on a separate piece of paper, and ask them to complete the tasks. Then simply observe what they do in the program, the routes they take to accomplish the tasks and the problems they encounter. Take plenty of notes.

The most difficult part of a usability test is watching a tester struggle to locate a menu or button. Although the correct action might seem apparent to you immediately, the

value lies in learning and identifying what is not obvious for other users. Do not give hints. If a tester is unable to finish a scenario, that's okay; just move on to the next scenario.

At the end of the scenarios, take a few minutes to ask follow-up questions of your tester. Identify any areas that seemed particularly difficult for the user. For example, you might ask "You struggled when you tried to do X; what would have made it easier?" or "What were you expecting to see on the screen when you were doing Y?" As a final wrap-up, ask the tester to describe what worked well in the program and what features should be improved.

Welcome to My Usability Test

I reviewed three common open-source projects in a formal usability test. I did this both to demonstrate the usability test process and to generate usability test results that could be generally applied to other open-source programs. Choosing the programs for my study required careful consideration. The ideal programs for my demonstration needed to balance multiple qualities: not be too big, because very complex menus can "lose" the audience in the details and confound the usability test results, and not be too small, as trivial programs will not support generally applicable conclusions. Further, the programs needed to be approachable by general users.

I solicited advice on several on-line forums, asking which open-source software programs had good usability. Sorting through the suggestions, three projects matched the criteria for my usability test:

1. Gedit (a text editor for GNOME).
2. Firefox (a popular Web browser).
3. Nautilus (a file manager for GNOME).

Because I work on a university campus, I invited students, faculty, staff and members of the public to participate in a usability study. I didn't ask for a specific level of technological expertise, as I was looking for typical users with average knowledge. In most formal usability tests, it's common to present each tester with a small gratuity; I gave out free pop and candy for them to take home with them after the test.

Although my preferred goal was about a dozen testers, I was satisfied with the seven who participated in the usability test. They ranged in age from about 20 to about 40, with three

men and four women. Most testers (five) claimed "low" experience with computers, and almost all (six) used Windows as their primary desktop. Testers used separate guest accounts on a laptop running Fedora 17 Desktop Edition, and so they started from the same initial default settings.

At the start of the usability test, I gave each tester a brief context of the usability study. I explained that this was a usability test, so it was about the software, not about them. If the tester experienced problems during the test, I let them know that would be okay, and we could move on to the next section. I was there only to observe their interaction with the software, not to judge their performance. Along the way, I said I would take notes and watch what was happening on their screens.

I also asked the testers to speak aloud what was going through their mind during the usability test. For example, if they were looking for a Print button, they should simply say, "I'm looking for a Print button." And, I encouraged them to track the mouse cursor on the screen with their eyes, so I could observe where they were looking for menus and buttons.

During the usability test, I presented the testers with a number of scenarios, each providing a brief context and an action they were to complete. For example, after asking testers to navigate to the BBC News Web site in the Firefox browser, one scenario asked them to increase the size of the text on the screen. It's important to note that the scenario did not use the same wording that was present in the menu action to apply the font size change:

> You don't have your glasses with you, so it's hard to read the text on the BBC News Web site. Please make the text bigger on the BBC News Web site.

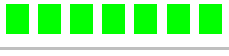
Overall,the usability test included 19 scenarios, which testers completed in 30–40 minutes.

What Were the Usability Issues?

A heat map is a good way to represent the issues uncovered during a usability test. In Figure 1, each row represents the task, and each block represents a tester's experience. Green blocks indicate the tester was able to complete the task easily, usually on the first attempt. Orange and red blocks denote scenarios where the tester experienced difficulty or was unable to complete the task, respectively.

Interestingly, almost everyone

Gedit		G1. Type a sample note (provided) and save the file.
		G2. Edit text within the note.
		G3. Replace all instances of several words.
		G4. Save the file under a new name.
		G5. Change the default font.
		G6. Change the default colors.

Firefox		F1. Search for and navigate to the BBC News website.
		F2. Set the website as the default page.
		F3. Increase the font size.
		F4. Create a new tab and navigate to www.freedos.org.
		F5. Save a copy of a web page for offline use.
		F6. Download an image from the website.
		F7. Create a bookmark to the website.

Nautilus		N1. Create a folder.
		N2. Move the folder to a new location.
		N3. Rename the folder.
		N4. Create a bookmark or shortcut to a folder.
		N5. Delete a file.
		N6. Search for a file.

Figure 1. Usability Heat Map

experienced the same four issues:

- G5: change the default font in Gedit.
- G6: change the default colors in Gedit.
- N4: create a bookmark or shortcut to a folder in Nautilus.
- N6: search for a file in Nautilus.

In Gedit, testers were very confused about how to set the default font

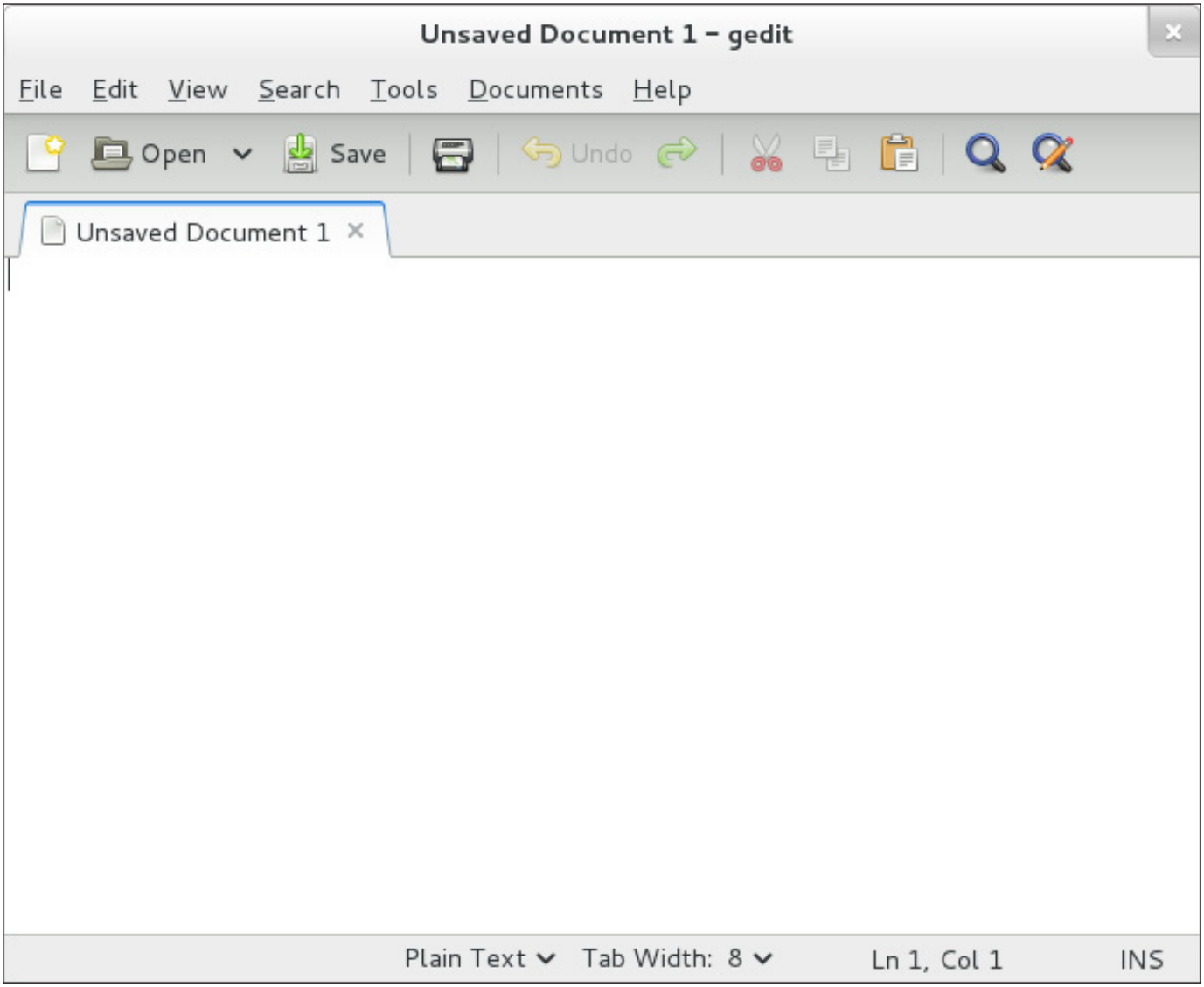


Figure 2. Gedit Screenshot—"How to Change the Font?"

and colors. Part of this confusion stemmed from thinking of the editor as if it were a word processor, such as Microsoft Word, which uses items on the toolbar to accomplish either action. Testers reported they were looking for a menu item called "Font". Failing that, testers also looked in File, Edit, View and Tools.

In Nautilus, testers became frustrated while trying to create a bookmark or shortcut to a folder, and the only user who successfully created the bookmark later commented she did so by accident. As explained in the scenario, the folder would be used for a project that collected photos and was located in the Pictures folder.

The most common action was to go into the Pictures folder, click on the

project folder, then select "Bookmarks - Add Bookmark". Nautilus doesn't display messages to the effect that "Add Bookmark" only creates a bookmark to the current location, not to a highlighted item, so testers were left confused when nothing happened.

Similarly, most testers found searching for a file in Nautilus a difficult task. They did not realize that the Search function starts from the current folder, not from their Home directory. Only two testers were able to use Search successfully. Of these, one happened to click on the Search button from the home directory. The other tried changing options in the drop-down Search action until eventually picking a combination that worked. One tester gave up with Search and

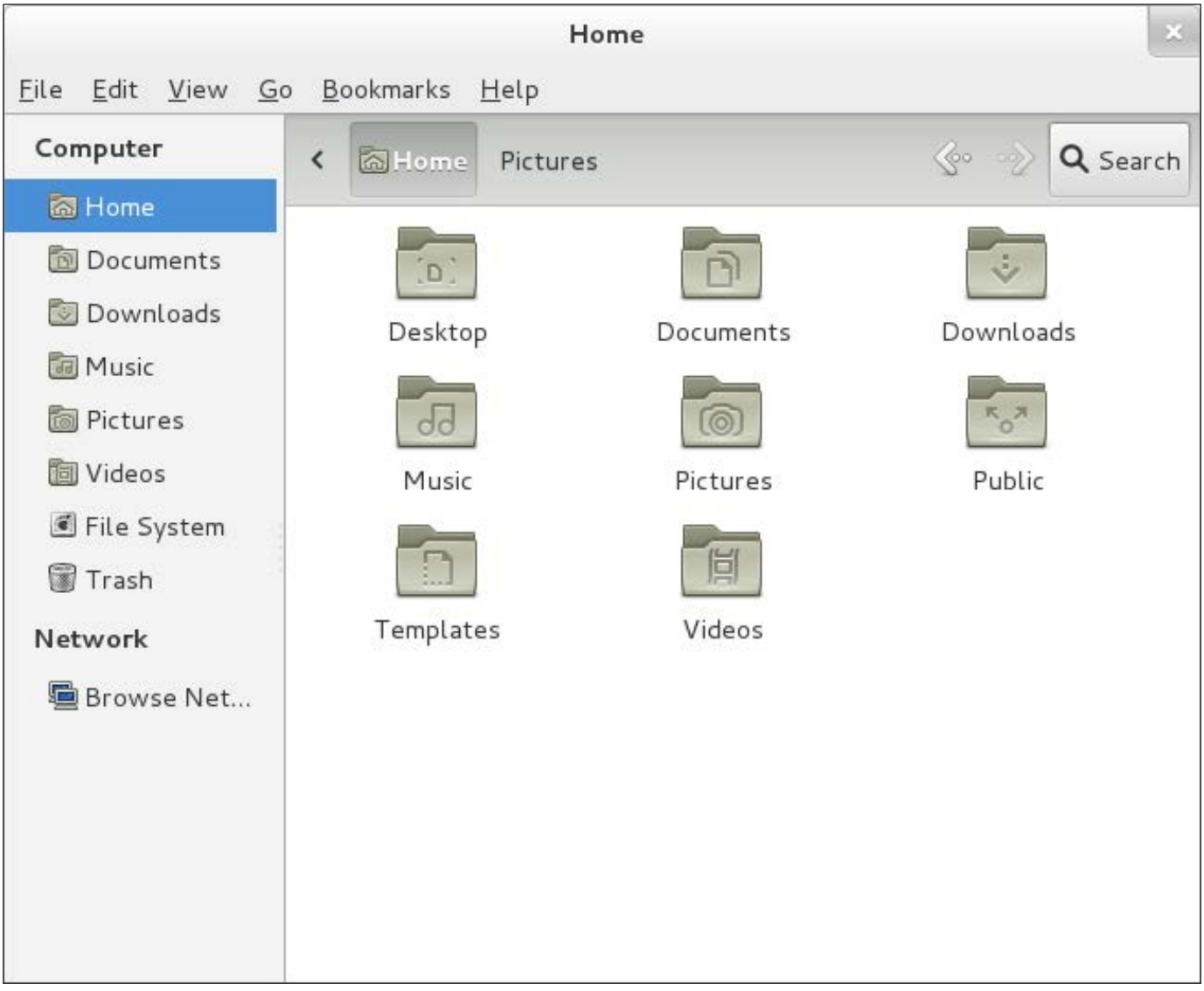


Figure 3. Nautilus Screenshot—"How to Create a Bookmark or Shortcut to a Folder?"

navigated into each folder in turn to find the file. Another user opted not to use Search at all, and used the same "seek and find" method.

And although GNOME was not part of the usability test, almost all testers experienced difficulty with the GNOME "Activities" menu hot corner. In the GNOME desktop environment, the Activities menu shows the list of available programs plus a view of the running applications. Users can bring up the Activities menu by clicking the menu icon in the upper-left corner of the desktop or by moving the mouse into that corner (the "hot corner"). Usually right away in the first scenario, testers would "overshoot" the program menu they were looking for, and hit the GNOME hot corner instead. This also occurred several other times throughout the usability test. Although testers were able to recover from the hot corner, it definitely caused frequent disruption.

What Worked Well for Usability?

Throughout the study, I observed four themes of good usability that allowed all testers to pass quickly through those parts of the usability test:

1. Familiarity: testers commented that the programs seemed to operate more or less like their counterparts in Windows or Mac OS X. For example, Gedit isn't very different from Windows Notepad or even Microsoft Word. Firefox looks like other Web browsers. Nautilus is quite similar to Windows Explorer or Mac OS X Finder. To some extent, these testers had been "trained" under Windows or Mac OS X, so having functionality (and paths to those features) that was approximately equivalent to the Windows or Mac OS X experience was an important part of their success.

2. Consistency: user interface consistency between the three programs worked strongly in favor of the testers and was a recurring theme for good usability. Right-click worked in all the programs to bring up a context-sensitive menu. Programs looked and acted the same, so testers didn't have to "re-learn" how to use the next program. Although the tool bars differed, all programs shared a familiar menu system that featured File, Edit, View and Help.

3. Menus: testers preferred to access the programs' functionality from the menus rather than via "hot keys" or icons on the toolbar. For example, the only toolbar icon that

testers used in the Gedit scenarios was the Save button. To complete other scenarios, testers used the drop-down menus, such as File, Edit, View and Help.

4. Obviousness: when an action produced a clear result, or clearly indicated success (such as saving a file in the editor, creating a folder in the file manager, opening a new tab in the Web browser), testers were able to move through the scenarios quickly. When an action did not produce obvious feedback, the testers tended to become confused. The contrast was evident when trying to create a bookmark or shortcut in the Nautilus file manager. In this case, Nautilus did not indicate whether the bookmark had been created, so testers were unsure if they had completed the activity successfully.

These are good lessons in open-source software and usability. Your program's user interface doesn't

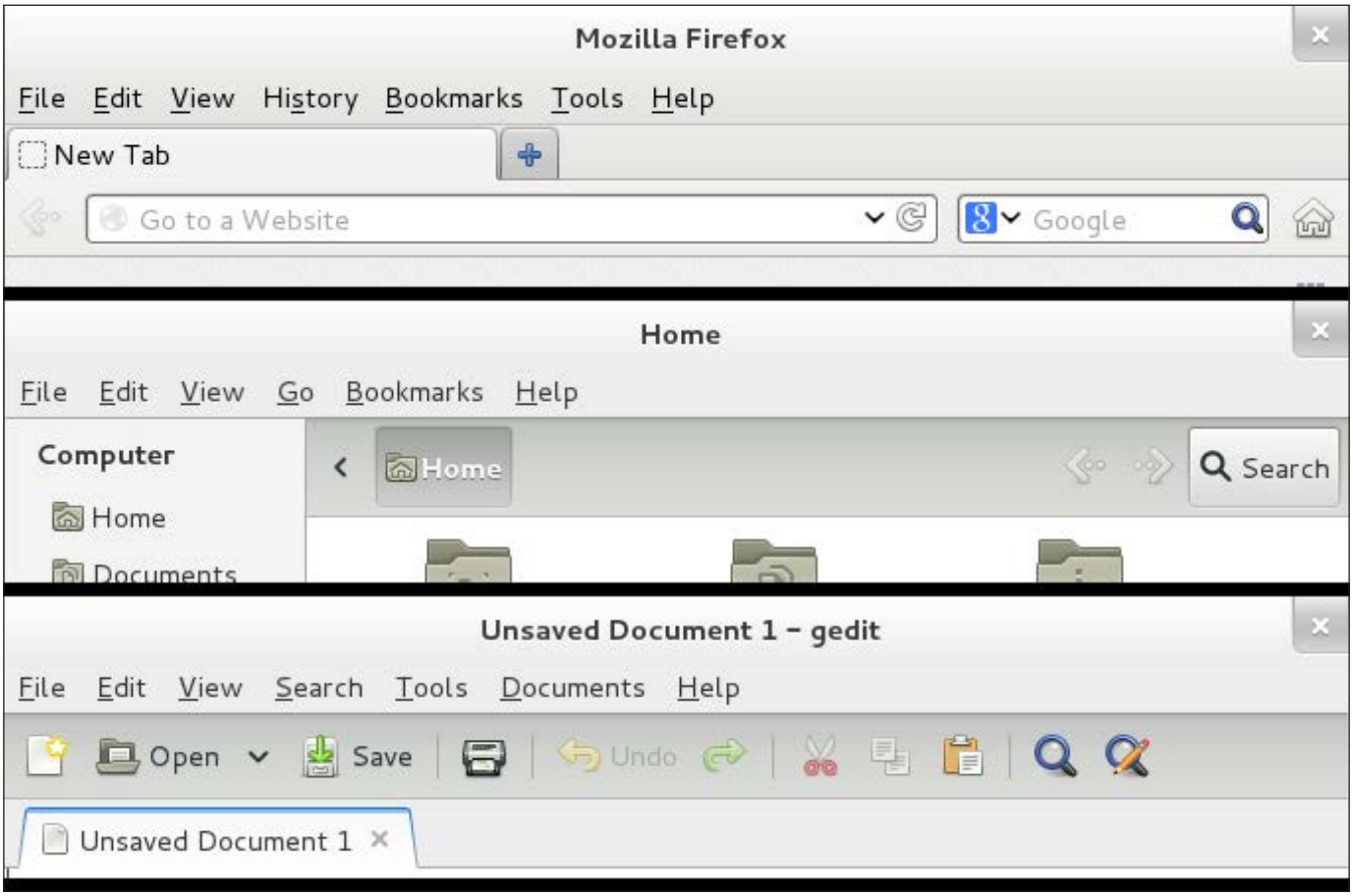


Figure 4. Firefox, Nautilus and Gedit Interface Comparison

have to be a beautiful impediment to understanding. Instead, leverage existing user interface paradigms. Be consistent with other programs on the same platform, whether they are other open-source software or proprietary programs. Use menus that are clearly labeled. Ensure that every action has a result that is obvious to the end user, especially if that result indicates a failure.

Where Do We Go from Here?

Usability should not be something that's just tacked onto a project or addressed only at the end of a development life cycle before releasing the next version of the software. Usability needs to be part of the design of open-source software, and addressed as part of a process. As open-source software developers, we generally are very good at applying good software development practices to our work. Now we need to take the next step and bring usability into that methodology.

Our next challenge in open-source software is finding ways to incorporate usability into our developer culture. This is a big step in open-source software. To date, usability has been antithetical to how open-source developers work. Most projects are written by developers for other developers. Crafting new functionality takes priority, and we rarely look at how our users will try to access those features.

In open-source software projects, the user community plays a strong part in testing each new release. Unfortunately, we cannot rely on the typical user-testing cycle to provide good usability feedback. Left on their own with no structure to usability testing, open-source software testers will respond with bland bug reports, such as "This feature is confusing." That's not helpful to a developer. In addition, usability researchers David Nichols and Michael Twidale comment in their published work that developers may not grant usability issues the same status as functionality bugs, leading to an inherent developer bias against usability bugs.

The approach to identify usability issues in open-source software, therefore, needs to be more structured. Open-source software developers can apply a variety of methods, although the ideal would be to conduct formal usability tests with a handful of users. Remember, you need only about five testers to get useful results.

Usability testing for open-source software projects doesn't need to be performed in a stuffy lab environment;

a project can find ways to "crowd source" usability testing with the user community. For example, the open-source Web content management system Drupal streamed testers' desktops as they undertook a usability test. This allowed Drupal developers all over the world to observe the usability test without having to travel to a single location. When developers can watch testers experience problems with their software, they better understand the issues and how to address them.

Another simple method is usability testing by "flash mob", a term suggested by Dana Chisnell on her Usability Testing Howto blog. To do testing via flash mobs, researchers simply can intercept people in a public space and ask them to try a few scenarios against a paper prototype and share their experiences. If each subject is willing to spare a few minutes, a "mob" of such testers will provide valuable feedback in a short amount of time. This idea of "flash mob" usability testing can be extended to other domains too. If your open-source software project is implemented as a Web site, you can conduct similar impromptu usability tests by intercepting Web visitors.

You don't have to be an expert to apply usability tests in open-source software. Anyone can do it. You only need to watch users try to use a program, and usability issues quickly will become clear. A handful of testers operating against a prototype can give you the feedback you need to make your open-source software even easier to use. And with good usability, everyone wins.■

Jim Hall is an advocate for free and open-source software, best known for his work on the FreeDOS Project. At work, Jim is the Director of Information Technology at the University of Minnesota Morris. He also is working toward his MS in Scientific and Technical Communication at the University of Minnesota.

Send comments or feedback via http://www.linuxjournal.com/contact or to ljeditor@linuxjournal.com.

Resources

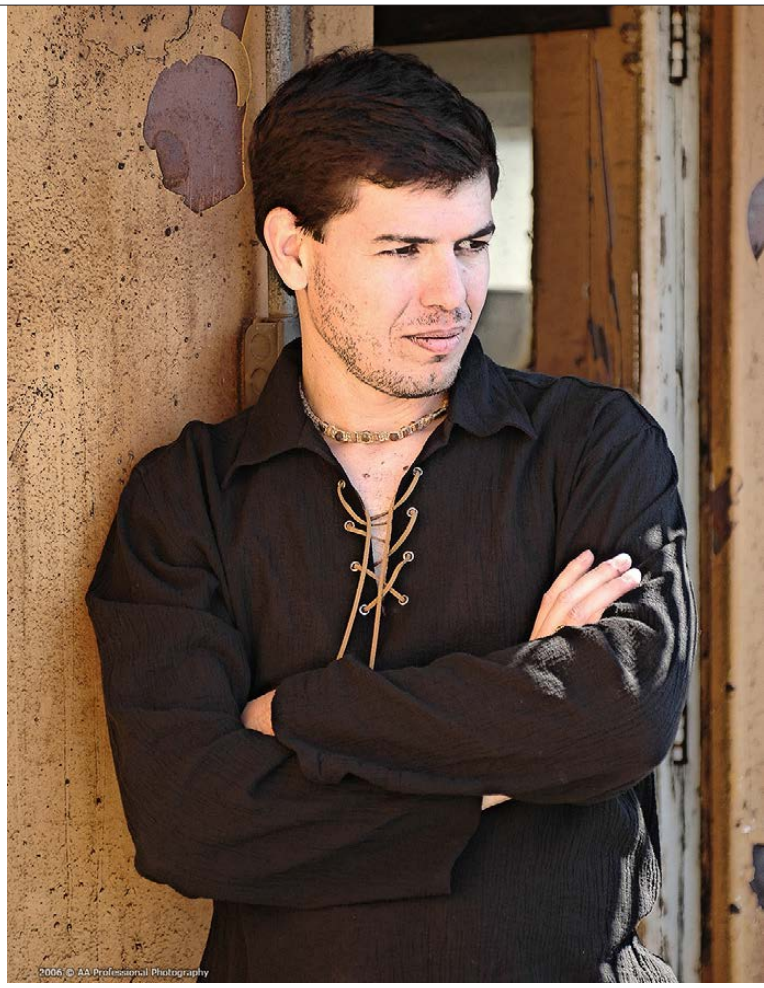
Open-Source Software and Usability (author's blog): **http://opensource-usability.blogspot.com**

More Information about Usability and Usability Testing: **http://usability.gov**

Janice (Ginny) Redish on Usability (good reference for Web developers): **http://www.redish.net**

Jakob Nielsen on Usability: **http://www.useit.com**

Interview with Australis



Linux developers are usually interesting people. So are graphic artists, musicians and sound engineers. So when a successful musician combines all of these traits, we simply have to interview him.

FRED MORA

FM: Australis, who are you?

Australis: I am Oscar Aguayo, a music composer and producer. Australis is, of course, the name under which I release my music. I've released three physical albums since 2005, along with one free digital compilation. I am currently completing the material for a fourth album that I hope to release later this year.

FM: You are a successful indie musician. Do you have a day job?

A: Yes, I do have a day job. I'm a software engineer at a well-established local company in Salt Lake City, Utah.

However, we indie artists define "success" differently from what the music industry has got the world used to for a few centuries. As independents, we have absolute freedom to define what "success" means to each one of us.

To me, success is when my music is listened to and enjoyed around the globe, or when listeners from as far away as Singapore or The Netherlands ask me for the release date of my next material, or when I am approached by serious contributors to serious publications, like you.

The money that comes from my music is nice—I can't deny it. But it is not part of what I consider my personal definition of "artistic success".

FM: How did you discover Linux?

A: Some years ago, I was working as a head of software development. I discovered that several of the paid Windows tools we needed to develop and test our product existed as comparable and fully functional open-source projects on Linux. That was my turning point.

I started with an Ubuntu Live CD to try things out without repercussions to the existing Windows installation we used, and it almost felt like I had been lied to all my life. As I explored office suites, development environments, tools and utilities, I kept asking myself why I hadn't explored this universe of openness and power before!

After a few months (mostly spent convincing the directors), the company's software development department had migrated to a pure Linux environment.

For me, the conversion also was complete. I still find myself forced to use Windows in one specific context, but for all others, I use only Linux.

FM: Your music is, schematically, melodic instrumental synths. It has been compared to Jean-Michel Jarre and recent Tangerine Dream. Do you agree?

A: Yes, but it is a relative "yes". Jean-Michel Jarre and Tangerine Dream are, among others, some of the first exponents of electronic new age music. I was a very young teenager when they were the only source of music in this genre. Their influence on what would become my personal composing style is something I have to admit. Vangelis and Enigma are other particularly strong sources of influence in my music.

As a composer though, I discovered that without a conscious effort, every composer ends up "boxed" in his own style, repeating himself again and again, creating music that is only a copy of what he has already created before. For that reason, I always get away from my comfort zone in order to explore other genres and styles and offer a fresh and varied mixture of musical landscapes to listeners.

FM: Do you use Linux for the composition?

A: This is the one context in which I am forced to use Windows instead

of Linux. Unfortunately, and I say this with real regret, the number of synths and virtual libraries for music production available on Linux is still seriously behind compared to what's available on Windows.

There are some very serious pieces of software for music production out there, don't get me wrong. But you have to see it from the composer's point of view. Inspiration comes to you at any moment, and when it does, you have to record what it's telling you by whatever means you have at hand at that moment.

A piano is ideal if you are at home. A digital recorder works too if you are away from an instrument. If nothing else, a notepad or even a napkin can work too. But later, when you are ready to build your new music piece from that initial annotation, would you do it with only a fraction of the orchestra present?

That's the dilemma when trying to compose in Linux. You have very nice instruments available, but not all. And, since you are composing—in other words, you don't know beforehand what sounds you'll want to use to assemble your piece—you need all sounds available, all libraries and instruments on standby in case you need to use or tweak them.

FM: You do use Linux to master the MP3s and other files you sell or distribute, in order to guarantee they are not propagating viruses. Is the public sensitive to the security argument?

A: I do use Linux when mastering my material, yes. And one of the many strong reasons I do is because of how secure and clean Linux is regarding viruses and malware, compared to Windows. I know the public is very sensitive to the security of their computers and networks, especially now that phishing, malware and identity theft are so widespread.

Personally, I have not heard concerns about how secure the digital versions of my music are. That is a main concern of mine, however, and knowing that the systems I use to produce, master and deliver my music to the world are 100% secure and clean is a very important step in the whole process.

FM: You are using Linux for composing CGI images. Can you tell us about your graphic artist side? Which tools are you using?

A: I've always been interested in graphic composition, including 3-D modeling and rendering. In fact, I

Australis Cover Image

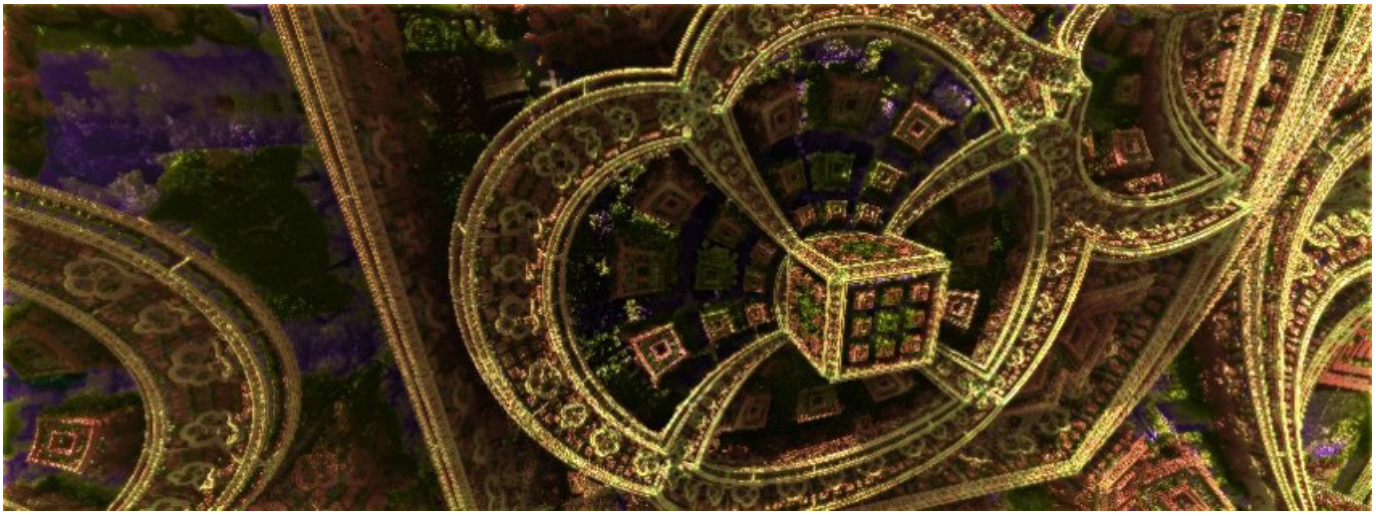
used to do commercial 3-D animations in the previous century. I had to relegate all of this to an "as-needed" level when I decided to concentrate on music composition and production back in 1999 even before Australis.

Fortunately, indie artists have more possibilities of participating in other aspects of their music. Fortunately, I had the opportunity to create the covers for all my albums, the graphics for Australis' Web site and its profile on Facebook. Obviously, there have been other people behind most of those graphic compositions, like

Australis Cover Image

photographers and graphic designers providing their talent here and there. But for the most part, I've been lucky to be able to be directly involved in the graphic side of Australis too.

When dealing with photography—like for the cover of my album *The Gates of Reality*—I use GIMP almost exclusively, complementing with Inkscape for typography and vectors. When creating abstract compositions, I use other tools. For example, for the cover of *Sentient Genus*, I used a Linux port of the fractal app *Apophysis* that allows you to manipulate all parameters directly to create incredibly complex images. Lately, I've been exploring



Fractal Image Titled "Inside the Castle, Part I"

Fractal Image Titled "Left Behind"

a 3-D fractal creation tool for Linux called Mandelbulber.

Whatever tools I use to create the main graphic components though, I always finish the final images on GIMP.

FM: From your very first release (*Lifegiving*, 2005), you were noted for your polished sound and high-level production. You were clearly not a beginner when you released the first Australis album.

A: Well, I've always been the composer for the bands I have played with, but in 1999, I had the wonderful opportunity of also being the producer for an independent rock/pop band called Cabala (pronounced with an accent in the first a, like cAbala). You should have heard the first material I produced! It was so deficient, lacking and even disproportionate in how every element participated in the mixes. You may know how to use the tools for the production of audio, but until you have educated your ears, you can't produce clean and high-quality music!

The band dissolved in 2004 (which is what pushed me to create Australis as a solo project that same year), but by then I had a better trained auditive perception, which fortunately has helped me with Australis' sound.

FM: Do you find uses for Linux in sound engineering?

A: Part of the pleasure of becoming a Linux user is to migrate as many of your normal tasks to Linux as possible. As I mentioned earlier, most of the composition of my music has to be done in Windows. But as soon as I am done with the creation part of the process, I switch to Linux to do sound processes like mixing, normalization and mastering, using Audacity's many professional-level features.

As a side comment, I wish more

Oscar's Workstation Running Linux

companies and/or developers would engage in producing serious virtual music instruments and libraries for Linux. The platform is more stable and reliable than the rest out there. And I am sure I am not the only composer who would make a complete migration if there were enough tools for music creation on Linux.

FM: Do you rent time in a professional studio, or do you have an in-house studio?

A: I have had an in-house studio for several years. When playing with Cabala, we rented a professional studio at the beginning, and although the results were very satisfactory, the expense left us broke for a couple months. That's how I became their producer and sound engineer—we couldn't afford that expense again for a long time.

When I created Australis, I was in a better position to modify a room at home in order to get a clean input when recording acoustic instruments.

FM: What instruments do you play? Do you hire musicians for recordings?

A: With a few exceptions, I play all the instruments you can hear in my music—piano, synthesizers, guitar, drums, percussion. Even some vocals throughout my albums are my real voice. When I need something I can't do by myself though, I bring in other musicians to participate—every time I need a female voice, for example, or whenever I am creating an ethnic piece that requires very specialized performances with less traditional instruments.

FM: Do you carry over your software development habits when you work on music? Or does the musician side prefer improvisation and inspiration-driven, spur-of-the-moment setups?

A: No, I am a structured being. I need everything organized in a meaningful manner to be able to function musically—or to function in general. I've come up with my own ways to organize files, backups, documentation and so on—nothing weird or cryptic, just my own way to place things where I know I'll search for them later.

This is an enormous advantage when the "musician side" kicks in and wants to improvise and follow the inspiration of the moment. I know where to go to load the sounds I want at that moment, and I keep adding tracks to the improvisation as inspiration keeps flowing. You really

appreciate being organized when you are in a hurry trying to register the inspiration suddenly striking you. Being able to load a specific percussion set within 30 seconds of arriving to the studio, for example, versus spending ten minutes browsing through libraries in search of some useful percussive sounds, may actually be the difference between registering your musical idea or losing it.

FM: What was your first computer? Do you remember your first program?

A: I do remember my first program, and it was in BASIC!

My first computer was a Tandy TRS-80. It was the early 1980s, so not many young people may be familiar with that computer. It was the time when personal computers were still very expensive, and a new line of hobby computers (like the Commodore 64, and others) came about. It was basically a keyboard attached to a small CPU that connected to your TV.

I was a young teenager then, and living in Peru, I knew no English whatsoever. The youth-oriented BASIC book that came with the computer was, of course, in English. But the attraction was too strong, and with the help of a dictionary and equivalent amounts of frustration and curiosity, I started to understand the underlying concepts and finally make sense of the new universe opening in front of me.

My very first BASIC program was a simple implementation of the "hangman" game.

FM: Let's go back to your software engineering side. Your Web site says: "I believe there's an important difference between 'engineering' software and merely 'programming' it. I believe in engineering. I also believe in sound logic. I believe in clean code. I believe in documentation. I believe in coding with the future in mind....I believe in creating solid components for the software I develop." That's a strong philosophical statement about how you view your work. Do you think that using Linux helps you adhere to this philosophy?

A: Definitely. I was a software developer way before I became a Linux user. Windows was all I knew at that time, and like people who have not had the opportunity to travel to other cultures and see other societies and idiosyncrasies, I ended up thinking this small circle was all there is. I developed programs with Microsoft tools, using Microsoft languages, deployed on Microsoft

platforms. However good of a developer you are, your ignorance about what else is there in the world is a strong limitation.

Beyond developing, engineering is the ability to see the whole of a system, not just its parts. If you don't have a clear view of the whole picture (which includes other platforms, issues like portability, compilation and so on), how can you be an engineer?

Moving to Linux from Windows first and Mac later gives you a wider understanding of these issues, allowing you to code better, thinking ahead for possibilities like migration, compatibility, maintenance and so on.

Another big help comes from the development infrastructure available on Linux. Not only does it have nothing to envy Microsoft's or Apple's, but also I honestly feel it is much better. The open-source philosophies have reached maturity and have favored such an amount of environments and tools and resources—most of them naturally oriented towards open-source platforms like Linux—that I would seriously recommend to any developer or software engineer to use Linux from the start.

FM: Some musicians complain that the Internet makes it too easy to pirate music. You don't use DRM in your releases. That's a hot topic among OSS supporters. What's your opinion?

A: This is a touchy subject in the music industry, and every publishing musician I know has an opinion, one way or another.

My own opinion is that, yes, the Internet makes it very easy to pirate a musician's work. However, it is also my opinion that piracy always has existed. Remember when you would insert a blank cassette into your tape recorder and wait for hours until your favorite station played the song you wanted to record? According to the modern definition of music piracy, we were pirates when doing this in the 1980s.

Why is it that nobody cared about piracy back then? Why is it such a hot topic now? Imagine musicians lobbying for the suppression of sales of blank cassettes. Ridiculous, isn't it? What is different now though? I believe the answer is money.

The Internet allows the acquisition of music, for free, at a much, much larger scale. And then the record labels started to lose money (the musicians too, but at a much smaller scale; the big money from music goes to record labels). Is this bad? I honestly don't know.

I can't go into the philosophical principles at play here, but I think there is something fundamentally wrong with the model the music industry has been following since the last century. Piracy is a problem, yes, but I think it is more a problem for the record label than for the musician. Record labels keep 90% of music sales while the artist receives 10% or less.

On one hand, this is understandable because a record label has professionals and facilities and channels to make the artist's music reach the audience, and that has a cost. But on the other hand, doesn't it ring wrong when 90% of what you are paying for a CD is non-artist-related overhead?

So, yes, the Internet has allowed the piracy that has existed for decades to grow much wider. But is it wrong? Or is this a sign that the music industry's model is obsolete? I don't know. Maybe record labels need to make themselves leaner now that the Internet is doing most of the distribution work for them? Maybe we artists need to find a way to give more to listeners if they buy the CD than if they download the music for free?

FM: Your work is independently released. Any misgivings about big labels?

A: Yeah. Distrust. Without any disrespect intended, a record label is nothing more than a company with the sole purpose of making money. Their purpose is not to mentor aspiring artists or to provide them with useful artistic services. And certainly, their purpose is not to "help" any artist, but themselves. I know it sounds bitter, but consider it at the same level as a bank or an insurance company. They care about their customers as long as they produce profit for the company. The moment customers become expenses, the company loses all interest in them. Although cold, this is just the way any commercial company works.

We artists, however, tend to "romanticize" record labels as some sort of benefactor that will recognize our true artistic potential and will then, out of love for art—"the world needs to hear your talent" kind of thing—will take our hand and guide us through the maze-like path to stardom.

The truth is that record labels are looking only for an investment. And they expect a return on that investment. After all, as any commercial organization, they need to pay salaries, expenses and so on and hopefully make a profit while at it. So, when searching for

a potential new artist, their first criteria has to be "will this new artist make us money?"

I went through the process of sending demos to dozens of record labels when playing with Cabala, feeling confused when only a few responded with a "got your demo" e-mail, and a couple "we'll give it a listen and will let you know". The band had a decent following, and its sound was better than what was on the radio at the time.

In time I realized that, like our band, thousands of other bands were constantly sending demos to record labels. They have literally piles of submissions waiting to be heard by their scouts. If a demo—typically two or three songs long—requires 15 to 20 minutes to be heard and analyzed, imagine how long it would take to go through hundreds of them every week!

So, unless you are lucky and are "found" by a label, that approach resembles a lottery more than a professional step. I learned this as Cabala was starting to dissolve and went to create Australis with the intention of releasing my music independently.

FM: Any advice for young aspiring musicians?

A: Yeah. Take your dream by the steering wheel and drive it yourself. Don't invest your time and energy trying to find someone to drive it for you. It is your dream. Nobody understands it like you do. Just perfect your skills and prepare to do everything by yourself. Your chances to succeed will be much, much better that way than trying to be chosen by labels that are drowning in the noise made by hundreds of others like you.

FM: Any advice for young aspiring software engineers?

A: Hmmm, I would say to remain a learner forever. There are no plateaus in engineering, and if you feel you've reached one, it should be a warning that you are falling behind compared to the rest of the world. However creative and innovative you are, there's always someone better. Keeping a learning attitude will always push you to be better.■

Fred Mora is a software engineer working in DevOps. His day job and hobbies involve Linux. He lives in Connecticut with his geeky wife and his Linux-using kids.

Send comments or feedback via http://www.linuxjournal.com/contact or to ljeditor@linuxjournal.com.

EOF

DOC SEARLS

Mars Needs Women

Linux is pretty much an all-male project. Let's change that.

Here at *Linux Journal*, nearly 100% of our subscribers are male. So are all of our editors and regular writers, with the single exception of the one in charge. Consistent with that, our publisher and Webmistress are both female. So is our entire ownership. I bring this up because I believe women have leadership advantages that most guys—especially in tech—fail to respect, mostly because we were poorly taught to respect them. Garrison Keillor (**http://en.wikipedia.org/wiki/Garrison_Keillor**) explains this in *The Book of Guys* (**http://www.amazon.com/The-Book-Guys-Garrison-Keillor/dp/0140233725**):

> Here's what they won't tell you in class:
>
> Girls had it better from the beginning, don't kid yourself. They were allowed to play in the house, where the books were and the adults, and boys were sent outdoors like livestock. Boys were noisy and rough, and girls were nice, so they got to stay and we had to go. Boys ran around in the yard with toy guns going *kkshh-kkshh*, fighting wars for made-up reasons and arguing about who was dead, while girls stayed inside and played with dolls, creating complex family groups and learning to solve problems through negotiation and role-playing. Which gender is better equipped, on the whole, to live an adult life, would you guess? ...Is there any doubt about this? Is it even close?
>
> ...Men adore women. Our mothers taught us to. Women do not adore men; women are

Since then, great schemes and failed brilliance have been running non-stop in the technology world, even through bust cycles.

> amused by men, we are a source of chuckles. That's because women are the makers of life, and we aren't. We will never breast-feed. We get more than our share of loot and we are, for some reason, incredibly brave and funny and inventive, and yet our role in procreation basically is to get crazy and howl and spray our seed in all directions.
>
> So we carry adolescence around in our bodies all our lives.

Later he adds this:

> Spectacular dumbness is a guy type of gift. We are good at great schemes and failed brilliance, and some eras seem to encourage this.

He wrote that in 1993, one year before Linux hit v1.0 and *Linux Journal* was born, and two years before the Net as we know it today (graphical browsers, ISPs, Amazon, Craigslist, cookies) came together (**http://www.businessinsider.com/flashback-this-was-the-internet-in-1995-2013-4?op=1**). Since then, great schemes and failed brilliance have been running non-stop in the technology world, even through bust cycles. And, with too few exceptions (for example, *Linux Journal*), guys have run the show.

It's easy to see this as a matter of leadership. In "Silicon Valley Has a Code Name for Sexism & Racism" (**http://blogs.wsj.com/accelerators/2013/10/07/vivek-wadhwa-a-code-name-for-sexism-and-racism**), Vivek Wadhwa (**http://blogs.wsj.com/accelerators/vivek-wadhwa**) says:

> ...with a couple of notable exceptions, women are rarely found in the executive ranks of tech companies. The Valley's echo chamber—what I call the "mafia"—is oblivious to criticism about this. It doesn't seem to care about the imbalance. Note the Twitter IPO filing. It shows that all of its board members are male, as are all of its executives—other than one lawyer

> whom the company added a few weeks ago—and all of its investors.

After digressing into an exchange of insults with Twitter CEO (and former comic) Dick Costolo, Vivek gets down to business:

> This exclusionary behavior is also harmful to companies and their shareholders. To start with, having women on boards produces better outcomes. Research by analyst firm Catalyst shows that companies with the highest proportions of women board directors outperform those with the lowest proportions by 53% (**http://www.catalyst.org/media/companies-more-women-board-directors-experience-higher-financial-performance-according-latest**). They have a 42% higher return on sales and 66% higher return on invested capital. When it comes to entrepreneurship, the advantages of diversity become even clearer.
>
> Firms founded by women are more capital efficient than those founded by men (**http://www.kauffman.org/research-and-policy/sources-of-financing-for-new-technology-firms-a-comparison-by-gender.aspx**). Women-led high-tech startups have lower failure rates (**http://www.gemconsortium.org/docs/download/2409**). Venture-backed companies run by a woman have annual revenues 12% higher than those by men (**http://www.illuminate.com/whitepaper**); and organizations that are the most inclusive of women in top management positions achieve a 35% higher return on equity and 34% higher total return to shareholders.

I don't doubt that *Linux Journal* would be long gone today without women running the magazine. That's not a knock on men (or on our founder, Phil Hughes, who remains a leader in spirit). It's just that women, on the whole, are better at running business, which lives to serve customers. Guys, on the whole, see business as a the grown-up version of what they learned in back yards as boys.

Case in point. A few years back, when Guy Kawasaki was running Garage Technology Ventures (**http://www.garage.com**), he said he liked to vet start-up business plans with women first, because men tend to talk in those plans about how their new company will kill other companies. Women, he said, know

that killing other companies is not what makes a business succeed—or what customers want. If, as John Gray (http://www.marsvenus.com) famously put it, "men are from Mars and women are from Venus" (the title of his bestseller, http://www.amazon.com/Men-Mars-Women-Venus-Understanding/dp/0060574216), our planet needs a mass migration of Venusian immigrants.

Focusing on founders and CEOs isn't a bad thing, but focusing on the tops of pyramids misses the depth and scale of the problem. Mars needs more programmers, more engineers, more scientists, more mathematicians, more hackers of all kinds. When I look in the LKML (https://lkml.org) for lists of contributors to Linux (https://lkml.org/lkml/2013/10/6/148), I tend to see something that looks like our subscriber role: all-male or damn close. We have a long way to go.

This is a known issue, and well-documented on the prescriptive side, starting with Val Henson's "HOWTO Encourage Women in Linux" (http://tldp.org/HOWTO/Encourage-Women-Linux-HOWTO), at The Linux Documentation Project (http://www.tldp.org). From the "About the Author" (http://tldp.org/HOWTO/Encourage-Women-Linux-HOWTO/x28.html#AEN66):

> Val Henson is a Linux kernel developer, an active member of LinuxChix (http://www.linuxchix.org) and female. Her interests include operating systems research, women and computer science, and fine beer. Many other women collaborated with her to produce this HOWTO, including Raven Alder, Suzi Anvin, Poppy Casper, Claudia "Texchanchan" Crowley, Steph Donovan, Joy Goodreau, Telsa Gwynne, Amy Hieter, Hanna Linder, Anna McDonald, Marcia Barret Nice, Miriam Rainsford, Carla Schroder, Jenn Vesperman, Jenny Wu, Megan "Piglet" Zurawicz, Safari and others who choose to remain anonymous.

That the document was last revised on October 29, 2002—more than eleven years ago—speaks volumes. So does the slow pace of posting on the LinuxChix home page/blog and its mailing lists (http://www.linuxchix.org/join-our-email-lists-or-read-archives.html), or at least those I checked.

But I am encouraged to find a LinuxChix post from February 2013 titled "Joseph Reagle on the gender gap in geek culture" (http://www.linuxchix.org/2013/02/26/joseph-reagle-gender-gap-geek-culture.html). Joseph

(**http://reagle.org/joseph**) is a colleague of mine at the Berkman Center (**http://cyber.law.harvard.edu**) and a wise dude who knows what he's talking about. His book, *Good Faith Collaboration: The Culture of Wikipedia* (**http://reagle.org/joseph/2010/gfc**), belongs in the Canon on Collaboration, should there ever be such a thing. The post from last February leverages this text from Jerry Brito's (**http://jerrybrito.com**) "Surprisingly Free" (**http://surprisinglyfree.com/2013/02/26/joseph-reagle-2**):

> According to Reagle, only 1% of the free software community and 9% of Wikipedia editors are female, which he sees as emblematic of structural problems in the geek community. While he does not believe that being a geek or a nerd is in any way synonymous with being a sexist, he concludes that three things that he otherwise loves—geekiness, openness, and the rhetoric and ideology of freedom—are part of the problem inasmuch as they

allow informal cliques to arise, dominate the discussion, and squeeze out minority views. Reagle also comments on a unintentional androcentricity he has observed even amongst free software community heroes, highlighting the ways in which this behavior can be alienating to women and prevents geek culture from growing beyond its traditional base.

Reagle prescribes a 3-step solution to sexism in geek culture: talking about gender; challenging and expanding what it means to be a geek; and not allowing the rhetoric of freedom to be used as an excuse for bad behavior.

So we're taking the first step here. Let's talk about it.■

Doc Searls is Senior Editor of *Linux Journal*. He is also a fellow with the Berkman Center for Internet and Society at Harvard University and the Center for Information Technology and Society at UC Santa Barbara.

Send comments or feedback via http://www.linuxjournal.com/contact or to ljeditor@linuxjournal.com.

Advertiser Index

Thank you as always for supporting our advertisers by buying their products!

ADVERTISER	URL	PAGE #
Drupalize.me	http://www.drupalize.me	109
Emac, Inc.	http://www.emacinc.com	11
EmperorLinux	http://www.emperorlinux.com	29
iXsystems	http://www.ixsystems.com	7
Manage Engine	http://www.manageengine.com	43
OVH	http://www.ovh.com/us/index.xml	17
Silicon Mechanics	http://www.siliconmechanics.com	3
Usenix Conferences	https://www.usenix.org/conferences	2