

Nexus 7 | Raspberry Pi | GIMP 2.8 | Grive
FREE TO SUBSCRIBERS

EPUB, Kindle, Android, iPhone & iPad editions

LINUX JOURNAL™

Since 1994: The Original Magazine of the Linux Community

A Look at the New Features in GIMP 2.8

DECEMBER 2012 | ISSUE 224 | www.linuxjournal.com

BEST *of the* BEST

2012 Readers' Choice



GRIVE
the Open-Source Client for Google Drive

Get Started with **Raspberry Pi**

REVIEWED:
Nexus 7

PLUS:
Troubleshooting a Colocated KVM Instance

WEB DEVELOPMENT
Book Roundup

SCIENTIFIC VISUALIZATION
with NCL

TESTING AND REDIRECTING
stdin, stdout and stderr

December 9–14, 2012
San Diego, CA

Register by November 19th
and SAVE!

www.usenix.org/lisa12

ERTED! CRISIS AVERTED! CRISIS AVER

LISA '12

26th Large Installation System Administration Conference

Strategies, Tools, and Techniques

Keynote Address by Vint Cerf, Google

Join us for 6 days
of practical training
on topics including:

- Virtualization with VMWare
John Arrasjid, Wade Holmes,
David Hill, Ben Lin, and Mahesh
Rajani, VMware

- Using and Migrating to IPv6
Shumon Huque,
University of Pennsylvania

- Puppet
Eric Shamow, Puppet Labs

Plus 3-day Technical Program:

- Invited Talks by industry leaders such as Owen DeLong, Valerie Detweiler, Matt Blaze, and Selena Deckelmann
- Refereed Papers covering key topics: storage and data, monitoring, security and systems management, and tools
- Workshops, Vendor Exhibition, Posters, BoFs, “Hallway Track,” and more!

Dec. 9–14, 2012
San Diego, CA

Sponsored by:



useNIX
THE ADVANCED
COMPUTING SYSTEMS
ASSOCIATION

in cooperation with LOPSA

MISSION CRITICAL



visit us at www.siliconmechanics.com or call us toll free at 888-352-1173

RACKMOUNT SERVERS STORAGE SOLUTIONS HIGH-PERFORMANCE COMPUTING

**"Just because
it's badass,
doesn't mean
it's a game."**

Pierre, our new Operations Manager, is always looking for the right tools to get more work done in less time. That's why he respects NVIDIA® Tesla® GPUs: he sees customers return again and again for more server products featuring hybrid CPU / GPU computing, like the Silicon Mechanics Hyperform HPCg R2504.v3.

We start with your choice of two state-of-the-art processors, for fast, reliable, energy-efficient processing. Then we add four NVIDIA® Tesla® GPUs, to dramatically accelerate parallel processing for applications like ray tracing and finite element analysis. Load it up with DDR3 memory, and you have herculean capabilities and an 80 PLUS Platinum Certified power supply, all in the space of a 4U server.



When you partner with Silicon Mechanics, you get more than stellar technology - you get an Expert like Pierre.

Expert included.

CONTENTS

DECEMBER 2012
ISSUE 224

FEATURES

70 Readers' Choice Awards 2012

The votes are in!

Shawn Powers



INDEPTH

90 Introducing Grive

Simplify your life up in the cloud.

Mehdi Poustchi Amin

98 GIMP 2.8

A look at the many changes in this latest release.

Shashwat Pant

REVIEW

60 Getting the Most from the Nexus 7

Philip Raymond

COLUMNS

30 Reuven M. Lerner's At the Forge

2012 Book Roundup

38 Dave Taylor's Work the Shell

Working with Stdin and Stdout

42 Kyle Rankin's Hack and /

Home, My Backup Data Center

48 Shawn Powers' The Open-Source Classroom

Your First Bite of Raspberry Pi

108 Doc Searls' EOF

Biggest Data

IN EVERY ISSUE

8 Current_Issue.tar.gz

10 Letters

18 UPFRONT

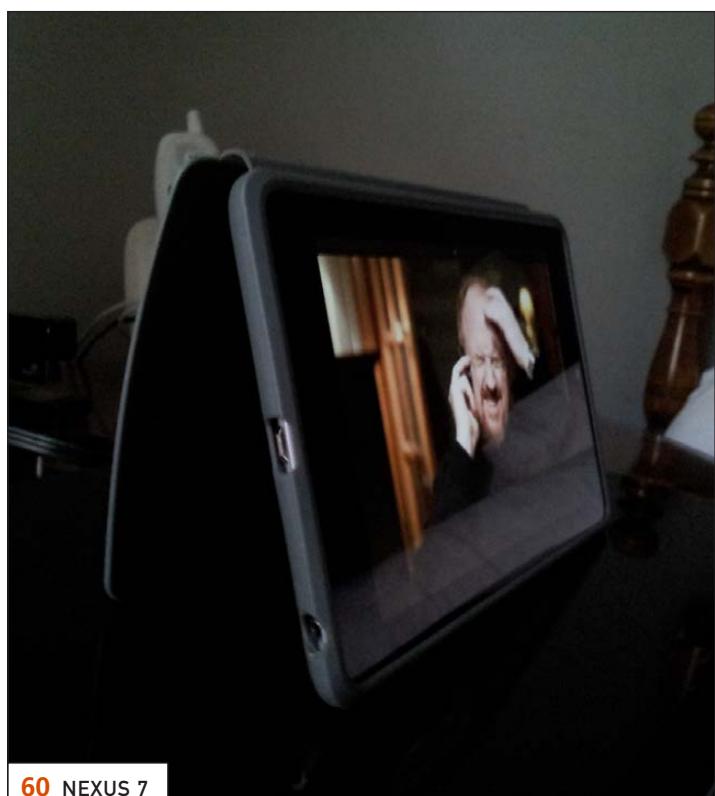
28 Editors' Choice

56 New Products

111 Advertisers Index



48 RASPBERRY PI



60 NEXUS 7

ON THE COVER

- 2012 Readers' Choice, p. 70
- A Look at the New Features in GIMP 2.8, p. 98
- Grive: the Open-Source Client for Google Drive, p. 90
- Get Started with Raspberry Pi, p. 48
- Reviewed: Nexus 7, p. 60
- Troubleshooting a Colocated KVM Instance, p. 42
- Web Development Book Roundup, p. 30
- Scientific Visualization with NCL, p. 22
- Testing and Redirecting—stdin, stdout and stderr, p. 38

LINUX JOURNAL™

**Subscribe to
Linux Journal
Digital Edition
for only
\$2.45 an issue.**



ENJOY:

Timely delivery

Off-line reading

Easy navigation

**Phrase search
and highlighting**

**Ability to save, clip
and share articles**

Embedded videos

**Android & iOS apps,
desktop and
e-Reader versions**

SUBSCRIBE TODAY!

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

Publisher	Carlie Fairchild publisher@linuxjournal.com
------------------	------------------------------------------------

Advertising Sales Manager	Rebecca Cassity rebecca@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

Editorial Advisory Panel

Brad Abram Baillio • Nick Baronian • Hari Boukis • Steve Case
Kalyana Krishna Chadalavada • Brian Conner • Caleb S. Cullen • Keir Davis
Michael Eager • Nick Falty • Dennis Franklin Frey • Alicia Gibb
Victor Gregorio • Philip Jacob • Jay Kruizinga • David A. Lane
Steve Marquez • Dave McAllister • Carson McDonald • Craig Oda
Jeffrey D. Parent • Charnell Pugsley • Thomas Quinlan • Mike Roberts
Kristin Shoemaker • Chris D. Stark • Patrick Swartz • James Walker

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

LINUX is a registered trademark of Linus Torvalds.

TrueNAS™ Storage Appliances

Harness the Cloud

Unified. Scalable. Flexible.

Thanks to the Intel® Xeon® Processor 5600 series and high-performance flash, every TrueNAS Storage appliance delivers the utmost in throughput and IOPS.

As IT infrastructure becomes increasingly virtualized, effective storage has become a critical requirement. iXsystems' TrueNAS Storage appliances offer high-throughput, low-latency backing for popular virtualization programs such as Hyper-V, VMWare®, and Xen®. TrueNAS hybrid storage technology combines memory, NAND flash, and traditional hard disks to dramatically reduce the cost of operating a high performance storage infrastructure. Each TrueNAS appliance can also serve multiple types of clients simultaneously over both iSCSI and NFS, making TrueNAS a flexible solution for your enterprise needs.

For growing businesses that are consolidating infrastructure, the **TrueNAS Pro** is a powerful, flexible entry-level storage appliance. iXsystems also offers the **TrueNAS Enterprise**, which provides increased bandwidth, IOPS and storage capacity for resource-intensive applications.

Call **1-855-GREP-4-IX**, or go to www.ixsystems.com



- ✓ Supports iSCSI and NFS exports simultaneously
- ✓ Compatible with popular Virtualization programs such as Hyper-V, VMWare, and Xen
- ✓ 128-bit ZFS file system with up to triple parity software RAID

TrueNAS Pro Features

- One Six-Core Intel® Xeon® Processor 5600 Series
- High Performance Write Cache
- Up to 480GB MLC SSD Cache
- Up to 220 TB SATA Capacity
- Quad Gigabit Ethernet
- 48GB ECC Memory

TrueNAS Enterprise Features

- Two Six-Core Intel® Xeon® Processors 5600 Series
- Extreme Performance Write Cache
- Up to 1.2TB High Performance ioMemory
- Up to 500TB SATA or 320TB SAS Capacity
- Dual Ten Gigabit Ethernet
- 96GB ECC Memory





SHAWN POWERS

Data, Data Everywhere

When I was younger, I read a lot of books. Although they've fallen out of style, some of my favorite books were of the "Choose Your Own Adventure" variety. The only downside is that three or four nested page-turning decisions into the book, I'd run out of fingers to hold my place. See, I didn't want to commit to the wrong choice. (I may have missed the point of those books.) Based on feedback from last year, I suspect most of you read "Choose Your Own Adventure" books in the same way. After last year's Readers' Choice issue, you wanted more data! This year, we obliged and are giving you the full results, down to tenths of percentage points. (Those writing in for more precise numbers will get *such a pinch!*)

Normally with the Readers' Choice issue, I feign laziness and claim that readers have done all the heavy lifting. This issue, however, is chock full of interesting articles. Reuven M. Lerner starts off with his annual book roundup. I always struggle with which

books are worth my time, and Reuven aims to help with that problem. Dave Taylor, on the other hand, gives a lesson in stdin, stdout and stderr. If you've ever been confused about adding `2>&1` to the end of your cron jobs, Dave will enlighten you.

Kyle Rankin takes us to the depths of system administration with his real-life data-center problems (his own data center). There's no way to learn Linux administration quite like doing it, so follow along with Kyle and his escapades, and be sure to take notes. I follow Kyle with my Open-Source Classroom column and reach for the opposite end of the server spectrum: the Raspberry Pi. Although the RPi can do countless cool things, when it first arrives in its tiny little box, it can be a bit overwhelming. I do my best to make your first taste of Pi a little sweeter and show you some cool things along the way. If you bought a Raspberry Pi, but don't know where to begin, I can hook you up.

If you follow me on Twitter, you

Google still promises that a Linux native client is in the works, but Grive is open source, and it's available now.

know I whine rather regularly about wanting a Nexus 7 tablet. Although the gift-giving fairies at Google apparently don't subscribe to my tweets, Philip Raymond helps a little this month with his review of the Nexus. The downside is that now I want a Nexus 7 even more, and since Google recently has released an updated model with cellular options, I'm hoping Santa Claus reads my tweets!

Speaking of Google, if you're a faithful fan of its products, yet feel abandoned by the lack of a native Google Drive application, Mehdi Poustchi Amin might soothe those wounds with his introduction to Grive. Grive is an open-source implementation of Google Drive, and it aims to bring Google's latest feature to the penguiny masses. Google still promises that a Linux native client is in the works, but Grive is open source, and it's available now.

Although it's a bit of a spoiler, the GIMP has won favorite Graphics/Design Tool in our Readers' Choice survey once again. That's not likely a surprise to anyone who ever has edited a photo

in Linux, but in light of our readers' votes year after year, we've included Shashwat Pant's introduction to GIMP 2.8. This trusty editor is sporting a brand-new UI this season, and looking at its current iteration, a victory for the GIMP in next year's Readers' Choice is looking like a big likelihood as well.

The main feature of this issue is the Readers' Choice results. If you wonder where you line up with the bulk of our readers, or if you're just curious about what companies are currently the most Linux-friendly, you're reading the right issue. We also have product reviews, product announcements and many other helpful, geeky things to make this issue useful and entertaining. To begin your adventure, turn to page 10 now. To reread this article, turn to page 8. To start your adventure anew, turn to page 1....■

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



@steven orr:

They Fixed find

Regarding Steven Orr's letter in the September 2012 issue: a few months ago, I happened

to be looking at the `find` man page and discovered they fixed `-exec`. If you use the old format, it's still broken. However, if you end the command with a `+` instead of an escaped semicolon, it works correctly. They've also added an action `-execdir`, which behaves exactly like `exec`, except it'll `chdir` to the directory the file is in. If you've ended the command `it's exec-ing` with a `+`, it will run one copy of it per directory containing matching files. The man page didn't say what it does about excessive numbers of matches, but I haven't run into any issues yet. I'm pretty sure I've at least come near the pertinent limits, but I'm not certain I've actually crossed them yet.

In other news, if you use `zsh`, and turn on the `extendedglob` option, you can mostly do away with `find`, so long as you're willing to go with some line noise.

That being said, sometimes, I partially agree with Dave Taylor about the legibility issue, so I go with a `find <parameters> | while { read file } { do stuff }` approach. (For those of you who use bash, that's basically `find <parameters> | while read file; do; do stuff; done, except.`) As an old C programmer, I find it more legible, as well as easier to type (`zsh` takes either form).

—Ed Grimm

Archiving Conceptual Software Engineering Designs for Reuse

I work as a software engineer, but I'm studying mechanical engineering. In mechanical engineering, we often represent systems in a simple blackbox input/output diagram. When we break the system down further, we create subsystems that also are interconnected blackboxes with clear definitions of inputs and outputs.

Some mechanical engineering design researchers are trying to represent these systems (which perform functions) in a standardized way so that the conceptual design of the product can be systematically archived. This also allows a computer to try to make new designs by matching up inputs with outputs. For example, let's say the input

to a system is chemical energy and the output is rotational energy. Someone is designing a car. Traditionally, we would input gasoline, have an engine convert it to torque, and output rotational energy. A new innovative engineer thinks that a battery-powered drill has the same basic function as the car—input chemical energy and output rotational energy. So this new innovative engineer designs a battery-powered car.

Okay, that is dumb example, and abstraction is nothing new to software developers, but I hope you see the idea of how we can represent function abstractly with inputs and outputs.

Why not take the same idea and document all the inputs and outputs of all the software in the Open Source world in a very systematic way? Then, we can more easily reuse existing code, and we can have computers help us search for new innovative combinations of existing code. I'm tired of endlessly searching open-source repositories for existing code so that I don't have to re-invent the wheel.

Any thoughts on the idea?

—Anthony

Anthony, I'm not sure I understand where you're going here—it sounds like you're describing reusable libraries? I agree, it's a great way to prevent the re-invention of wheels. Most languages have a fairly good selection of reusable code available, although some have a more robust "catalog" from which to choose.—Ed.

Dave Taylor's Work the Shell

I like the “old-school” coding option. Dave’s scripts are clear, and even a new guy can follow along most of the time. Keep up the good work. I look forward to reading Dave’s next article.
—Lucius Wilson

Spectacular Uses of Linux

I would love to see a story on the uses of Linux in some of the more spectacular applications. I know you’ve run stories like this from time to time, but I’m thinking more of an annual feature that updates us on the use of Linux in supercomputing, major research projects or highly critical applications. I’m prompted to suggest this after learning (or at least, I think this is true) that Linux is at the base of the Space X Falcon 9 avionics system that saved the latest launch (as I write this) when

an engine depressurized and flight trajectory had to be recalculated instantly in real time.

—**Greg Laden**

Greg, I love to read about Linux running the world as well. As we hear about such things and have someone to cover them, we try to mention them. Sometimes, we get the heads-up from Letters to the Editor referencing Space X...—Ed.

Bash Notational Shortcuts, I

I have to wholeheartedly agree with Dave's opinion that clarity is way more important than efficiency for those one-shot scripts. Even if you need the efficiency, it sometimes really is better to switch to another language than to stay in the shell. And then, even though I have a few fairly up-to-date Linux installations, at \$WORK, I also have to administer AIX, Solaris and HP-UX, so the smallest common subset is often mandatory. And, it spares you a lot of interoperability problems and searching man pages and Google for the meaning of some obscure command (and, by the way, how does one search this command on Google?).

—**Klaus**

Bash Notational Shortcuts, II

As Dave Taylor requested comments, here are two remarks: 1) please keep clarity over efficiency, and 2) FORTRAN 77—spaces in variable names? Variable names in FORTRAN consist of 1–6 characters chosen from the letters a–z and the digits 0–9 to my knowledge and experience.

PS. I belong to the generation having heard of APL and having seen APL keyboards.

—**Fred**

Dave Taylor replies: FORTRAN definitely didn't have good variable names, but F77 had arbitrarily long names and spaces were allowed. That's what made it so much more interesting than old FORTRAN—or perhaps I'm getting my old languages confused and I was programming in Ada at the time.

The Android App

As a Linux fan, I (of course) have an Android phone and tablet. I subscribe to the digital edition of *LJ* because I get it immediately after release. But one thing annoys me: whenever I leave the app, I have to log in again with my e-mail address.

That's quite annoying. Otherwise, the content is perfect.

—Klaus Kolle

Klaus, that isn't normal. Something must be going wrong. Perhaps try deleting the app and removing its data, then re-installing it and see if that helps. Sorry it's been acting up on you. I agree, that would be frustrating!—Ed.

Bash Notational Shortcuts, III

In Dave Taylor's September Work the Shell column, he wrote: "Drop me a note and tell me if you would prefer us publishing sample scripts with all of these notational tricks, even at the risk of broad portability across environments and systems, or do you prefer more "standard" old-school scripting techniques that will even work on that clunky old server you administer?"

My short answer: please publish sample scripts "with all of these notational tricks".

My longer answer: as a 30-year veteran of writing shell programs (why demean a program written in a particular language by calling it a "script?") in C shell, Bourne, Korn, Bash and others,

I think it is important to demonstrate clear, concise and efficient solutions to problems. The last is best accomplished in a shell programming language by employing built-ins, so long as they are not obfuscated.

In closing, the first programming language I used (8th grade) was APL. I wrote ray tracing code for optical system design and loved it. However, I agree. Although APL is semantically dense, it is the closest to a write-only language ever invented!

Keep up the good work Dave.

—Andrew Piziali

Android Reader Makes Puppies Cry

I really enjoy your magazine and understand why it went digital. However, the Android reader app, especially on the Nexus 7, brings me no joy. It crashes often, has trouble working over Bluetooth networks on the Nexus 7, and sometimes it gets stuck while downloading, and pages end up missing. It's also quite slow. I have been grabbing the PDFs, but having a central reader as intended would be nice.

Anyhoo, hope it gets fixed soon.

—Kwan

I don't have my Nexus 7 yet (but I'm referring to it as "my Nexus 7", so perhaps I'm one step closer!), but I haven't heard of too many issues with crashing. I'm not sure very many people use Bluetooth networking, so that might be a lesser-tested scenario. Thanks for letting us know. As to the crashing, like I said in my note to Klaus, if you haven't tried removing and re-installing the app, be sure to give that a go.—Ed.

mk802s

Having just bought one of those rinky-dink little ARM-based USB computers, I'd love to see an article on getting Linux running on them (instead of the Android they ship with). Special kudos would be for leading me to images running MythTV. There are some sources and wikis here and there, but a well-written and researched article would be appreciated.

—Bob Hepple

We're open to such an article, but we haven't gotten much in the way of article submissions or projects related to those Android computers yet. I think folks are still Raspberry Pi-crazy! If we come across anything great, we'll be sure to publish it.—Ed.

Bash Notational Shortcuts, IV

I recently read Dave Taylor's column titled "Bash Notational Shortcuts: Efficiency over Clarity" and the letter to the editor that caused him to write the column. I strongly agree with Dave's emphasis on portability and clarity. In an average day, I use Linux, Windows and OS X. On unusual days, you can add in AIX and maybe even one of IBM's non-UNIX systems. Although scripts don't necessarily need to be portable, knowledge does need to be portable, as I don't want to spend the time to acquire a system-admin level of understanding on each release of each OS. I want to spend my time on things that are more important to my specialty.

—Bruce Moore

Change to Digital Format

I initially was unhappy about the change to digital-only, but now I love it and read more of the magazine than I ever did with my print subscription. I read issues more or less cover to cover—something I never did before. I'm using the .epub format on a Barnes & Noble NOOK, so the ads are pretty much unreadable, but reading text examples and even illustrations is just fine. I am surprised to say that I miss the ads—they are helpful for keeping up with what vendors are offering.

I don't necessarily read an issue the moment it arrives, but I do load them on my NOOK regularly. I read them when I have a block of uninterrupted time—like an 11-hour flight home from Europe, when I read the August, September and October issues cover to cover. I never would have considered taking the print editions with me.

I have decided that a 10" tablet is in my future (primarily for reading magazines), and I second the requests for a user survey and perhaps a buyers guide article for tablets.

I also would like an easier way to download an issue. Right now I download it, add it to Calibre and then sync my device to Calibre. My preferred method would be a Calibre plug-in/news recipe that downloads new issues automatically whenever I start Calibre and sync my NOOK.

I would love to have access to back issues in .epub format, but I realize that this is a pretty big challenge and not likely to happen.

Thanks for publishing a great magazine.
—Bruce Moore

I'm happy to see you're enjoying the digital edition. I think there are some serious contenders for "good tablet" now, so perhaps a buyer's guide makes sense. We'll have to see what we can come up with.

As far as getting it on the device, in previous Letters sections, readers sent in a few scripts for automatically sucking it off the e-mail link. If you combine that with Calibre's e-mail feature, I know the process is pretty simple for Kindle for reading as a personal document. I'm not sure if the NOOK has a similar e-mail feature or not, but it's worth checking into. Usually, I just store my .epubs in my Dropbox and locate them with the Dropbox app on my Android device.

Hopefully something will work well for your NOOK. Oh, and for back issues, you do have access to all the digital issues as a subscriber, but indeed they don't go all the way back to the beginning.

Kyle's Raspberry Pi Beer Fridge Suggestion (Part 2)

Just an addition to my letter in the

November 2012 issue. Why not brew your own? BrewPi (<http://brewpi.com>) can help.

—jschiavon

Photo of the Month

My four-month-old granddaughter, Dylan, getting introduced to CrunchBang Linux. She likes to keyboard! She just may be the youngest CrunchBanger ever.

—Vrai Chevalier

Note: please send your Linux-related photos to ljeditor@linuxjournal.com!



Youngest CrunchBanger Ever?

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

LINUX JOURNAL

At Your Service

SUBSCRIPTIONS: *Linux Journal* is available in a variety of digital formats, including PDF, .epub, .mobi and an on-line digital edition, as well as apps for iOS and Android devices. Renewing your subscription, changing your e-mail address for issue delivery, paying your invoice, viewing your account details or other subscription inquiries can be done instantly on-line: <http://www.linuxjournal.com/subs>. E-mail us at subs@linuxjournal.com or reach us via postal mail at *Linux Journal*, PO Box 980985, Houston, TX 77098 USA. Please remember to include your complete name and address when contacting us.

ACCESSING THE DIGITAL ARCHIVE:

Your monthly download notifications will have links to the various formats and to the digital archive. To access the digital archive at any time, log in at <http://www.linuxjournal.com/digital>.

LETTERS TO THE EDITOR: We welcome your letters and encourage you to submit them at <http://www.linuxjournal.com/contact> or mail them to *Linux Journal*, PO Box 980985, Houston, TX 77098 USA. Letters may be edited for space and clarity.

WRITING FOR US: We always are looking for contributed articles, tutorials and real-world stories for the magazine. An author's guide, a list of topics and due dates can be found on-line: <http://www.linuxjournal.com/author>.

FREE e-NEWSLETTERS: *Linux Journal* editors publish newsletters on both a weekly and monthly basis. Receive late-breaking news, technical tips and tricks, an inside look at upcoming issues and links to in-depth stories featured on <http://www.linuxjournal.com>. Subscribe for free today: <http://www.linuxjournal.com/enewsletters>.

ADVERTISING: *Linux Journal* is a great resource for readers and advertisers alike. Request a media kit, view our current editorial calendar and advertising due dates, or learn more about other advertising and marketing opportunities by visiting us on-line: <http://www.linuxjournal.com/advertising>. Contact us directly for further information: ads@linuxjournal.com or +1 713-344-1956 ext. 2.

1&1 DYNAMIC CLOUD SERVER

Our data centers offer top security, Cisco firewall protection and maximum uptime. With more than 20 years experience and an extensive server range, we know what IT professionals need. Get full root access for complete control. We are a strong global company with 3 billion dollars in annual revenue and over 6,000 employees worldwide



LIFETIME DISCOUNT
50% OFF
INCLUDING CONFIGURATIONS,
NO SETUP FEE

1&1 DYNAMIC CLOUD SERVER

A fully flexible server for a range of requirements including applications, databases, gaming and much more!

- Independently configure CPU, RAM, and storage
- Accurate and fair: Control costs with pay-per-configuration and hourly billing
- Up to 6 Cores, 24 GB RAM, 800 GB storage
- 2000 GB of traffic included free
- Parallels® Plesk Panel 11 for unlimited domains, reseller ready
- Up to 99 virtual machines with different configurations under one contract
- No setup fee
- 24/7 phone and e-mail support

\$24.99 per month* ~~\$49.99~~ per month*

✓ MAXIMUM FLEXIBILITY

Independently adjust CPU cores, RAM and hard disk space and add up to 99 virtual machines. We offer cost transparency through hourly billing.

✓ MAXIMUM SECURITY

Redundant storage and mirrored processing units reliably protect your server against any failure

✓ SNAPSHOT

Create a snapshot image of your server configuration

✓ PARALLELS PLESK® PANEL 11

for unlimited domains

✓ INCLUDED TRAFFIC

2000 GB included

II Parallels®
Plesk Panel



1&1



www.1and1.com

*Offer valid for a limited time only. Lifetime 50% off applies to base fee and configurations. Base configuration includes 1 processor core, 1 GB RAM, 100 GB storage. This offer applies to new contracts only. 12 month minimum contract term. Other terms and conditions may apply. Visit www.1and1.com for full promotional offer details. Program and pricing specifications and availability subject to change without notice. 1&1 and the 1&1 logo are trademarks of 1&1 Internet, all other trademarks are the property of their respective owners. © 2012 1&1 Internet. All rights reserved.

diff -u

WHAT'S NEW IN KERNEL DEVELOPMENT

Developers can run into trouble if they don't follow security procedures for submitting code. **Thierry Reding** tried to send the new **PWM** subsystem to **Linus Torvalds**, only to be stopped at the gate. The code had been under development for more than a year; it had been in linux-next for a while already, and Linus thought the code itself looked good.

But, Thierry had not participated in a key-signing event to join the kernel developer web of trust. Without that assurance of his identity, Linus wouldn't accept any code from him. On a more technical note, Linus also wanted to see some "Signed-off-by" tags from people who actually used the PWM subsystem.

The PWM (Pulse Width Modulation) subsystem works on rectangular pulse waves of the sort needed to adjust the power signals to computer fans and display backlights. Thierry's work was intended to replace the old clunky API completely, with its various restrictions and problems, with a nice clean interface.

A lot of developers offered their "Signed-off-by" tags to Thierry's work

after that, including **Arnd Bergmann**, who added that the code was an essential piece needed to build **ARM kernels** for multiple platforms.

Sebastian Andrzej Siewior also met up with Thierry and signed his GPG key, thus indoctrinating Thierry into the web of trust. Linus accepted the code, and kernel development proceeded.

Sometimes it's a good idea to use special-case data values that have a completely different meaning from usual, and sometimes it's not.

Jean Delvare wanted to address a problem with **device ID numbers**. Sometimes, a single device type could have multiple instances on a given system, where the actual device ID didn't really matter and couldn't be predicted. Only one of the devices would have a meaningful ID at any given time.

Jean's idea was to use negative ID numbers for those devices. Then, the system would be able to know right away that those ID numbers were just throw-aways that could be discarded afterward. But, users would see only an ordinary positive ID number that they could use to mount and unmount

the device as usual.

Greg Kroah-Hartman didn't like "overloading" the device IDs in this way. Jean's approach might be more efficient with RAM usage, because it wouldn't be creating a whole new variable to hold the information about which IDs were throw-aways, but it would add complexity and messiness to the kernel.

So, Jean redid his patch, keeping his prized feature of having throw-away device IDs, but using a regular old boolean variable to identify those IDs, instead of encoding it into the sign of the number. Greg accepted the new patch and sent it along to Linus Torvalds for inclusion in the main tree.

Recently, **Ben Hutchings** announced that he'd decided to maintain the **3.2 kernel** as a stable kernel series as a long-term commitment, at least through 2015.

Typically, every kernel that's released by Linus Torvalds has a few follow-up releases from various folks like Greg Kroah-Hartman and others that add stability. But once in a while Greg, or Ben or really anyone who wants to, will pick a kernel and do longer-term maintenance on it.

This time, Ben had decided to support the 3.2 kernel for the rest of the **Debian 7.0** life cycle, which he

estimated would take him through to the end of 2015. After that, he said, he'd either hand maintainership over to someone else who was interested or just let that kernel drop.

Of course, just because he's maintaining the kernel specifically for Debian, doesn't mean the same kernel won't run great on any other system. The 3.2 kernel probably will come to be quite well-respected throughout the world before it's set aside.

—ZACK BROWN

Low Cost Panel PC

PDX-089T

- Vortex86DX 1 GHz Fanless CPU
- Low Power Consumption
- 1 RS232/422/485 serial port
- Mini-PCI Expansion slot
- 2 USB 2.0 Host Ports
- 10/100 BaseT Ethernet & Audio
- PS/2 mouse & keyboard
- CompactFlash & MicroSD card sockets
- Resolution/Colors: 1024 x 600 @ 256K
- Resistive Touch Screen
- Free EMAC OE Linux
- Free Eclipse IDE




2.6 KERNEL

Setting up a Panel PC can be a *Puzzling* experience. However, the PDX-089T comes ready to run with the Operating System installed on flash disk. Apply power and watch the Linux X-Windows desktop user interface appear on the vivid color LCD. Interact with the PDX-089T using the responsive integrated touchscreen. Everything works out of the box, allowing you to concentrate on your application rather than building and configuring device drivers. Just Write-It and Run-It... Starting at \$450 Qty 1.

www.emacinc.com/panel_pc/pdx089.htm

Since 1985
OVER
27
YEARS OF
SINGLE BOARD
SOLUTIONS

EMAC, inc.
EQUIPMENT MONITOR AND CONTROL

Phone: (618) 529-4525 • Fax: (618) 457-0110 • www.emacinc.com

Gaming Like It's 1993

Two things happened in 1993: I attended my first year of college, and I played a lot of *Scorched Earth*. Oddly enough, the latter seemed to have more of an effect on my later career choices.

If you're nostalgic for the good old days of Shareware software on floppy disks, you really might enjoy the open-source *Atomic Tanks*, which is a modern-day clone of the old shoot 'em up tank game. Sticking remarkably close to the original, *Atomic Tanks* looks and plays just like the *Scorched Earth* of old. There are newer twists on the old classic, even some open-source ones like *Scorched3D* (<http://www.scorched3d.co.uk>) with better graphics and more realistic gameplay. I actually prefer *Atomic Tanks*' retro look, however, and if you played *Scorched Earth* from a floppy disk back in the day, you might too. Check it out at <http://atanks.sourceforge.net>.

—SHAWN POWERS



Image from <http://atanks.sourceforge.net>

They Said It

MISGUIDED EDITION

**You have zero privacy anyway.
Get over it.**

—**Scott McNealy**

As the majority of hobbyists must be aware, most of you steal your software.

—**Bill Gates**

**I'm sorry, Dave,
I'm afraid I can't do that.**

—**HAL, from 2001: A Space Odyssey**

It looks like you're writing a letter. Would you like help?

—**Clippy, from Microsoft Office 97**

The Internet is a series of tubes.

—**Senator Ted Stevens**

Android Candy: AirDroid

I love my Samsung Galaxy S2. Unfortunately, like I mentioned last month, it's difficult to access the internal storage. FolderSync was great for managing the internal storage, but there's a lot more to Android than file storage. Enter: AirDroid.

By creating a Web server on your phone or tablet, AirDroid allows you to connect via the browser on your desktop computer. Once connected, AirDroid allows just about every interaction imaginable with your Android device. With AirDroid you can:

- Send SMS messages from your browser.
- Install and remove apps.
- Manage photos, ringtones and music.
- Control and record the Android screen (rooted devices only).

For managing files, I still say FolderSync is hard to beat. For every other sort of interaction with your Android device, AirDroid



is pretty great. Plus, it's completely free! Check it out at <http://www.airdroid.com>.
—SHAWN POWERS

Scientific Visualization with NCL

Many of my previous articles have looked at software packages that do scientific calculations and generate scientific results. But, columns of numbers are nearly impossible to make sense of—at least, by regular human beings. So what can you do? The answer is visualization. We do massive amounts of processing visually, and the easiest way for us to review information is through some graphical format. Because of the power of graphical representations, several packages have been written by different groups. In this article, I'm looking at NCL (the NCAR Command Language, <http://www.ncl.ucar.edu>). NCAR is the National Center for Atmospheric Research, where the Computational and Information Systems Laboratory develops NCL. NCL is an interpreted language designed specifically for data analysis and visualization. It supports several different file formats used in scientific computations, and it also provides several data analysis functions built-in.

NCL is available as both source code and precompiled binaries. These are hosted at the Earth System Grid Web site

(<http://www.earthsystemgrid.org>). The first step is to get an account on the Earth System Grid site. Once your registration is complete, you can download either the source code or the binaries for the system that interests you. The binaries are packaged as tarballs, and they have been compiled for both Debian-based and Red Hat-based systems. You can select between 32-bit and 64-bit binaries. Once you have the correct tarball downloaded, you can unpack it in the directory of your choice. The usual place on UNIX systems is /usr/local, but you are free to place it anywhere on your system.

After unpacking, you need to set several environment variables before actually using NCL. The first is the environment variable `NCARG_ROOT`. You need to set this to the parent directory containing the NCL executables and libraries. For example, if you unpack NCL in /usr/local, you would set `NCARG_ROOT=/usr/local`. You also need to add the location of the NCL executables to your path. In this example, you would add `$NCARG_ROOT/bin` or `/usr/local/bin` to the `PATH` environment variable. You

also need to create a configuration file in your home directory, named .hluresfile. This file will hold configuration options like the default font and the default color table.

Once everything is set up, you quickly can test whether NCL is working correctly by running:

```
ng4ex gsun01n
```

This command copies an NCL script file named gsun01n.ncl into your current working directory and runs it through NCL to produce some graphical output.

If you decide to build NCL from source, you need to install a number of extra libraries to handle all the possible input file formats, as well as all the possible output graphics formats. A full set of instructions is available at http://www.ncl.ucar.edu/Download/build_from_src.shtml.

It is strongly advised that you try to use the prebuilt binaries first, before going through all the work involved in building NCL from scratch.

Now that NCL is installed and ready to use, what can you actually do with it? If you enter the command ncl, you will get a prompt where you can enter commands. Entering the following command tells NCL to start

recording the commands you issue in your current NCL session:

```
record "my_script.ncl"
```

When you are done, type stop record to tell NCL to stop recording your session. At this point, you will have an NCL script that you can reuse later, if you wish.

Once you have your NCL script, you can run it through NCL with the command:

```
ncl <my_script.ncl
```

This is only one way to develop an NCL script. Remember, NCL is a full programming language, and you can write your scripts from scratch, once you know enough of the commands that are available.

The first structure to learn is the format of an NCL script. Scripts all start with the command begin and finish with the command end. Everything else happens between those two statements.

In NCL, four general groups of objects are available. The first group is the workstation objects. These objects represent the graphical devices that act as display devices that the graphical functions use to draw on. These could be an

X11 window, an NCAR Computer Graphics Metafile or a PostScript file. This second group is data objects. Data objects store the actual information used in your analysis and graphical presentation. The third group is the view objects, which represent the elements of your graphical representation. These might be things like text objects, tick marks or contour plots, among others. The fourth is the group of "other" objects, which includes everything else, such as overlays or annotations. You can create new objects by using the commands:

```
objectname = create "object_character_name"
  ↪class_name parent_object
end create
```

where you create a new object based off an existing parent object.

So, what does a basic graphical display look like? You can generate something simple with the following example:

```
begin
x11 = create "x11" xWorkstationClass defaultapp
end create

text = create "text" textItemClass x11
end create
draw(text)
frame(x11)
end
```

Save these commands into a file named sample1.ncl. To run it, you can execute `ncl <sample1.ncl`. The first three lines create an X11 object to draw on. The next two lines create a text object. You need the `draw` command to generate the text object, while the `frame` command instantiates the graphical display on your screen.

What if you want to import data in order to do some processing? NCL can handle many of the file formats used in scientific computational work. As a simple case, let's say you have some code that simply dumps the result as an ASCII representation of the values of interest. You can load this data with the command:

```
file_data = asciread("/full/path/to/file", file_size, "float")
```

This will read in `file_size` numbers from the file given and import them as a float type. Once you've done this, you can grab subsets using indices and assign them to variables. For example, you could create a new array with:

```
array1 = new(64, float)
```

and then assign the first 64 elements from the file with the command:

```
array1(0:63) = file_data(0:63)
```

Graphing is a bit more complex. The first step is to create a data object that will store the values to be graphed. For example, you could use something like this:

```
field1 = create "field1" coordArraysClass defaultapp
  "caYArray": (Y-data)
  "caXArray": (X-data)
end create
```

where you would replace X-data

and Y-data with the actual data arrays you want to use. Once this is done, you actually can do the graphing with:

```
xy_plot = create "xy_plot" xyPlotClass wks
  "xyCoordData": field1
end create
draw(xy_plot)
frame(wks)
```

This creates an `xy_plot` object, based on the data you imported into NCL. Next you need to run the

Read
LINUX JOURNAL
 on your
Android device

Download the app
 now in the
Android
Marketplace



www.linuxjournal.com/android

For more information about advertising opportunities within *Linux Journal* iPhone, iPad and Android apps, contact Rebecca Cassity at +1-713-344-1956 x2 or ads@linuxjournal.com.

draw command and then call the frame command to instantiate the plot.

If you simply want to view the graph, you can call frame on an X11 display object. Or, you can save the graph to a file by calling frame with a file object, like a PostScript file.

There are many possible gotchas when writing and running NCL scripts. Many of these will lead to inefficient code, mostly due to the fact that NCL is an interpreted language. The first item to look at is whether you are using loops unnecessarily. Whenever possible, you should try to use built-in functions to do any processing, rather than try to do the processing yourself. As an example, say you wanted to multiply two 100x100 arrays. If you did it yourself, you probably would write some looping code like this:

```
do i = 0,99
  do j = 0,99
    c(i,j) = a(i,j)*b(i,j)
  end do
end do
```

In NCL, you equivalently could write:

```
c = a*b
```

This is not only shorter, it is also much more efficient. Instead of having to go through each iteration of the loop and ending up incurring the cost of evaluating the statements on each iteration of the loop, the multiplication is evaluated as a single statement by NCL and is handed to the underlying libraries that handle matrix multiplication as a single statement. You also should try to make sure that you pull out as many operations from the loop as possible and apply them a single time after the loop operation is finished. Additionally, sometimes the processing you are trying to do is simply too intensive. In those cases, you can tell NCL to load and use external C- or FORTRAN-compiled objects to handle this processing.

This has been only the barest of introductions to NCL. NCL is a full programming language, with loop structures and conditional statements. You have variables, objects and loads of built-in functions. You can check out all of the possibilities by visiting the NCL and Earth System Grid Web sites. You might be surprised at the analysis that is possible.

—JOEY BERNARD

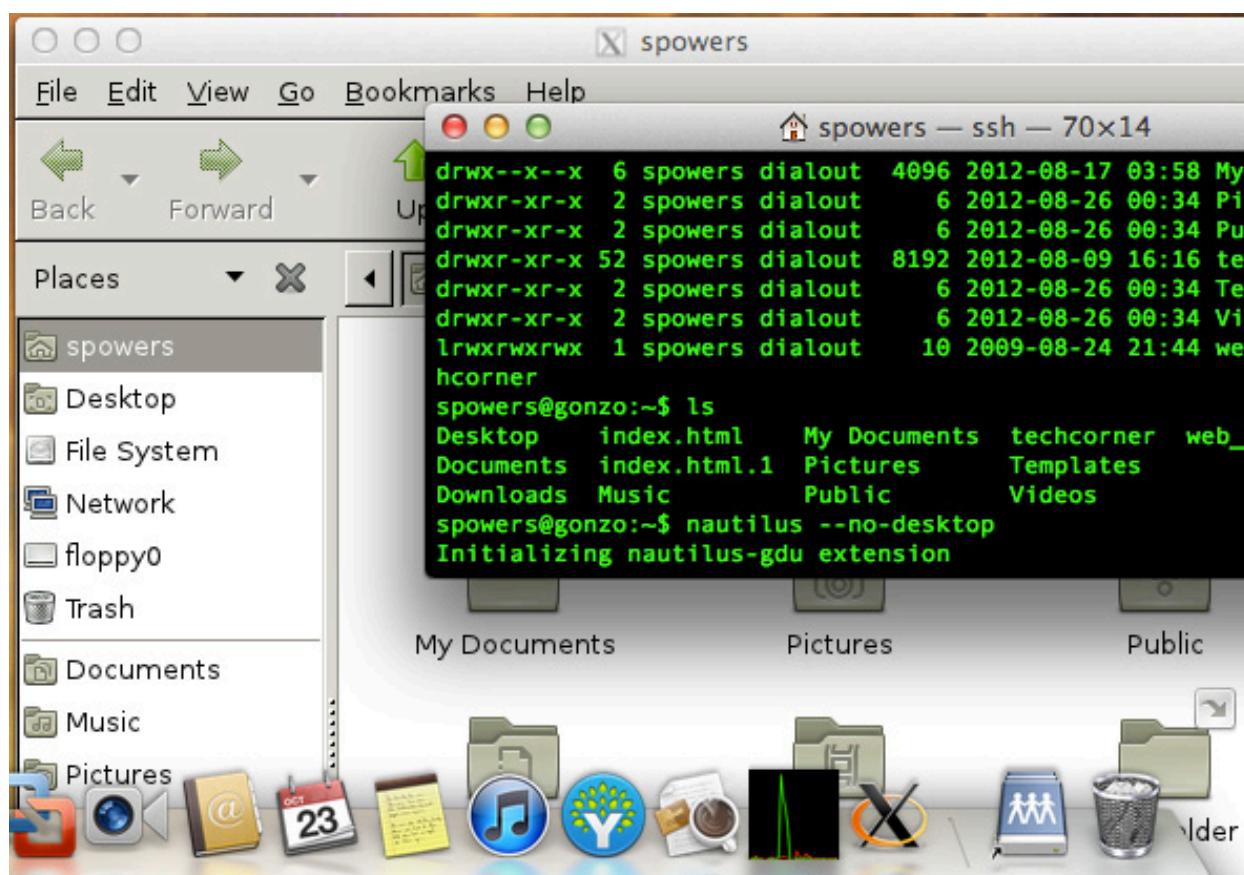
Non-Linux FOSS

Apple has shipped an X11 application with its OS for as many versions as I can remember. With its latest releases, however, the use of XQuartz really has provided a seamless X Window System experience. If you want to use the latest version of XQuartz, you'll have to get it from the Web site (<http://xquartz.macosforge.org>). The one Apple delivers via Software Update isn't as new. Because Apple does include older versions of XQuartz,

it's important to know the Software Update process will write over your newer version if you're not careful.

What does XQuartz provide? For me, its native X11 server means tunneling X11 traffic over an SSH connection can be done right inside the OS X terminal app. With older versions of OS X, you had to start X11 separately, then tunnel in with the non-native-looking Xterm. XQuartz make the process seamless!

—SHAWN POWERS

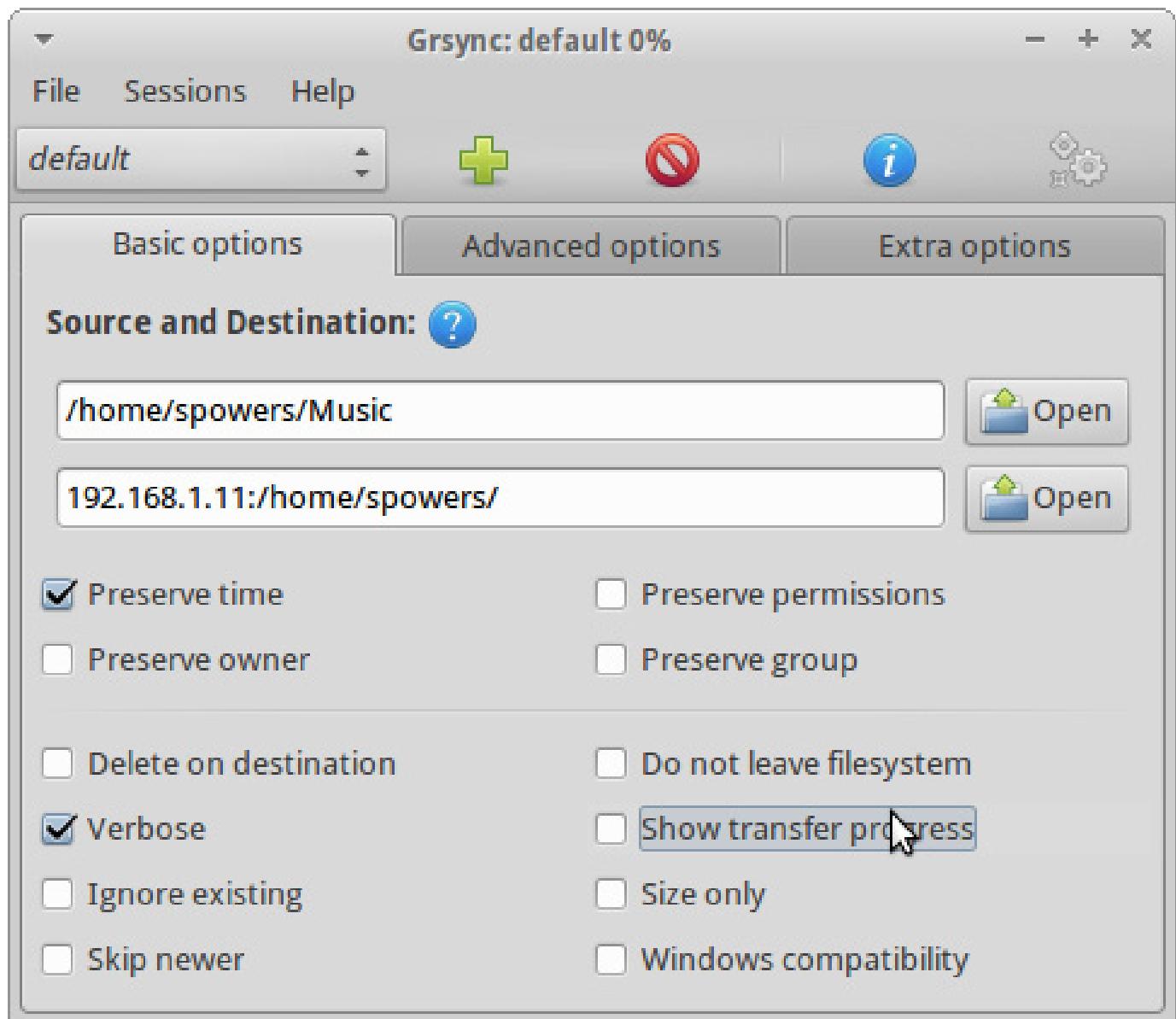


OS X's Terminal app tunneled into a Linux machine, running gnome-terminal, with XQuartz running in the dock.

Rsync, It's GRRRRaphical!

Every year for our Readers' Choice survey, the venerable tool rsync gets votes for favorite backup tool. That never surprises us, because every time I need

to copy a group of files and folders, rsync is the tool I use by default. It really has everything—local folder support,



SSH tunneling support, delta-only synchronization, speed, versatility, and quite frankly, it's just a great program. It has everything—except a GUI.

Don't get me wrong; rsync works great without a GUI. I use it on the command line almost daily. The problem with rsync's amazing power is a rather complex set of arguments. It's possible to learn those flags, but for the neophyte user, they can be overwhelming. That's where Grsync really shines.

Grsync does a great job of turning countless command-line options into

a manageable collection of checkboxes and text-entry areas. When you add the nifty "sessions" feature that remembers settings along with source and destinations, it turns into the perfect filesystem sync tool. If you've ever felt rsync was powerful but too complex to use on a regular basis, I highly recommend Grsync. For making such a powerful tool accessible to the unwashed masses, Grsync gets this month's Editors' Choice Award. Now go copy some files! See <http://grsync.sourceforge.net>.

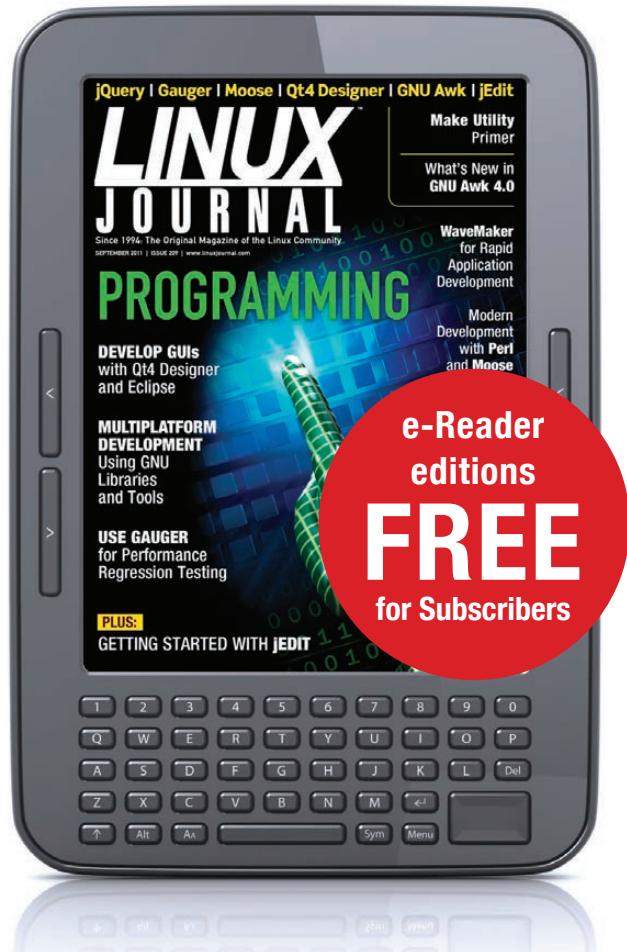
—SHAWN POWERS

LINUX JOURNAL

on your e-Reader

Customized
Kindle and Nook
editions
now available

LEARN MORE





REUVEN M.
LERNER

2012 Book Roundup

What has Reuven been reading? And, what should you be reading? It's time to review this year's crop of technical books.

Once again, I'm dedicating my December 2012 column to a review of the latest books (and other resources) that I've found interesting, useful or just entertaining during the past year.

It continues to amaze me that despite the state of the global economy in general, and of publishing in particular, we continue to enjoy a large number of high-quality technology-related books each year. A growing number of these books are being produced as e-books, rather than printed copies, and many of them are being produced by independent authors and small publishers, rather than the more-established names in the publishing business. By paying attention to blogs, forums and technology podcasts, you can learn about these independent publications, which often are both low in price and high in quality. (Of course, there also are some for which I've paid that turned out to be not nearly as good as I had hoped.)

Ruby

I spend a large part of my time developing software (as opposed to consulting or training); thus, I'm always interested in improving my techniques, learning more about the technologies I use on a day-to-day basis and incorporating the latest ideas into my work. My first-choice programming language continues to be Ruby, along with the Ruby on Rails framework. For that reason, I'm happy to see that the number of resources for Ruby programmers continues to grow. Once again, The Pragmatic Programmers have led the way with a good number of Ruby-related titles.

Several years ago, The Pragmatic Programmers came out with a book called *Rails Recipes*, which contained all sorts of interesting, bite-size ideas, suggestions and practical solutions for Rails developers. This year, The Pragmatic Programmers released *Rails Recipes: Rails 3 Edition*, which brings the book into

the modern era of Rails development. Some of the topics are simple or are well documented elsewhere, such as setting a site's home page, model versioning and sending e-mail with attachments. Other topics were, I thought, descriptions of features that I hadn't seen elsewhere, such as custom collection methods to ActiveRecord results and use of composed_of. This book is useful for Rails developers of all levels.

One of the biggest knocks against Ruby is its relatively slow execution speed, and a second issue is the lack of serious threading. Both of these problems are largely eliminated if you use JRuby, an implementation of Ruby that uses JVM bytecodes. I haven't personally used JRuby on very many projects, but there have been times when I have needed or wanted to do so—and quite frankly, although programming with JRuby isn't very different from the standard Ruby 1.9 interpreter, deploying is another story altogether. I haven't worked with Java on a day-to-day basis for many years, so the array of files, configurations and systems needed to take a JRuby program and make it available to the world is a bit confusing for me.

Fortunately, the book *Deploying with JRuby*, by Joe Kutner and published by The Pragmatic Programmers, aims to fill this gap. It walks you through

the steps necessary to deploy a JRuby application (including Rails applications written in JRuby). The book makes many assumptions regarding how you want to deploy your server, using a virtualized environment and Amazon's EC2. However, this is a common enough pattern that I can understand the reasons for pointing people in this direction, rather than individually configured servers.

Finally, the book *Crafting Rails Applications*, written by Jose Valim and also published by The Pragmatic Programmers, is a rare book aimed at advanced Rails developers. I still haven't gone through it entirely, but Valim presents a number of projects and programs in this book, each of which exercises a different area of Rails that I didn't personally feel like I understood well enough—for example, the "Metal" interface for controllers. Even if I don't use the knowledge gained from this book immediately, my deeper understanding of Rails and how it works undoubtedly will be beneficial in my work, helping me to debug existing applications and structure new ones.

JavaScript

Even the most die-hard Rails developer needs to know at least one other language, namely JavaScript. JavaScript is, by some accounts, the

most widely deployed programming environment in the world, given the incredibly large number of browsers that ship with it, on computers of all shapes, sizes and makes. I still have issues with the JavaScript language, but there's no avoiding its use in the modern Web era, and there are many add-ons and frameworks that make its use more palatable.

I was pleasantly surprised to see that O'Reilly has released a new version of David Flanagan's classic book, *JavaScript: The Definitive Reference*. I have had a copy, physical or virtual, of this book on my shelf since its first edition was released, and although the physical book has become almost unmanageably large at more than 1,000 pages, the content continues to be useful. The book has been updated to include some of the latest trends and libraries in the JavaScript world, including jQuery, HTML5 (including such items as client-side storage and Web sockets) and node.js. It's nice to know that I can continue to rely on Flanagan's clear writing to teach (and just remind) me of aspects of JavaScript that I need to know.

I often tell students in my classes that the problem with programming today is not the development, but rather the maintenance. JavaScript, with its extremely loose structure and lack of built-in systems for modular code,

makes it even easier for problems to occur. Thus, I was happy to see the book *Maintainable JavaScript*, written by Nicholas C. Zakas and published by O'Reilly. Many of the things that Zakas writes are obvious to experienced JavaScript programmers, such as the fact that you can use either single or double quotation marks to define strings.

What is more useful and impressive is his collection of facts about style guides and the implications of your choices—contrasting, for example, the jQuery style guide (which prefers double quotation marks) and the Google style guide (which suggests single quotation marks). I wish this book had included a mention of CommonJS, which offers a standardized system for modularizing and deploying JavaScript, an important aspect of maintainable code. I also see this book as complementary to Crockford's small but excellent book, *JavaScript: The Good Parts*, which was released a few years ago but still contains very useful information.

A new series written by Rails-veteran Noel Rappin and independently published at his site (<http://www.noelrappin.com>) attempts to introduce developers to JavaScript development, including the use of Backbone, CoffeeScript and test-first development (TDD). The series

isn't complete as I write this, but I've enjoyed the parts I have read, and I definitely have learned from it. I look forward to the completion of part 3 and to the forthcoming part 4 as well.

Although I used to laugh at the idea of server-side JavaScript, node.js clearly has become a force to be considered in the server-side world. I'm still not convinced that the node.js paradigm is one that is appropriate for all sites, but there's no doubt that having a single language in the browser and the server can be useful, and that the speed with which server-side JavaScript executes is impressive. O'Reilly's new *Learning Node*, written by Shelley Powers, is a gentle introduction to node.js for people who already know JavaScript, but who haven't yet used it with a server.

Powers covers the basics of installation, moves onto common patterns and paradigms of working with node.js, and then introduces a number of topics that should be familiar to all Web developers, from application frameworks (Express) and templating systems (EJS), to relational and non-relational databases, to testing and SSL. As node.js grows in popularity, it's probably a good idea for all Web developers to gain at least some familiarity with it, and this book provides a great way to gain some experience.

Other Languages

Although it's not quite a language, HTML5 has become quite the superstar in the Web world—and for good reason, as I have discussed in many columns to date. HTML5, although imperfect and still far from being universally delivered, offers the tantalizing prospect of full-fledged applications running inside our browsers. If you're still trying to make sense of the many HTML5 features and how they fit together with your favorite JavaScript library, you should take a look at O'Reilly's *Programming HTML5 Applications*, written by Zach Kessin.

Kessin, like many HTML5 authors before him, discusses a number of HTML5-related technologies, each in its own chapter. So if you want to understand how to work with files, local storage, Web workers or even Web sockets, you won't be surprised to find a chapter on each of those topics. What I found refreshing in this book was the attention to testing the JavaScript portion of HTML5 applications using QUnit and Selenium. The book also looks at how each of these HTML5 features works with existing JavaScript frameworks, so that jQuery and extJS users will understand what they need to do, or what the framework provides, to

ensure smooth usage.

If you already are familiar with HTML5, but (like me) still are learning how to work with mobile devices, O'Reilly's *Head First Mobile Web*, written by Lyza Danger Gardner and Jason Grigsby, might fit the bill. Like other volumes in the "Head First" series, this book aims at deep understanding, rather than teaching you a great many things. So you shouldn't expect to come away from this book an expert in mobile Web application development, but you should come away with a good understanding of how to design a site such that it's appropriate for use with mobile devices.

Although I love Ruby (and Rails), I'm starting to look into a number of other languages, some of which I will explore in these pages during the coming year. One language I definitely want to spend more time learning and working with is Erlang, in which creating new processes is as fundamental, safe and easy as allocating a new string. Zach Kessin's book *Building Web Applications with Erlang*, published by O'Reilly, is an eye-opening introduction to the language and its use as a Web back end for massively scalable applications.

I must also mention R, a language designed for statistical work that I'm

finding is increasingly in demand.

The Art of R Programming, written by Norman Matloff and published by No Starch Press, is a great introduction to R as a programming language, not just for statistical programming. *The R Cookbook*, published by O'Reilly and written by Paul Teator, is notable for the large number of basic and advanced techniques, presented clearly and without too much text or background, for aspiring R hackers.

And although it's not a programming language, I'm starting to understand and use tmux (<http://tmux.sourceforge.net>), an alternative to GNU Screen that is gaining many fans. *tmux: Productive Mouse-Free Development*, published by The Pragmatic Programmers and written by Brian P. Hogan, is short and practical, and really got me to appreciate how great this tool is and how much I need to use it even more in my work.

Databases

Readers of this column know I'm a big fan of PostgreSQL, which not only continues to be an excellent database, but also a great example of a well-run open-source project. However, as a consultant asked to work on a variety of different systems and platforms, and as someone who tries to apply the best

tool to each job, I try to stay abreast of the latest goings-on with other open-source databases, both relational (SQL) and non-relational (NoSQL).

By all standards, MySQL is the most popular open-source database, and one of the main questions people always ask is, "How can I make my queries run faster?" An excellent book on the subject, now in its third edition, is *High-Performance MySQL*, written by Baron Schwartz, Peter Zaitsev and Vadim Tkachenko. It's true that this book focuses on the speed aspects of MySQL—how to design, index and query MySQL in the optimum ways, and then also how to figure out what "the optimum ways" are for your particular database. The chapters on replication (including master-master) and high availability were particularly interesting, both for someone coming from more of a PostgreSQL background and for understanding what tools and techniques are available (and standard) in the MySQL world.

Just as developers constantly should be learning new programming languages, it's also important to learn and work with new databases and storage systems. An interesting entry in this department is *Seven Databases in Seven Weeks* by Eric Redmond and Jim R. Wilson and published

by The Pragmatic Programmers. As the title says, it introduces seven different databases—PostgreSQL, Riak, HBase, MongoDB, CouchDB, Neo4J and Redis—and shows how to work with each of them, broken down across several days per database. If you're interested in the differences (and similarities) between different database products, you almost certainly will enjoy this book.

Podcasts and Screencasts

I continue to be a big fan of podcasts, which I typically listen to while walking between train stations and my clients' offices. I find them to be fun, interesting, up-to-date and plugged in to the technical and other communities I want to follow.

My favorite podcast, hands down, is *Ruby Rogues* (<http://rubyrogues.com>), which I strongly recommend even for people who aren't Ruby developers, because of the excellent and interesting discussions of software engineering, object-oriented programing and entrepreneurship. The podcast keeps me laughing almost as much as it keeps me thinking. Charles Max Wood, who runs the *Ruby Rogues* podcast, has two others to which I subscribe: *JavaScript Jabber* (<http://javascriptjabber.com>) and *Ruby Freelancers* (<http://rubyfreelancers.com>).

Although I enjoy those as well and try to listen to them as they are distributed, they don't seem to pack the punch of *Ruby Rogues*.

A podcast I have discovered only recently, but that I'm mentioning because it covers the high-tech world from a business perspective, is *Mixergy*, Andrew Warner's frequent interviews with entrepreneurs. It's always interesting for me to hear the stories of people who are creating successful businesses and to learn about what they have gone through in order to achieve success. I think it's important for software developers to know about the business, marketing and customer-service side of things; we tend to have a view that the software is the most important part of any business, when it's actually the customer experience and value.

I should also mention the *Economist's Babbage* podcast, which looks at the technology and high-tech worlds. Like *Mixergy*, it covers technology topics from a business perspective, but with the *Economist's* typical interesting insights and humor.

Screencasts continue to be a growing part of the on-line ecosystem, and although I don't watch as many of these as I would like (mostly because I cannot do so while walking to or from the train station),

they can be quite interesting and useful. The weekly *RailsCasts* (<http://railscasts.com>) always are excellent, with Ryan Bates offering a \$9/month "pro" subscription that provides additional screencasts as well.

A relative newcomer, at least to me, is Gary Bernhardt's *Destroy all Software* (<http://www.destroyallsoftware.com/screencasts>), a \$9/month biweekly screencast mainly aimed at Ruby and Git, but with a large smattering of Python and Linux insights as well. I have found these to be quite interesting and useful, as well as humorous and focused.

The latest individual offering a paid monthly subscription to screencasts is Avdi Grimm, with his *RubyTapas* (<http://rubytapas.com>) ten-minute screencasts released three times each week, with the Monday edition being free of charge. Grimm is a well-regarded speaker and writer, and the few samples I've seen of this new offering have been interesting.

Finally, although they're not podcasts or screencasts, weekly e-mail newsletters have become increasingly useful, with links to interesting blog posts, books, releases and trends. You can think of these newsletters as smart, weekly RSS readers in your inbox. Among the newsletters I receive are "Ruby

Weekly" (<http://rubyweekly.com>), "JavaScript Weekly" (<http://javascriptweekly.com>), "StatusCode" (<http://statuscode.org>), "Python Weekly" (<http://www.pythonweekly.com>) and "Freelancing Weekly" (<http://freelancingweekly.com>). I don't really read all of the links they present, but that's sort of the point. There's no way I would have time to find all of these links, let alone read them, on my own.

Other Books

There's not much room left for me to recommend my favorite non-computer books of the year. I was very excited to teach a Ruby class in Beijing in June 2012, and I prepared for that trip in part by reading some books. I already had recommended Peter Hessler's *Country Driving*, so it made sense for me to read (and now recommend) his two other books, *Oracle Bones* and *River Town*, both of which were as fascinating and deep as they were funny. I also read *Postcards from Tomorrow Square* by James Fallows and *Dreaming in Chinese* by Deborah Fallows; I didn't think that they were quite as strong as the Hessler books, but I still enjoyed them a great deal.

I also managed to learn a bit of Mandarin through ChinesePod

(<http://chinesepod.com>), which not only helped me learn some of the language and customs, but also provided my children with hours of entertainment (or torture) as they listened to their father trying to mimic the sounds.

When I was younger, I thoroughly enjoyed Raymond Smullyan's writing, a logician and mathematician who wrote some great books of logic puzzles. I decided to get two of his books for my children to enjoy with me, and although I perhaps wore out my welcome with an overdose, we all enjoyed thinking through the puzzles in *What is the Name of this Book?*

Finally, if you're interested in understanding a bit about how the brain works and allows us to follow such puzzles, *Thinking, Fast and Slow* by Daniel Kahneman has been a pleasure to read, not only because of his insightful descriptions of how people think, but also for his descriptions of many great psychology experiments through the years, the designs of which are as interesting to me as the results they provided. ■

Reuven M. Lerner is a longtime Web developer, consultant and trainer. He is also finishing a PhD in learning sciences at Northwestern University. His latest project, SaveMyWebApp.com, went live this spring. Reuven lives with his wife and children in Modi'in, Israel. You can reach him at reuven@lerner.co.il.



DAVE TAYLOR

Working with Stdin and Stdout

Testing and redirecting stdin, stdout and stderr can be complicated within a shell script. Dave digs in and shows you how to change your redirects even halfway through running a script.

Last month, I erroneously titled my column as "SIGALRM Timers and Stdin Analysis". It turned out that by the time I'd finished writing it, I had spent a lot of time talking about SIGALRM and how to set up timers to avoid scripts that hang forever, but I never actually got to the topic of stdin analysis. Oops.

So this month, let's start with that topic. The behavior to emulate here is something a lot of utilities do without you paying much attention: they behave differently if their input or output is a pipe or file than they do when it's stdin (the keyboard) or stdout (the screen). Try `ls` versus `ls|cat` to see what I mean.

The `test` command has a helpful flag in this regard: `-t`. From the man page:

True if the file whose file descriptor number is `file_descriptor` is open and is associated with a terminal.

Worth knowing is that file descriptor #0 is stdin; #1 is stdout, and #2 is stderr (pronounced "standard in", "standard out" and "standard error", respectively). That's why using `>&` to redirect by file descriptors works with `2>&1` to cause error messages to go to stdout just like regular output messages.

Back to the topic though. In practice, the `-t` test can be used like this:

```
#!/bin/sh
if [ -t 0 ]; then
    echo script running interactively
else
    echo stdin coming from a pipe or file
fi
```

It's easy to test:

```
$ sh inter.sh
script running interactively
$ sh inter.sh < inter.sh
```

Now, what about identifying if the output is an interactive terminal, file or pipe?

```
stdin coming from a pipe or file  
$ cat inter.sh | sh inter.sh  
stdin coming from a pipe or file
```

Perfect. Now, what about identifying if the output is an interactive terminal, file or pipe? It turns out that you can use the same basic test, just replace the file ID 0 with #1:

```
if [ -t 1 ] ; then  
    echo output going to the screen  
else  
    echo output redirected to a file or pipe  
fi
```

The results:

```
$ sh inter.sh  
script running interactively  
output going to the screen  
$ sh inter.sh | cat  
script running interactively  
output redirected to a file or pipe  
$ sh inter.sh > output.txt  
$ cat output.txt  
script running interactively  
output redirected to a file or pipe
```

Pretty cool, actually.

Let's back up a bit and have another look at file redirection before leaving this topic, however.

I already talked about the common trick of 2>&1 to redirect stderr to stdout—something that's very helpful on the command line. You also can redirect specific lines of output in a shell script to stderr, so your error messages are sent to the screen even if stdout is being sent to a pipe or file:

```
echo Error: this is an error message >&2
```

But, what if you want to have your script force stdout to a specific target regardless of what someone does on the command line? It can be done—of course—although it involves a very different approach: the use of the exec command.

At its most basic, the exec call is like a subshell invocation (which is really what happens each time you invoke any system command like ls or fmt), but it's the existing shell that's replaced with the specified command, effectively killing the current process. If you have a shell script that sets up specific parameters

for an external call, for example, you could end it with:

```
exec $cmd $args
```

and anything you might have after that point in the original script is jettisoned because the script is no longer running, it's replaced by \$command.

But exec actually is more nuanced than that, and in particular, a quirk of its behavior gives the solution we seek: exec replaces all the current assignments for stdin, stdout and stderr with those specified as part of the invocation.

So here's the solution, redirecting stdout to a file:

```
exec > output.txt
```

In practice, you can see how it works with this snippet:

```
echo This is stdout
exec > output.txt
echo This is still stdout but goes elsewhere
```

Let's actually put a few different things together in this script so you can see how this all works together:

```
echo this goes to stdout
echo and this goes to stderr >&2
exec > output.txt
```

```
echo This is still stdout but goes elsewhere
echo but where does this go? >&2
exec date
echo this script is kaput
```

Here's what happens when you run the program:

```
$ sh test.sh
this goes to stdout
and this goes to stderr
but where does this go?
```

But, what's actually in output.txt?

```
$ cat output.txt
```

This is still stdout, but it goes elsewhere:

```
Sun Oct 7 10:29:56 MDT 2012
```

Interesting. Notice that, as expected, "this script is kaput" never shows up because once the exec invokes an external program (in this case, date), the script itself is done, because its process has been replaced with the date program.

Notice that the exec redirected only stdout, so that the error message at the very end still goes to the screen. Want to have both stdout and stderr redirected to the file? It's literally a one-character change! Instead of the

Want to have both stdout and stderr redirected to the file? It's literally a one-character change!

above exec redirect, use this:

```
exec &> output.txt
```

That's easy enough, isn't it?

Now, what about the opposite situation where the user has redirected stdout to a file, but you still want it to go to the screen anyway? That's done with yet another sequence on the exec invocation: 1>&2, which redirects stdout to stderr.

Let's look at the same script as above, with exec 1>&2. Here's what happens:

```
$ sh test2.sh > /dev/null
and this goes to stderr
This is still stdout but goes elsewhere
but where does this go?
Sun Oct  7 10:47:44 MDT 2012
```

Pretty cool, eh?

That's it for this month. As always, if you have any interesting scripting projects, challenges or ideas, drop me a note via <http://www.linuxjournal.com/contact>, and I'll have a look. Input always is welcome!

Also, if you have an extraordinary memory, you might recall that Mitch

Frazier wrote about similar topics in *Linux Journal's* Upfront section, during 2010, but his approach was considerably more complicated than mine. Sorry Mitch!■

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>.

New on
**LinuxJournal.com,
the White Paper
Library**



www.linuxjournal.com/whitepapers



KYLE RANKIN

Home, My Backup Data Center

When Kyle's colocated KVM instance won't boot, he has to pull out every troubleshooting trick to get his most valuable server back on-line.

New Linux users often ask me "what is the best way to learn about Linux?" My advice always comes down to this: install and use Linux (any distribution will do but something stable works better), and play around with it. Inevitably, you will break something, and then instead of re-installing, force yourself to fix what you broke. That's my advice, because I've personally learned more about Linux by fixing my own problems than just about any other way. After years of doing this, you start to build confidence in your Linux troubleshooting skills, so that no matter what problem comes your way, you figure if you work at it long enough, you can solve it.

That confidence was put to the test recently when I had a problem with

a KVM host. After a power outage, it refused to boot a virtual machine that was my primary personal server for just about everything. In this article, I walk through a problem that almost had me stumped and show how I was able to find a solution in an unorthodox place (at least for me).

The Setup

Before I dive too deep into my problem, it would help to understand my setup. Although I do have servers at home, my primary server is colocated in a data center. I share the server with a friend, so the physical server simply acts as a secured KVM host, and I split the server's RAM and CPU across two virtual machines 50/50. All of my most important services from my primary DNS server

and e-mail for me and my immediate family, a number of different Web sites and blogs, and even my main Irssi session sits on one of those two VMs. I end up hosting secondary DNS and e-mail from a server on my home connection, but due to a one-megabit upstream connection, I don't host much else at home for the outside world.

One day (while a relative happened to be visiting from out of town), I noticed that both my main server and the physical server that was hosting it were unavailable. I notified my contact at the data center, and it ended up being an accidental power outage that affected my cabinet. I was taking my relative out to the coast for the day, far away from decent cell-phone reception. So, since there wasn't much I could do, I assumed that long before I got back into town that afternoon, power would be restored, and other than losing over a year's uptime, I would be back up and running.

Everything but the Sync

The first time I knew there was a real problem was when I got back into town and my main server still was down. I could log in to the physical host, however; so at first I wasn't too worried. After all, I had seen KVM

instances not recover from a physical host reboot before. In the past, it was either from not setting a VM to start at boot or sometimes even a wayward libvirt apparmor profile that got in the way. Usually once I logged in to the physical host, I could change any bad settings, disable any troublesome apparmor module, then manually launch my VM with virsh. This time was different.

When my VM wouldn't boot manually, I was ready to blame AppArmor. It had blocked VMs from booting in the past, but this time, neither setting the libvirtd AppArmor module to complain mode, disabling all AppArmor modules nor even forcefully stopping AppArmor seemed to help. I even resorted to rebooting the physical host to heed AppArmor's warning that forcibly stopping it after it was running may cause some modules to misbehave. Nothing helped. When I connected a console to the VM as it booted, I started seeing initial kernel errors as though it was having trouble mounting the root filesystem. Great. Did the power outage corrupt my data?

The next step in the troubleshooting process was to attempt to boot from a rescue disk. With KVM, it's relatively easy to add a local ISO image as though it were a CD-ROM.

So after not much effort, I discovered I could, in fact, boot a rescue disk and confirmed from the rescue disk I could mount my VM's drives, and the data did not seem corrupted. So then why wouldn't it boot? After I ran a manual fsck from the rescue disk, I attempted to reload GRUB, and that was when I got my first strange clue about the nature of the problem—even from the rescue disk, I wasn't able to write to the filesystem reliably. I would get virtual ATA resets, even though I seemed to be able to read fine.

So, I assumed I had some level of corruption with that particular VM, but because my data wasn't affected, I figured in the worst case, I could spawn a fresh VM and migrate the data over. So, that's what I tried next using the `ubuntu-vm-builder` wrapper script I used previously to build my VM. The VM seemed to spawn fine; however, once again, even this brand-new VM refused to boot properly and had the same strange disk errors.

It was at this point that my troubleshooting steps start to get a bit hazy, because I started trying more desperate things. I booted different kernel versions in GRUB (after all, the kernel had been updated a few times in the year the server had been up). I audited all of the filesystem permissions on

my VM disk images, and I tried to launch the VMs as root just in case. I even tried converting one VM's disks from qcow2 to raw with no results. Even Web searches came up empty. This server had been down longer than it ever had before, and I was starting to run out of options.

The Sync

My first break came when I decided to copy the VM I had just spawned over to almost identical hardware I had at home with the same distribution and see if I could reproduce the problem there. I picked the new host simply because since qcow2 filesystems grow on demand, it happened to have the smallest disks and was the fastest to sync over. The process was pretty straightforward. First I exported that KVM instance's configuration XML file with `virsh` on the colocated host:

```
$ virsh dumpxml test1.example.net > test1.example.net.xml
```

Then I copied that XML file to my home server, created a local directory named after this VM to store its disk images and synced them over from the physical host:

```
$ mkdir test1.example.net
$ rsync -avx --progress remotehost:/var/lib/libvirt/
➥images/test1.example.net/est1.example.net/
```

Once the disk images were copied, I had to edit the file test1.example.net.xml, because the disk images now were stored in a new location. After I did that, I used virsh again to import this XML configuration file and start the VM:

```
$ virsh define test1.example.net.xml  
$ virsh start test1.example.net
```

The VM actually started! Although I still had no idea what the problem was on the colocated server, I felt pretty confident that if I could sync over my main server, it would run on this home machine. Of course, with a 12Mb-down, 1Mb-up connection at home, it was going to take a bit longer to copy the 45GB disk images for this VM. Other than the time it took, the process was essentially the same as with the test machine, except once the host booted, I had to change its network configuration to reflect its new public IP.

With my server back up and running, I just had to change a number of DNS entries and firewall rules to reflect the new IP, and even with my slower upstream connection at home, I at least had some breathing room to troubleshoot the problem on the colocated server.

The Last Resort

Now that my VM and its data were safe and services were restored (if a bit slow), I felt free to perform more drastic steps on my colocated server. The first step was trying to figure out what was so different about it compared to my home server. They had the same Ubuntu 10.04 server install and most of the same packages. Luckily, I had a number of old cached libvirt and KVM packages on my home server, so at first I iterated through all of those packages to see if the problem was due to some upgrade. Once I exhausted that, I tried different kernel versions on the physical host and still no results.

Believe me when I tell you that during that week I tried every troubleshooting measure I could think of before I finally went to the second-to-last resort. The fact that I was even considering this should tell you how desperate I was getting. The last resort would be to do a complete re-install from scratch—something I wasn't ready to do yet. I was desperate enough though that I went with the second-to-last resort: an in-place distribution upgrade from 10.04 to 12.04. Once the dust settled, I tried my small test image, and it actually worked. We were back in business.

The Sync Back

Well, we were almost back in business. See, I had been using that server at home for a number of days now, and between the e-mail, blogs and other services, it had a lot of new data on it. This meant I couldn't just start up the image that was already on the colocated server. I had to sync up the changes from my home server.

The real trick to this was that I couldn't just sync the server hot. For one, the disk would be changing all the time, and two, I didn't want to risk having the same server running in weird states on two different physical hosts. This meant syncing the actual disk images. The problem was that while the 45GB disk images synced to my house relatively quickly over my 12Mb-downstream (plus the server was already down at the time, so downtime wasn't a consideration), syncing the same data up with my 1Mb upstream was going to take a long time—too long for a pure cold sync to be a solution, as I just couldn't have that much downtime.

The solution here was going to be two-fold, and it was based on a few assumptions I could make:

- Although a fair number of files had changed on my local VM instance,

the actual size of the change was relatively small compared to the size of the disk images.

- rsync has an excellent mechanism for syncing over only the parts of large files that have changed.
- A lot of the changes in my qcow2 files were likely going to be at the end of those files anyway.
- If I use rsync with the `--inplace` option, it will modify the existing disk image on the remote machine directly and save disk space and time.

So my plan for phase 1 was to run rsync from physical host to physical host and sync over the qcow2 disk images hot while the VM was running and tell rsync to sync the disk images in place. Because I could assume the remote images would be somewhat corrupted anyway (that's the downside of syncing a disk image while the disk is being used), I didn't have to care about `--inplace` leaving behind a potentially corrupted file if it were stopped midway through the sync. I could clean it up later.

The advantage of doing the phase 1 rsync hot was that I could get all

of the main differences between the home and colocated images sorted out while the server was still running at home. I even could potentially run that rsync multiple times leading up to phase 2 just to make sure it was as up to date as it could be. Here are the rsync commands I used to perform the phase 1 hot sync:

```
$ rsync -avz --progress --inplace disk0.qcow2
➥remotehost:/var/lib/libvirt/images/www.example.net/disk0.qcow2

$ rsync -avz --progress --inplace disk1.qcow2
➥remotehost:/var/lib/libvirt/images/www.example.net/disk1.qcow2
```

Between rsync's syncing only the bits that changed and the fact that I used -z to compress the data before it was transferred, I was able to sync these files way faster than you would think possible on a 1Mb connection. Of course, these commands ended up saturating my bandwidth at home, so since I wasn't under time pressure for the hot sync to complete, I ended up setting a bandwidth limit of 10 kilobytes per second for the larger disk1.qcow2 image:

```
$ rsync -avz --progress --inplace --bwlimit=10 disk1.qcow2
➥remotehost:/var/lib/libvirt/images/www.example.net/disk1.qcow2
```

Once phase 1 was complete, I

could start with phase 2. I needed the phase 2 rsync to run while the VM was powered off so I could make sure the disk wasn't being written to during the sync. Otherwise, I would risk corruption on the filesystem. Because this required downtime, I picked a proper maintenance window for my server when it would be less busy, finished a final phase 1 hot sync a few hours before, then halted the VM cleanly before I performed the final syncs:

```
$ rsync -avz --progress --inplace disk0.qcow2
➥remotehost:/var/lib/libvirt/images/www.example.net/disk0.qcow2

$ rsync -avz --progress --inplace disk1.qcow2
➥remotehost:/var/lib/libvirt/images/www.example.net/disk1.qcow2
```

Because of the previous work of syncing up the disk images, the final cold sync took only an hour or two with most of the time being spent with rsync seeking between the local and remote image to confirm they were in sync. Once the commands completed, I was able to power up the server again on my colocated host, change its IPs back, and I was back in business. ■

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.



SHAWN POWERS

Your First Bite of Raspberry Pi

Don't be shy, have some RPi.

I **have to** give Kyle Rankin all the credit for my Raspberry Pi collection. I never really felt geeky enough to do anything with an Arduino, and for some reason I mentally lumped the RPi into the same world. Boy was I short-sighted! Thankfully Kyle showed me the light, and I managed to snag some of the new 512MB model B units. You'll be hearing about the Raspberry Pi from Kyle as well, but this month, I want to introduce the RPi to those folks who have been hesitant to buy one, thinking they weren't geeky enough. I had to ask a lot of dumb questions when my Raspberry Pis arrived; hopefully, I can save you that embarrassment.

Batteries Not Included

To be honest, "batteries not included" is true, but the Raspberry

Pi literally can run with four AA batteries and a tiny little resistor. If your goal is to make a portable project, that might be a viable option for you. Just know the RPi needs five volts of consistent power, and it will suck about 700mAh. If you add USB devices, it will drain the batteries even faster. So while batteries will work, it's likely the Raspberry Pi will fail without warning when the batteries get too low. For that reason, it's recommended to use an AC adapter.

Luckily, the RPi is powered by a micro-USB port, which is the most common power adapter for cell phones and tablets. Chances are you have a dozen potential Raspberry Pi power supplies lying around your house. Like with the battery scenario, however, it's important to find a power supply

Plus, if you build a custom case out of LEGO blocks, you can add wings, rockets and LEGO dudes sitting on top.

that can provide at least 700mAh. Remember, when it comes to amperage, more is better, so don't worry about a one- or two-amp power adapter ruining the unit. It will draw only as much as it needs. If the power supply doesn't provide at least 700mAh, however, the Raspberry Pi will not be reliable. Most problems with the RPi tend to be related to a weak or faulty power adapter.

Don't Box Me In

Once you connect power to your RPi, you'll quickly find that it's awkward to handle. I didn't bother talking about a case first, because we all know building a case is not the first thing anyone is going to do. Still, if you want to keep your Pi in good shape, protecting it with some sort of case is important. Thankfully, there are more case options than stars in Hollywood.

Case Option: Buy One You bought the Raspberry Pi, and it was only \$35. It seems only logical to spend an additional \$10–\$15 on a

prebuilt case to house it. There are clear cases, colorful cases, expensive cases and cheap cases. Most of the commercially available cases are plain, square boxes, but the RPi is a little square device, so that makes sense. Buying a case also guarantees a snug fit and proper-fitting connections. If your Pi is going to be handled a lot, a fitted case is a very good idea.

Case Option: Steal from the Nursery The Raspberry Pi is small, and it really lends itself to a LEGO castle motif. Seriously, when it first came out, there were more LEGO-based RPi cases than anything else. If you have access to the right sort of LEGO blocks, it's surprising how sturdy those little suckers can be when properly built. Plus, if you build a custom case out of LEGO blocks, you can add wings, rockets and LEGO dudes sitting on top. Besides, who wouldn't want to have a desktop computer system that can double as a toy space station?

Case Option: 3-D Printing If you have a 3-D printer, I can't



Figure 1. This foldable case is simple, cheap and pretty cool-looking (image from <http://h2database.com>).

think of a cooler way to cover your Pi. Templates are available on the Internet, but even without a template, it's possible to make a case as tight-fitting or as ventilated as you want. Now with the different color options available for 3-D printers, the sky is really the limit.

Case Option: Regular Printing

I didn't buy a case. I don't have a 3-D printer. And, my kids no longer have LEGO blocks lying around for me to steal. For me, I wanted a case that was simple, cheap and did the job. Thankfully, the folks over at <http://h2database.com/raspi> have provided a printable template that folds into a Raspberry Pi case. The



Figure 2. I am probably the only kid on the block with a bootable box of Pop-Tarts.

coolest print stock is a clear mylar sheet. Figure 1 shows how cool a foldable clear case can look.

Case Option: Crazy Go Nuts You have a computer with a silly name, so why not compliment that with a silly case? The Raspberry Pi is small enough to fit inside a Pop-Tart box, an old shoe, a DVD case, a hollowed-out book or a carved Halloween pumpkin. Of course, any case that is wet likely

will damage or destroy your Pi, but it's awful tempting to make a "Raspberry Pumpkin Pi", isn't it? The possibilities are endless with RPi case mods, so let your creativity lead the way.

Getting the Boot

First off, the Raspberry Pi will not PXE boot. It also won't boot directly from a USB device. Both of those things really bum me out,

As the Raspberry Pi doesn't have a BIOS, the only way to allocate RAM is to put a binary file named start.elf into the /boot directory.

but nonetheless, the RPi does boot quite nicely from the included SD slot. Because it won't boot from USB, installing an operating system is easiest from a second computer. Several Linux distributions are compiled for the ARM-based RPi, but the most popular by far is the Debian-based Raspbian OS.

For best performance, a class 10 SD card is recommended. Once you get the SD card, instructions for copying the Raspbian OS are available at <http://www.raspberrypi.org>. The good news is that even if your SD card is small and slow, once the unit is booted, there's no reason the OS can't live largely on a USB drive. If you want to experiment with moving the bulk of the OS off the SD card, the Raspberry Pi Web site has some great tutorials.

Splitting Your Brain

The Raspberry Pi comes with a shared system for system RAM and GPU RAM. Because the GPU is a powerful device capable of driving a high-resolution monitor and hardware rendering

several types of video, if you're using the RPi in a multimedia setting, you'll want to be sure the GPU gets plenty of RAM. If you're running the Pi in a headless server mode, you'll likely want to use as little of the RAM as possible for the GPU, reserving the bulk for the system.

The absolute minimum the GPU needs is 16MB. That doesn't seem like a lot, but because the system maxes out at 256MB or 512MB, even 16MB can be significant. As the Raspberry Pi doesn't have a BIOS, the only way to allocate RAM is to put a binary file named start.elf into the /boot directory. Due to the proprietary nature of the GPU, those various .elf files are downloaded from <http://www.raspberrypi.org>. If you're running headless, you'll want to grab the .elf binary with only 16MB dedicated to GPU. Otherwise, you'll have to decide how much GPU RAM your application will require. Trial and error might be the only way to figure out the best split. The good news is that the new model B units come with 512MB of

RAM instead of 256, so it allows for much more flexibility with the memory split.

To Swap or Not to Swap

SD cards are getting better every year when it comes to wear leveling and write cycles. The concern is always that excessive use of the SD card will wear it out, and it will become read-only. Because the RPi also suffers from limited and non-expandable RAM, the issue of swap space is highly debated.

Raspbian comes installed by default with a swap file active—not a swap partition, but a swap file. Whether this swap file will wear out your SD card is still up for debate. I tend to be paranoid about such things, so I searched for an alternative. Because I have the 512MB units, I have a little more RAM to work with, so I decided to use zRam to compress some of the system RAM and use it as swap space. How does that work? Basically, zRam allows the system to use RAM as swap space, but in a compressed mode that allows it to appear as more RAM+swap than the system actually contains.

zRam already is compiled into the Raspbian kernel, so if you want to try my method, it just takes a little

scripting. I absolutely love the script I found at the eXtremeSHOK.com Web site. That site is off-line at the time of this writing, so I've included the script below. Create a file called /etc/init.d/zram, which contains the following:

```
#!/bin/bash
### BEGIN INIT INFO
#Provides: zram
#Required-Start:
#Required-Stop:
#Default-Start: 2 3 4 5
#Default-Stop: 0 1 6
#Short-Description: Increased Performance In Linux With zRam
#(Virtual Swap Compressed in RAM)
#Description: Adapted for Raspbian (Raspberry pi) by eXtremeSHOK.com
#using https://raw.github.com/gionn/etc/master/init.d/zram
### END INIT INFO

start() {
    mem_total_kb=$(grep MemTotal /proc/meminfo | grep -E
        ->--only-matching '[[[:digit:]]+]')

    modprobe zram

    sleep 1
    #only using 50% of system memory, comment the line
    #below to use 100% of system memory
    mem_total_kb=$((mem_total_kb/2))

    echo $((mem_total_kb * 1024)) > /sys/block/zram0/disksize
```

```

mkswap /dev/zram0

swapon -p 100 /dev/zram0
}

stop() {
    swapoff /dev/zram0
    sleep 1
    rmmod zram
}

case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        stop
        sleep 3
        start
        ;;
    *)
        echo "Usage: $0 {start|stop|restart}"
        RETVAL=1
esac

```

Then you need to make the init script executable and force it to start automatically. The following commands accomplish that:

```

sudo chmod +x /etc/init.d/zram
sudo update-rc.d zram defaults

```

Basically, as written, the script will use half the available RAM as compressed swap space. Again, all credit goes to the folks at eXtremeSHOK.com for creating the init script. I've been very happy with the performance, and it enabled me to get rid of the swap file altogether.

What's Next?

Hopefully this intro to the Raspberry Pi will make the device a little less intimidating for folks like me who were hesitant. With its ability to run XBMC, function as a complete desktop system and yet barely sip electricity, the RPi is a great device for countless projects. And at \$35, it's absolutely dirt cheap.

We expect to see many, many Raspberry Pi projects around the *Linux Journal* offices. Hopefully you'll see a few around your place as well. If you have any particularly cool projects or case mods, send a photo to ljeditor@linuxjournal.com. We'd love to see what you're all up to. ■

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.

9

THIS TIME
WE'RE TURNING IT UP
TO 11!

The 11th Annual
Southern California Linux Expo

Featuring Performances by:

- * Matthew Garrett
- * 70 other Open Source Acts!

With Special Events by:

- * PostgreSQL and MySQL Community Days
- * DevOps Day LA
- * UbuCon
- * SCALE U
- and more...



SCALE 11x

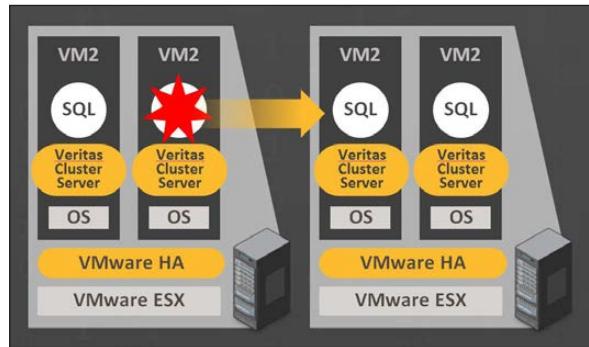
February 22-24, 2013
Hilton Hotel @ LAX
Los Angeles, CA

<http://www.socallinuxexpo.org>
Use Promo Code LJ11X for a 30%
discount on admission to SCALE

Symantec's Veritas Cluster Server

The beauty of the upgraded Veritas Cluster Server, says its maker Symantec, is this: the pinnacle level of high availability and disaster recovery that has been available in the physical world now applies to the virtual one. Not only can applications running in virtual machines be monitored, but they also can be recovered when an incident occurs. Virtual machine reboots, thus, are eliminated, keeping downtime to a minimum, allowing all users to do business even in the face of human error, system failure or natural disaster. Veritas allows users to choose how they want to architect their environment—that is, fail over from V to V, V to P or P to V. Downtime from patching, server maintenance and upgrades is eliminated, because applications can be moved between primary and secondary nodes with no disruption in business. The new version of Veritas Cluster Server works with existing VMware tools like vMotion, DRS and SRM and integrates with vCenter, such that users can continue to use the features of the native stack without compromise.

<http://www.symantec.com/cluster-server>



Dell KACE K1000 Management Appliance

Resource-strapped IT teams face myriad challenges, including endpoint security, regulatory compliance, server management and BYOD (Bring Your Own Device) issues. In an effort to face down these challenges, Dell beefed up the Dell KACE K1000 Management Appliance, which the firm describes as "an easy-to-use, comprehensive and affordable appliance that fulfills all of the systems management needs of organizations of all sizes". Functions include initial inventory and discovery, software distribution, configuration management, patching, security vulnerability remediation, asset management, service desk and reporting. This latest v.5.4 of the K1000 adds features related to endpoint security and regulatory compliance, which are important to the financial services and healthcare industries, among others. Additionally, this version expands visibility and audit-trail capabilities made possible by the new machine inventory API. But most important, say the folks in Round Rock, the new appliance extends operating system support to Windows 8, Mac OS X Mountain Lion, Red Hat 6, Ubuntu and SUSE Linux Enterprise Server 11 SP2—a critical piece of the puzzle for IT admins for solving BYOD issues.

<http://www.dell.com/KACE>



Onset HOBO Wireless Temp/RH Monitoring Kit

Adding measurement points to building automation systems can be expensive, and chart recorders tend to require a lot of maintenance. To overcome these limitations, Onset developed

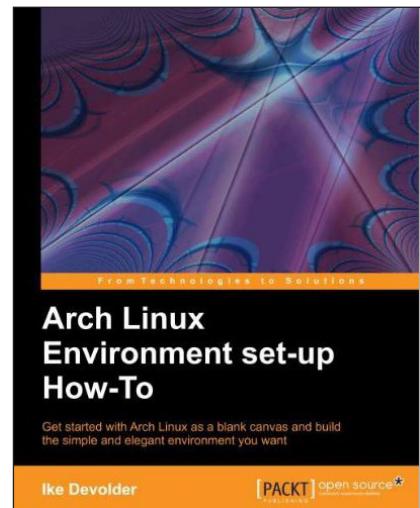
the HOBO Wireless Temp/RH Monitoring Kit—an easy-to-use, wireless system that includes everything needed to make environmental monitoring fast, cost-effective and convenient right out of the box. In just three steps, users can monitor critical temperature and humidity conditions, log data for trend analysis and stay notified of alarm conditions via text or e-mail—all without the hassles of wires or manual data offload. The monitoring kit is appropriate for applications in a broad range of facilities, including food processing plants, office buildings, laboratories and warehouses. The wireless kit includes HOBOOnode Manager software, which allows users to view near real-time energy and environmental data, set alarm notifications and get an at-a-glance view of the system.

<http://www.onsetcomp.com>

Ike Devolder's Arch Linux Environment Set-up How-To (Packt Publishing)

Arch Linux Environment Set-up How-To by Ike Devolder is a new book for people wanting to dig deeply into a Linux system. Through the years, myriad Linux distributions have sprung up, with some trying to do everything for the user, others less. Arch Linux seeks to be easy and user-friendly from the standpoint of developers and enthusiasts who want to customize their system to a maximum degree. Devolder's book aims to get the new Arch Linux user up to speed, and in true DIY style, it leaves the true experience open-ended. After covering installation and package management, Devolder develops the full Arch Linux environment step by step in a task + recipe format. The recipe as a solution consists of a carefully organized set of instructions to perform the task as efficiently as possible and a discussion on how to apply the solution in different situations.

<http://www.packtpub.com>



Attunity Replicate for Exadata

Readers working with Big Data for enterprise BI and analytics will be pleased to learn about Attunity Replicate for Exadata, a new high-performance data replication solution designed to enable quick loading of Big Data into and out of Oracle Exadata. Attunity's solution is designed to empower Oracle customers to load data from heterogeneous data sources quickly and maintain the changed data continuously and efficiently. Core features include high-performance full loads and continuous change data capture, support for a wide range of data sources, a drag-and-drop GUI for designing and monitoring replication tasks, automatic schema generation and implementation of metadata changes on the target, and transparent data type transformations between source and target.

<http://www.attunity.com>

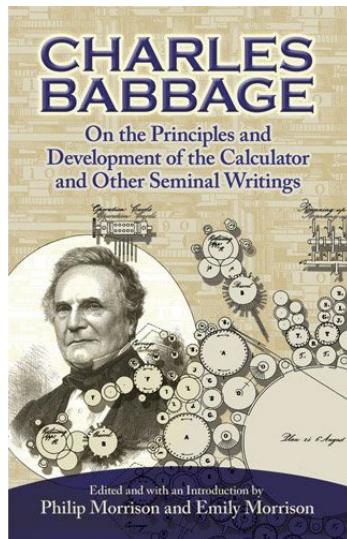


BeyondTrust's Retina Community



Today's landscape of security threats is changing as dynamically as IT environments, resulting in a prime breeding ground for vulnerabilities. To better assist organizations in dealing with the barrage of exploits and targeted attacks, BeyondTrust has upgraded Retina Community, a free vulnerability assessment tool based upon the network security scanner Retina Network. The solution helps to identify potential weaknesses and help prioritize which need to be addressed first. This new edition of Retina Community includes scanning profiles for security and compliance. IT professionals responsible for critical areas, such as servers, desktops, virtualization and databases can assess their focus area for vulnerabilities quickly and easily. Furthermore, the free license has increased to support the assessment of up to 256 assets, further reducing the attack surface in companies large and small.

<http://www.beyondtrust.com>



Charles Babbage's *On the Principles and Development of the Calculator and Other Seminal Writings* (Dover Publications)

Eighteenth-century British “Renaissance man” Charles Babbage developed the idea of a programmable computer and articulated the principles on which all modern computing machines are based. To keep Babbage’s seminal ideas from disappearing, Dover Publications is updating the 1961 release of *On the Principles and Development of the Calculator and Other Seminal Writings*,

a compilation of his writings and those of several of his contemporaries. The writings illuminate the work that eventually led to the development of the calculator. Interestingly, Babbage’s Difference Engine No. 2 from the late 1800s was able to compute the values of polynomial functions with results up to 31 digits. The ideas found in Babbage’s writings are essential reading for anyone interested in the history of computing technology.

<http://store.doverpublications.com>

Perforce's Git Fusion

Git is great, but now it's got more with help from Perforce's Git Fusion. With Git Fusion, developers who use Git can continue to use their preferred tools, but they now have flexible new capabilities for customization, reuse and sharing of projects. Teams can use Git repositories and their components in new ways, combining them and creating new repositories by selectively reusing elements from other Git or Perforce code lines. Git Fusion also extends Perforce's enterprise version management capabilities to Git repositories, bringing added IP security and defensibility, scalability, compliance and visibility to disconnected projects. Perforce and Git Fusion are free for up to 20 users.

<http://www.perforce.com>



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.

REVIEW

HARDWARE

Getting the Most from the Nexus 7

A look at the king of the 7" tablets. **PHILIP RAYMOND**

The ASUS/Google Nexus 7

arrived at my door on August 1, 2012 with a lot of anticipation from both me and the rest of the consumer electronics world. A quad-core Tegra 3 processor from NVIDIA, a 1200x800 HD IPS display covered with the latest scratch-resistant Corning Gorilla Glass and a 1.2 megapixel front-facing camera were among its most notable cool features. Fast-forward to late September 2012, and although it has gotten mostly positive reviews, it is viewed primarily as a consumption device, with a lack of ways to use it for producing anything short of e-mail messages. I believe this review will change your mind and show you how to use it for both production and consumption in ways you didn't know. The way to do this is with the right app—better known



I tried six different Android video editing-apps, some free some paid, and five out of six were horrible.

as the right tool for the right job.

Let's start with my favorite topic, producing videos. As long as your need is only for production of short-form, nothing-fancy videos, the Nexus 7 can do it. Yes, it has only a front-facing camera. However, I was surprised to discover I had good results when I held it in the general direction of the action, without the

aid of a screen to see what was captured. I used the app Camera ICS+, the plus being the pay-for \$.99 version that captures 720p HD video from the Nexus 7. It also can be used for shooting high-quality still photos as well.

Next, for editing, check out the app called Andromedia. I tried six different Android video editing-apps, some free

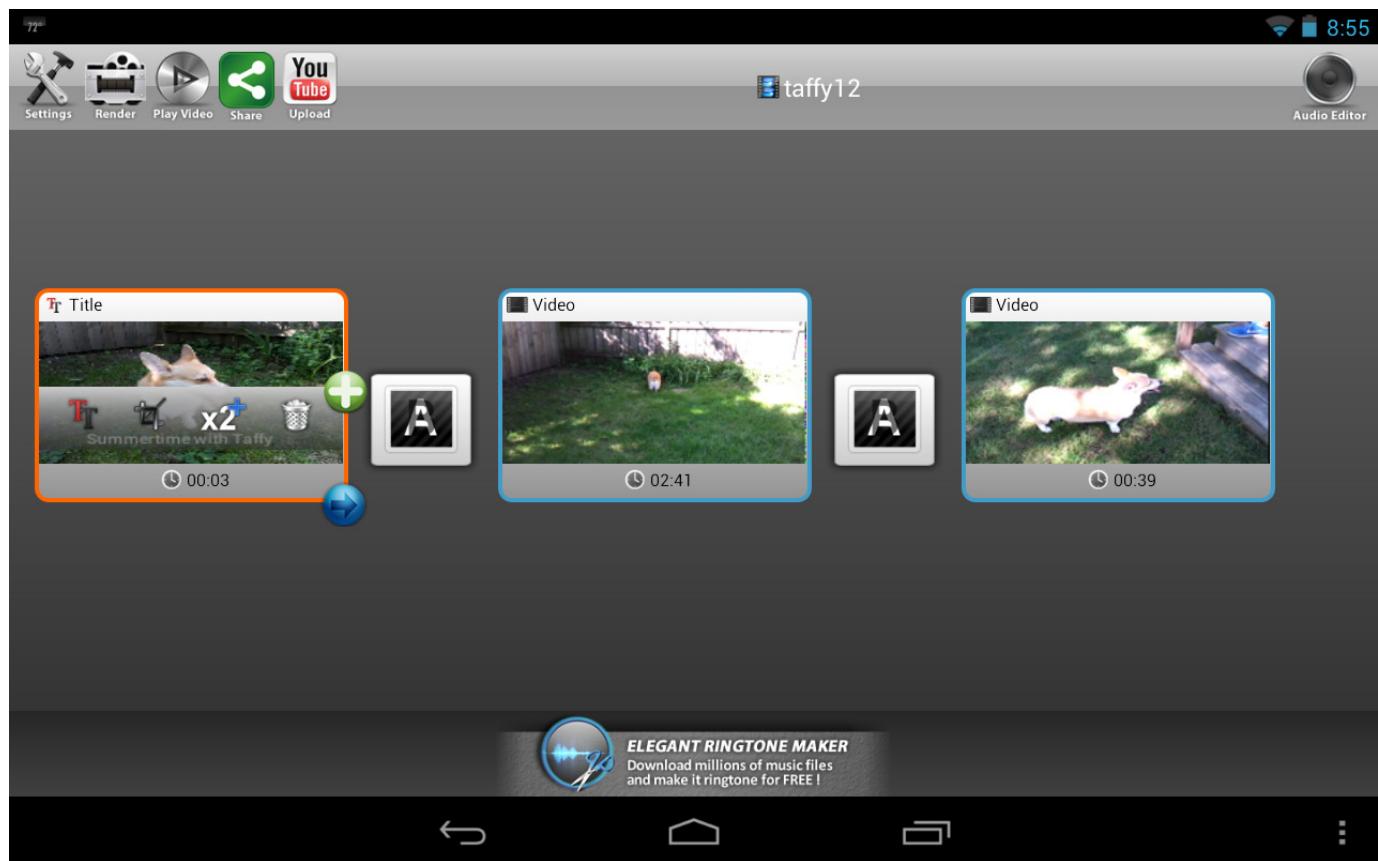


Figure 1. Storyboard Layout

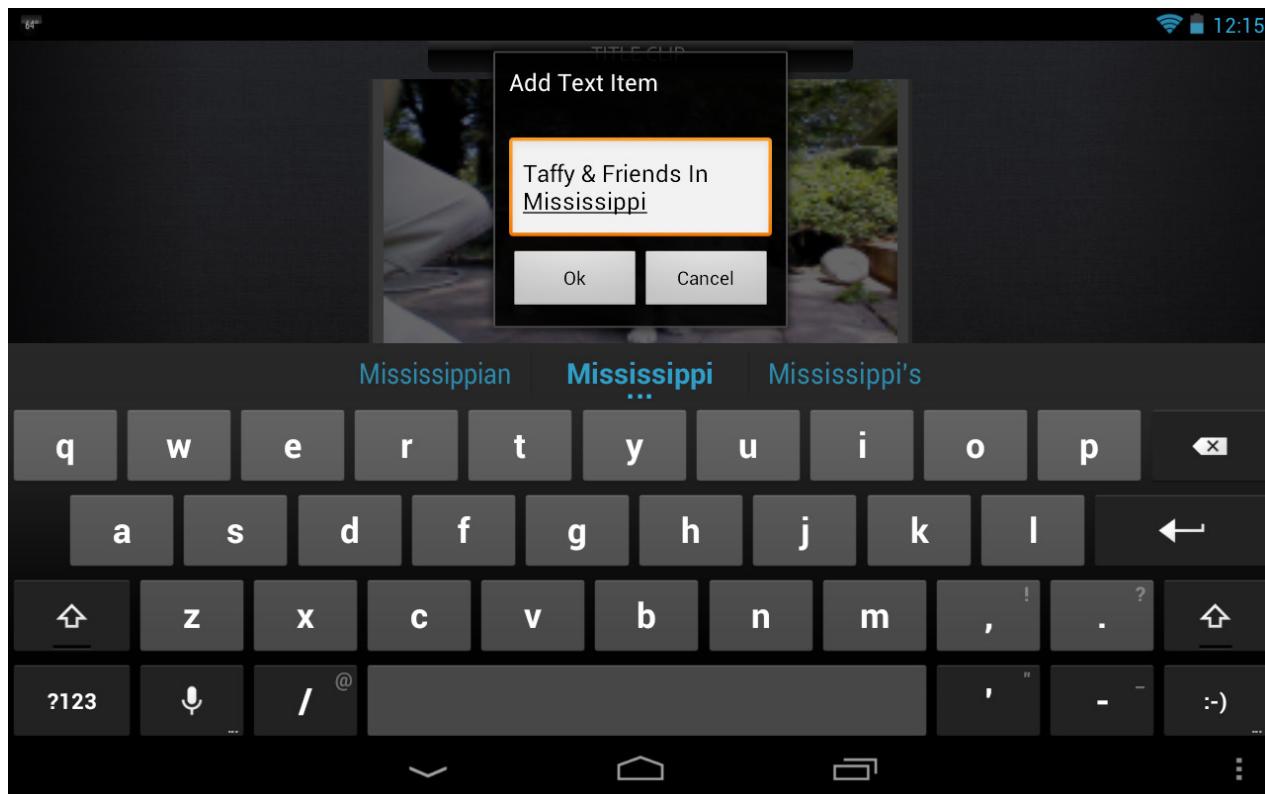


Figure 2. Adding a Title



Figure 3. Choosing the Title's Look



Figure 4. Selecting a Transition

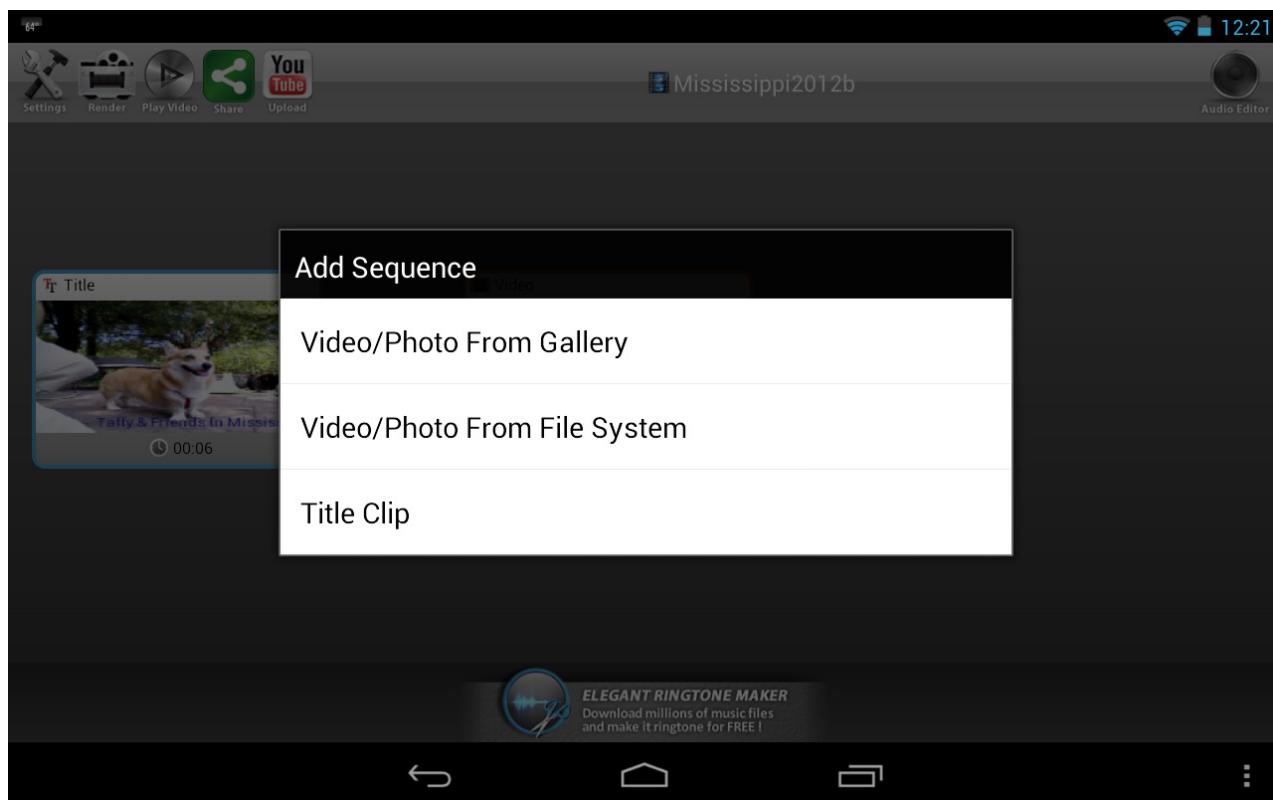


Figure 5. Selecting Where the Next Sequence Comes from



Figure 6. Export Settings

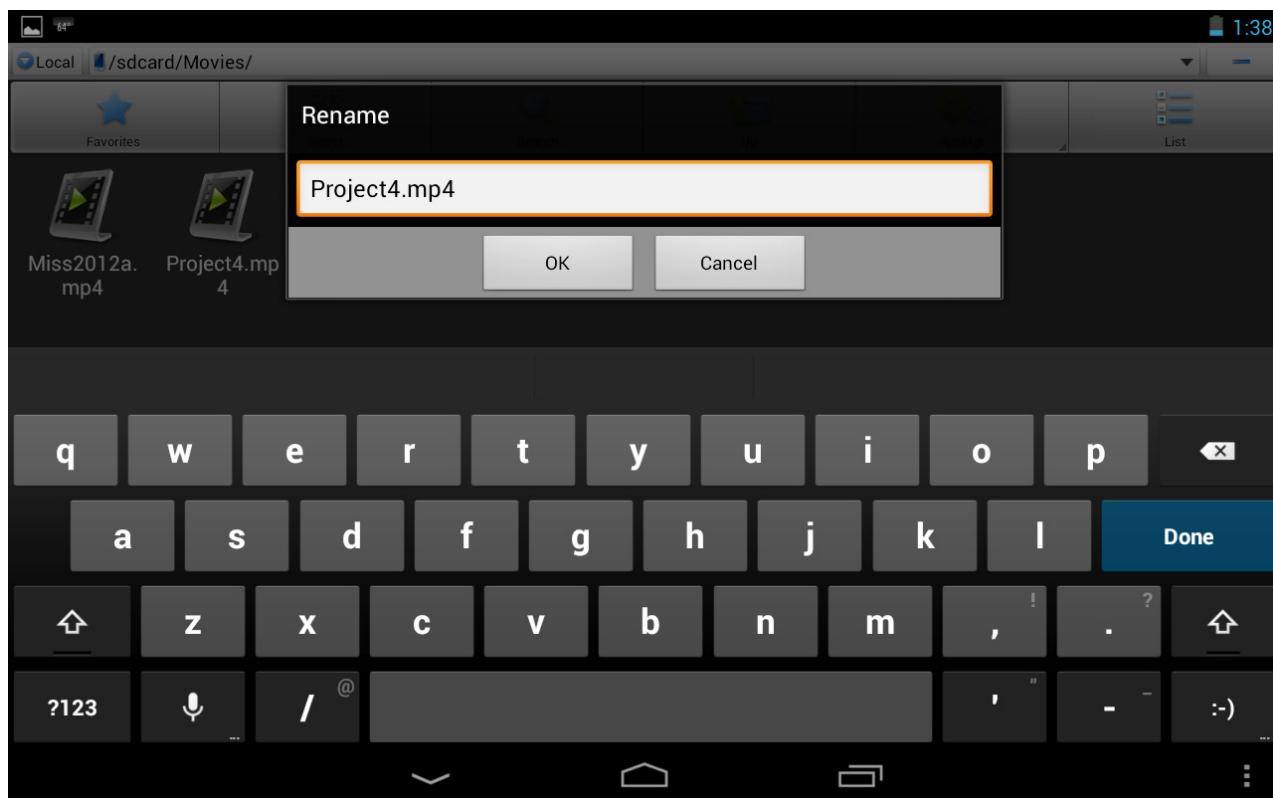


Figure 7. Naming the Project

some paid, and five out of six were horrible. Andromedia was the only one that worked as advertised, and it was free! It isn't perfect though. Rendering a six-minute HD video took nearly 45 minutes, but it does work. It has all of the basic editing tools you'd expect and some you might not. It edits and exports 720p HD video; it can super titles you create over either a color background or a picture; it has several choices of scene transitions, and it even allows importing and editing of audio, if you want to add music or sound effects. It's also easy to use. Because it is a free app, there are small ads at the bottom of the GUI, but I didn't find them to be annoying or intrusive. The screenshots in Figures 1–7 show the editing process.

Speaking of screenshots, on the Nexus 7, you can make them by holding down the power button and the volume button at the same time for about two seconds, which is followed by a nondescript sound.

Another very useful app is called ES File Explorer. This is the best way to handle the storage limitations on the Nexus 7, and it doesn't involve a cloud. This app lets you transfer to any sharable device with a hard drive on your home network. For me, that would be my 2TB network hard drive.

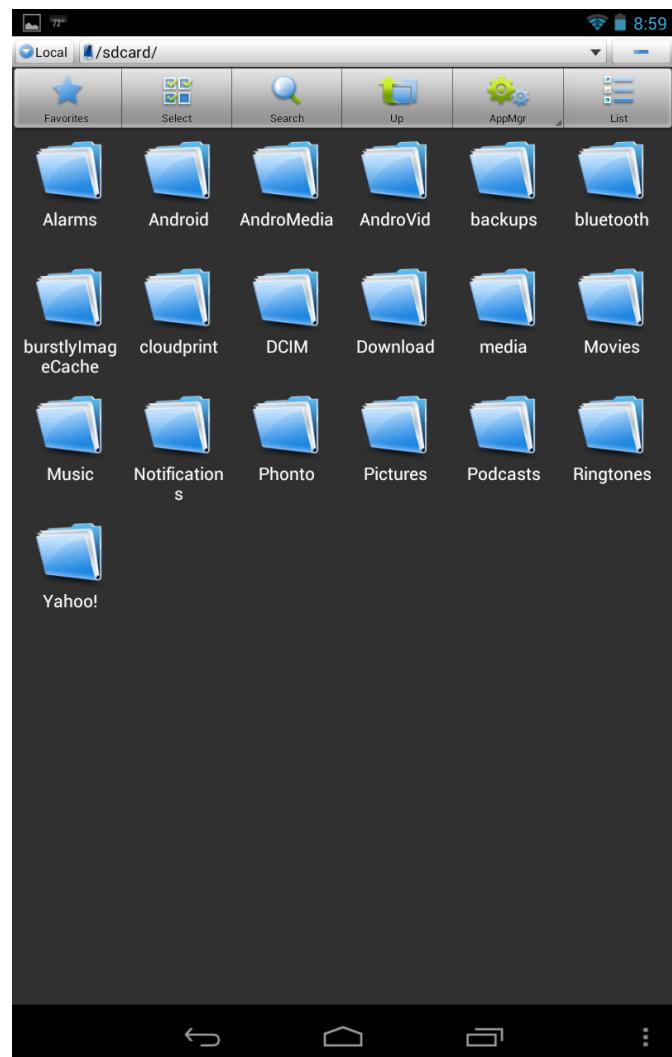


Figure 8. Choosing a Folder Using the ES File Explorer

Basic photo editing is achieved by choosing a photo in the gallery and touching the choice icon in the upper-right corner.

If you need to print, any of the Android apps for the major printer manufacturers will work, but it needs to be a network printer. Epson Iprint for Android worked for me and my Epson Workforce 630. It also will print

You can transfer files to and from the Nexus 7 and your smartphone via Bluetooth as well.

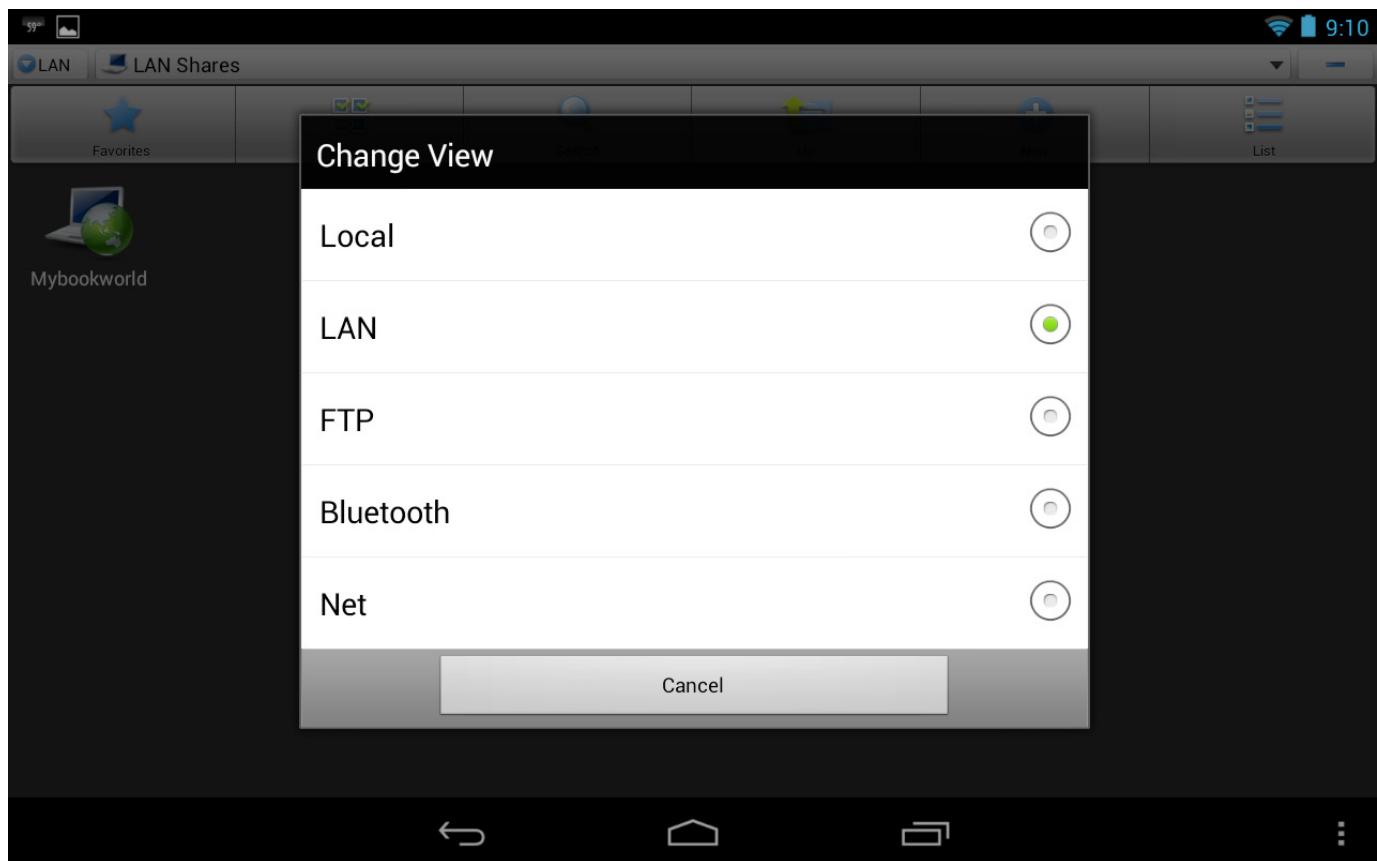


Figure 9. Choice of Views

documents and pictures stored on clouds, like Google Drive.

The Nexus 7 is a Wi-Fi-only device; however, it is possible to connect to the Internet by tethering the Nexus 7 to your Android phone via Bluetooth by pairing them and checking the box on the phone to allow Internet tethering. My experience with doing this is acceptable, but the Bluetooth Internet speed is knocked down to

about 400kbps—even when the phone is getting 4G service that's more than triple that speed. That said, it will get you on the Net when Wi-Fi isn't available. In fact, any smartphone that allows Bluetooth tethering to share the phone's Internet connection will work with the Nexus 7. The one step you need to do to make this work is to activate the Bluetooth visibility from both the Nexus 7 and the phone,

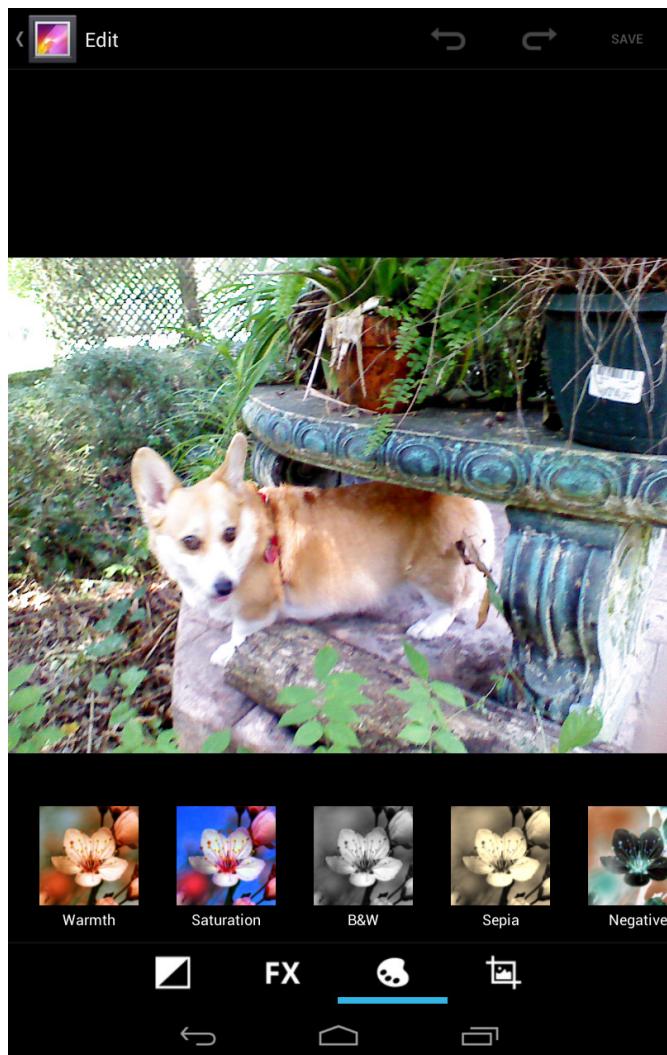


Figure 10. Picture Editing on the Nexus 7

prior to pairing.

You can transfer files to and from the Nexus 7 and your smartphone via Bluetooth as well. Another way to transfer files, playlists or anything on your browser is by means of NFC (Near Field Communication). Only a few devices other than the Nexus 7 have this, such as the Samsung Galaxy 3 phones. All you do is tap the tops of the two devices to transfer, and you'll

be the envy of anyone with an iPhone.

Now, let's talk about consumption. Much has been made about the lack of Flash support for all Android versions after 4.0. This affects the Nexus 7, which, as everyone should know by now, runs Jellybean 4.1. This is not as big of a problem as some might think. Some major video content sites like CNN and CNET do still stream only Flash from their Web sites. What some folks don't know is that these and other Flash-only Web sites have much or all of the same content on their YouTube channels in HTML5 that the Nexus 7 will play. In

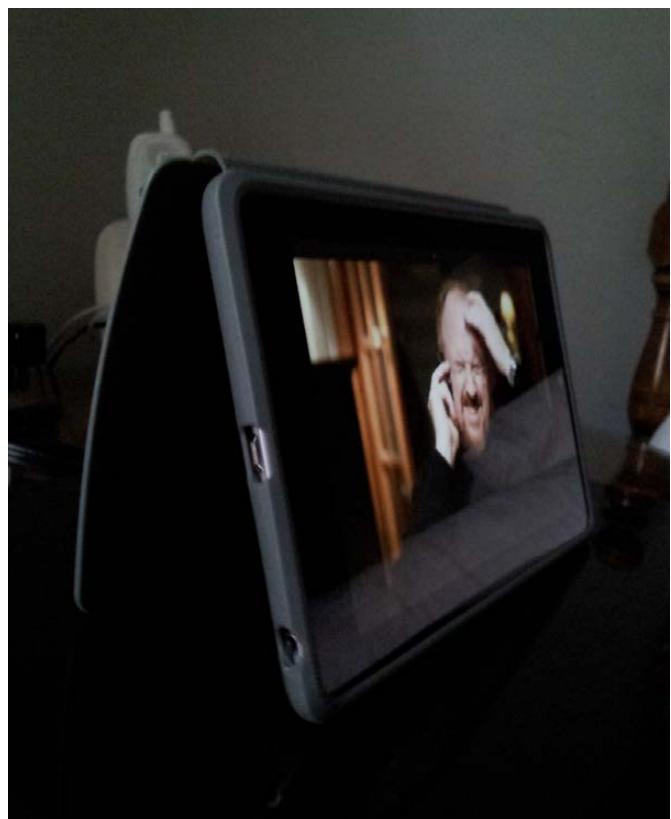


Figure 11. Side View of the Nexus 7

time (I'll predict less than a year), all major video content sites will stream HTML5—it is the future, Flash is not.

I bought my Nexus 7 with the optional gray neoprene case. As you can see in Figure 11, the front flap doubles well on most surfaces as a kickstand for viewing.

As far as other entertainment content, no problem: Netflix app—check, Pandora app—check, Hulu Plus app—check, Spotify app—check, YouTube app—check, Ustream app—check, Tune In Radio app—check, Skype app—check,

Linux Journal app—check.

Maybe you're like me and enjoy recording HD TV programs using Freevo and a Hauppauge HD PVR connected to a cable box to store on your network drive. Using the ES File Explorer app I mentioned earlier, the Nexus 7 will play it. The video I record from the HDPVR is MPEG-4 in a .m2ts container, which the Nexus 7 doesn't recognize. The trick is to hold your finger on the file icon until it asks you what type of file it is (choose video), then choose the player. With a good Wi-Fi signal, the Nexus 7 will stream



Figure 12. Nexus 7 Videoplayer



Figure 13. Playing *Shadow Gun* on the Nexus 7

that 1080i video.

Books and games? Google Play has you covered here too. Games look and play better and smoother than on just about any other Android device, thanks to the Tegra 3 processor.

As an e-reader, it excels as well, with ample choices of font size, background type and intensity, and the pages flip easily.

The point I want to make here is that the Nexus 7 is as full-featured as you want it to be. You just need to find the right app, which is usually if not always found at the Google

Play store. As my final exclamation point, this entire article was written, composed and published on my Nexus 7, using Google Docs within the Google Drive app. It also does spreadsheets. My thumbs that I typed with feel fine, and my eyeballs are firmly connected to my eye sockets—no tactile hallucinations to speak of, from using my Nexus 7. ■

Philip Raymond is a Technician at Fox Chicago and has used Android since the HTC G1. He also has used SUSE, Ubuntu and, most recently, Xubuntu for his desktops since 2003. You can contact him at email-pwraymond@gmail.com.

READERS' CHOICE AWARDS



2012

**Readers' Choice votes are in!
Compare your favorites with
other readers and see if you're
the oddball or everyone else is!**

SHAWN POWERS

It's that time of year again. After 11 months of us telling you about products, services and companies, this month, you, the readers, run the show. We've formatted things a little differently this year, as we've gotten lots of feedback desiring a deeper look into the results.

BEST LINUX DISTRIBUTION

The distribution category has similar results as last year, but it shows a trend. Other distributions, many still based on Ubuntu, are sneaking up the ladder. In fact, next year we'll have to break out the different official brands of Ubuntu, because if they were separated, I suspect the finalists would look a bit different. For that matter, if we lumped all the Debian-based distributions together, the spread would be huge between first and second place!



30.1% Ubuntu (any flavor, Kubuntu, Edubuntu, etc.)

14.7%	Debian	3.1%	Other*	.3%	Pardus
13.4%	Linux Mint	2.5%	Slackware	.3%	Sabayon
10%	Arch Linux	1.9%	Red Hat	.2%	Mandriva
8.6%	Fedora	1.5%	PCLinuxOS	.2%	Puppy
4.2%	OpenSUSE	.7%	MEPIS	.1%	Chrome OS
3.8%	Gentoo	.5%	Bodhi Linux	.1%	Slitaz
3.3%	CentOS	.3%	Novell/SUSE		

**Mageia had the most write-in votes.*

BEST DISTRIBUTION FOR NETBOOKS/LIMITED HARDWARE

Although we kept the title "Ubuntu Netbook Remix" this year, it's really a misnomer, as Ubuntu has consolidated its Netbook interface (Unity) into its main product. With their tiny screens, every pixel counts, and Canonical's one-size-fits-all design works well on a Netbook. It's neat to see Android so high on the list, however. With laptop docks for phones and keyboards for tablets, Android is becoming a viable operating system for more than just making calls and checking e-mail.

25.9% Ubuntu Netbook Remix (any flavor)

14.9%	Debian	3.9%	Fedora	1%	Slitaz
14.1%	Android OS	3.1%	Chrome OS	.9%	Jolicloud
12.7%	ArchLinux	2.5%	MeeGo		
11.3%	Other*	1.7%	PCLinuxOS (any flavor)		
7.1%	Puppy	1%	Slax		

**Both CrunchBang and Linux Mint received many write-in votes in this category.*

BEST MOBILE LINUX OS

88% Android7.3% MeeGo
4% Other

BEST DESKTOP ENVIRONMENT

This was a tough category to create and even tougher to sort out. KDE is easy, as it's still KDE. GNOME, however, has splintered and been modified into a mess of gnomish environments that are sort of GNOME, but sort of not. We split GNOME 2 and GNOME 3 into separate categories, but Unity and Cinnamon have more in common with GNOME than they do with KDE. Should they be lumped together with GNOME? Here are the numbers; we'll let you decide. Whichever way you slice it, KDE's top spot is a bit suspect, since the many GNOME alternatives dominate the chart, albeit separately.

25.9% KDE

21.7%	GNOME: 3.x	3.8%	LXDE
15.2%	GNOME: 2.x	3.6%	Openbox
13.7%	Other*	3%	Enlightenment
12.1%	Xfce	1%	Fluxbox

**Unity and Cinnamon both received several write-in votes.*

BEST WEB BROWSER

It seems as though the browser war is becoming the new GNOME/KDE war. Firefox is still the winner, but Google's browser is right on its heels. This past year, I became a Chromium convert, based mainly on its ability to sync so well and integrate with the Google lifestyle. I find many sites still work better with Firefox, however, so I'm thankful to have a choice when it comes to browsers.

**50.3% Firefox**

40.8%	Chrome/Chromium	1%	Mozilla Seamonkey	.5%	Konqueror
4.7%	Opera	.8%	Rekonq		



BEST E-MAIL CLIENT

40.2% **Thunderbird**

31.3%	Gmail Web client	1.5%	Alpine (pine)
7.5%	Evolution	1.2%	Opera
5.9%	mutt	.5%	Sylpheed
5.5%	Kmail	.3%	GMX
4.1%	Other		(Global Mail Exchange)
1.6%	Claws	.3%	Roundcube

BEST IM CLIENT

The results here line up with last year's results, in that Pidgin is still the top pick. Surprisingly, the Microsoft-owned, completely proprietary Skype is still in second place with a healthy lead over the competition. I switched to BitBee this year for all my IM needs, which falls into the "Other" category, but our favorite response, which came as a write-in from several people, was "Telepathy". It's great to see our digital version has traversed the galaxy and garnered readers on Vulcan. It will be interesting to see if "Mind Meld" makes the list of e-mail clients next year.



56% **Pidgin (formerly Gaim)**

15.1%	Skype	7.8%	Other	.4%	amsn
9.2%	Empathy	1.2%	Gajim	.4%	kmess
8.8%	Kopete	1.2%	Psi		

BEST IRC CLIENT

31.2% **pidgin**

27.8%	XChat	7.4%	Konversation	.8%	KVirc
13.2%	Irssi	6.8%	Other		
9.1%	Chatzilla	3.5%	Quassel		

BEST MICROBLOGGING CLIENT

36.8% **Gwibber**

16%	Other	2.7%	Seesmic
16%	Tweetdeck	2.5%	Jaiku
13.9%	Choqok	1.4%	Nitwit
5.2%	Pino	.7%	Spaz
5%	HootSuite		



BEST OFFICE SUITE

73.3% **LibreOffice**

12%	Google Docs	1.2%	GNOME Office	.2%	IBM Lotus Symphony
8.4%	OpenOffice.org	.5%	Calligra Suite		
3.8%	Other	.5%	KOffice		

BEST SINGLE OFFICE PROGRAM

There were significant write-ins for LibreOffice Writer, which we should have lumped together with OOWriter. Or, maybe we shouldn't have. Nonetheless, based on the office suite results, it makes sense that LibreOffice Writer shares the top spot with OOWriter. Again in the second-place spot is AbiWord. If you're looking for a fairly robust, no-nonsense word processor, it's tough to beat the single-purpose AbiWord.

27.2% **OOWriter**

18.6%	AbiWord	9.6%	Other	3.7%	KWord
13%	OOCalc	6.6%	Gnumeric	1.2%	KSpread
12.8%	Scribus	6.4%	LyX	.9%	OOImpress

BEST GRAPHICS/DESIGN TOOL



This year, GIMP tops the list again as the favorite graphics/design tool. Now that it supports single window mode, we suspect it will only gain in popularity. For some reason, folks found it difficult to navigate the many-window default layout of GIMP. If you're coming from Photoshop, that single window view makes you feel much more at home using Linux.

67.8% **GIMP**

13.3%	Inkscape	3%	Krita
8%	Blender	2.3%	LibreOffice Draw
3.6%	ImageMagick	1.2%	Other

.8% OpenOffice Draw

BEST DIGITAL PHOTO MANAGEMENT TOOL



digiKam edges out Picasa again this year, but since Google dropped the Linux version of Picasa, the number two spot will be gaping open next year. As the husband of a photographer, I have hundreds of thousands of photos to manage, and quite honestly, I haven't found a great way to keep track of them on any platform. digiKam and Shotwell certainly get better and better every year; the question is whether local native apps or cloud-based storage will truly be king.

22.5 **digiKam**

22.2%	Picasa	8.4%	gThumb	3.7%	GQview
17.3%	Shotwell	8.1%	Gwenview	1.6%	Bibble Professional
11%	F-spot	4.9%	Other	.3%	gtkam



BEST AUDIO TOOL

84.6% **Audacity**

7.2%	Ardour	4.4%	Other	3.8%	LMMS
------	--------	------	-------	------	------

BEST AUDIO PLAYER

It still surprises me that VLC is on the list as audio player, and this year it even took top place! I've long used VLC as a video player, but never really considered it a viable audio player. I'm clearly not one of the cool kids, because VLC received almost 25% of the popular vote! Much like photos, a music collection can grow to be unwieldy. It's great to have such a wide selection of tools to tame the beast.



24.9% VLC

20.4%	Amarok	6.4%	Audacious
14.1%	RhythmBox	5.1%	MPlayer
11.1%	Other*	3.5%	XMMS
10%	Banshee	1.6%	Exalie

1.4%	MOC
1.3%	Totem
.1%	Last.fm

*Clementine had several write-in votes.

BEST MEDIA PLAYER

69.1% VLC

15.4%	MPlayer	3.3%	Kaffeine	.6%	Miro
4.7%	SMPlayer	1.7%	XINE		
4.2%	Totem	1%	Other		

BEST BOOKMARK SYNCING TOOL

33% Firefox Sync

32.4%	Chrome Bookmarks	6.9%	Other
11.4%	Google Bookmarks	4.8%	Delicious
11%	xmarks (formerly Foxmarks)	.4%	Weave





BEST ON-LINE COLLABORATION TOOL

61.3% Google Docs

18.5% Wikis	2.6% Kolab
6.9% Other	2.5% Basecamp
5.9% Zimbra	2.3% eGroupWare

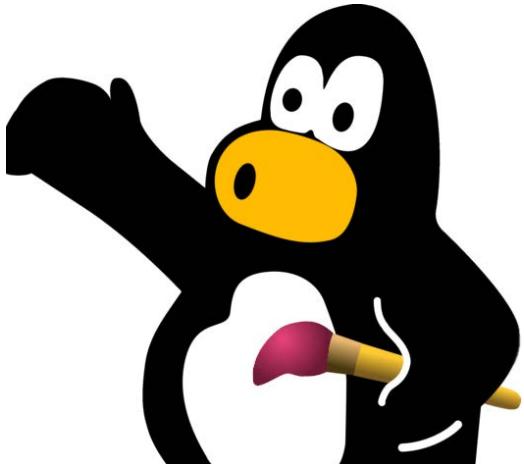
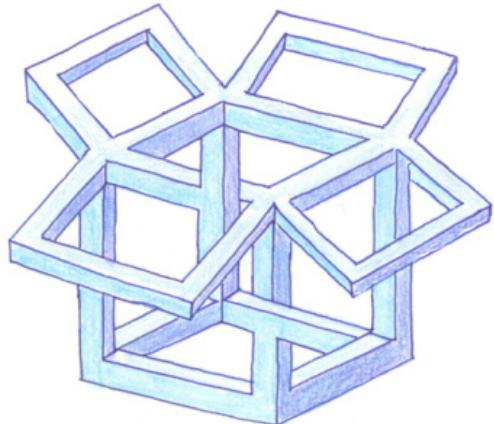
BEST CLOUD-BASED FILE STORAGE

We love Dropbox, and it takes the lion's share of votes again this year. It's also great to see viable alternatives with a decent following. Will Dropbox continue to reign supreme, or will features like SpiderOak's baked-in encryption motivate folks to switch? Dropbox is certainly our incumbent, but with cloud-based storage being such a new market, it's exciting to see what the future holds!

53.4% Dropbox

14.6% Ubuntu One	1.8% SparkleShare
12.2% Other*	.4% CloudSafe
8.6% Amazon S3	.4% SugarSync
5.1% SpiderOak	
3.5% Box.net	

*OwnCloud and Google Drive both had many write-in votes.



BEST KID-FRIENDLY APPLICATION

42.8% Tux Paint

14.6% Gcompris	7.9% Tux of Math Command
11.4% Scratch	
9.4% Tux Typing	5.3% KTuberling
8.6% Other	

BEST GAME

Oh "Other", how we love you. Will Linux users ever get the Steam platform? If so, will that change the landscape of Linux gaming? It's hard to say. We just know it's exciting to see games that work well under Linux, when not too many years ago, that wasn't the case. If you haven't been paying attention to the Linux-friendly Humble Bundle charity drives, be sure to watch for its next offering. Linux users are always the top donators, and all the games are Linux-compatible, so it's a win-win for everyone.

17.6% Other

12.9%	World of Goo	8.2%	Tux Racer (Planet Penguin Racer/ Extreme Tux Racer)	3.8%	freeciv
12.4%	Frozen Bubble	6.2%	Super Tux	3.8%	Urban Terror
9.5%	Doom	3.8%	Enemy Territory	2.8%	Nexuiz
9.1%	Minecraft/ Spoutcraft		Quake Wars	1.2%	Maelstrom
8.7%	Battle for Wesnoth				

BEST SQL DATABASE



This year we separated SQL databases from NoSQL databases. It seemed unfair to pit two different technologies against each other, when both are perfectly cromulent systems. We think the results are much more useful.

47.4% MySQL

28.8%	PostgreSQL	4%	Oracle	.8%	DB2
12.2%	SQLite	1.8%	Other		
4.1%	MariaDB	1%	Firebird		

BEST NoSQL DATABASE

33.4% MongoDB

22.3%	CouchDB	12.2%	Other	1.8%	Dynomite
21.9%	Cassandra	6.6%	Redis	1.8%	Riak

BEST BACKUP SOLUTION

41.7% rsync

11.6%	tar	3.2%	Arkeia
8.9%	Other*	3.1%	BackupPC
7%	Amanda	3.1%	rsnapshot
6%	Bacula	2.8%	luckyBackup
4.4%	Simple Linux Backup	1.7%	rdiff-backup
4.1%	Back in Time	1.2%	cpio

1.2%	Mondo
.1%	BakBone NetVault

*Deja Dup and “custom scripts” had several write-in votes.



BEST VIRTUALIZATION SOLUTION

VirtualBox continues to flourish under its Oracle ownership. Thankfully, the core product remains open source, and its compatibility with fresh versions of distributions seems to be fast and up to date. Although it's not something I'd use for a server room, VirtualBox does a great job of emulating desktop environments.

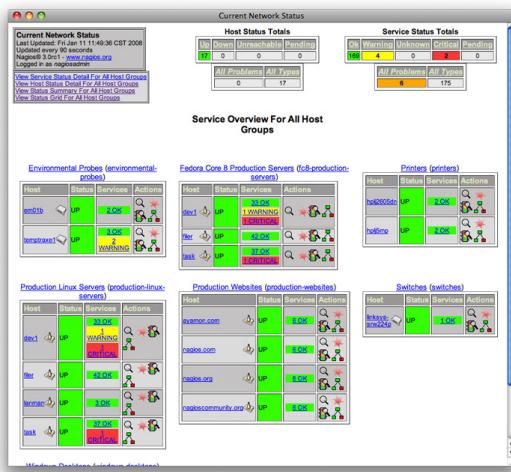
56.1% VirtualBox

17.7%	VMware	3.9%	QEMU	.4%	OpenVZ
13.5%	KVM	1.6%	Citrix Xenserver	.1%	Scalent V/OE Software
5.3%	Xen	1.5%	Other		

BEST MONITORING APPLICATION

51.8% Nagios

13.4%	Other	3.6%	Ganglia
7.9%	OpenNMS	3.1%	Munin
7.2%	Zabbix	2.5%	Hyperic
4.9%	Zenoss Core	1.6%	Groundwork
4%	Uptime Software		

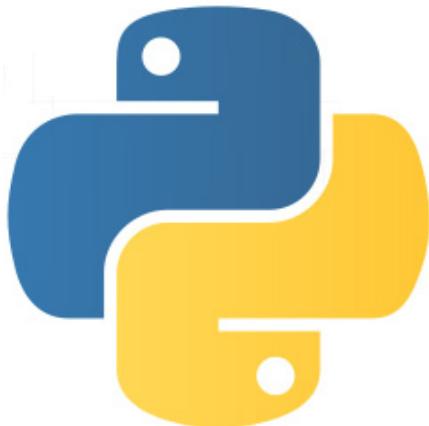


BEST REVISION CONTROL SYSTEM

63% **Git**

18.6%	Subversion	2.7%	Other
6.6%	Mercurial	1.8%	bazaar
6.2%	CVS	.8%	monotone

.1%	darcs
.1%	svk



BEST PROGRAMMING LANGUAGE

28% **Python**

19%	C++	4%	Ruby
18.9%	C	3.1%	JavaScript
8.9%	Java	2.4%	C#
8.2%	Perl	.7%	Erlang
6.1%	Other	.7%	Haskell

New: Intel Xeon E5 Based Clusters

Benchmark Your Code on Our Xeon E5 Based

Tesla Cluster with:

AMBER, NAMD, GROMACS, LAMMPS, or Your Custom CUDA Codes



**Microway MD SimCluster with
8 Tesla M2090 GPUs
8 Intel Xeon E5 CPUs and InfiniBand
2X Improvement over Xeon 5600 Series**

Upgrade to New Kepler GPUs Now!

Microway



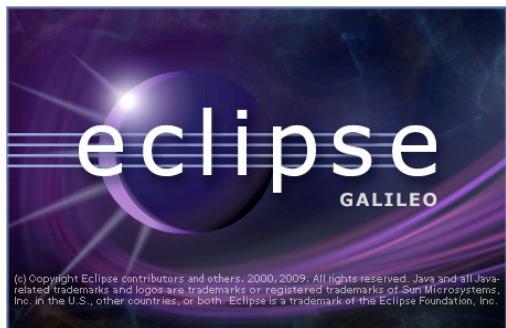
GSA Schedule
Contract Number:
GS-35F-0431N

BEST SCRIPTING LANGUAGE

36.3% Python

24.4%	Bash	4.8%	Ruby
14.2%	Perl	2.2%	Awk
13.8%	PHP	2.2%	Other

1.6%	Lua
.4%	Groovy



BEST IDE

26.7% Eclipse

25.6%	vim
7.5%	Geany
6.9%	Other*
6.1%	QtCreator
5.4%	NetBeans
4.1%	KDevelop
3.1%	Sublime 2
1.3%	IntelliJ IDEA

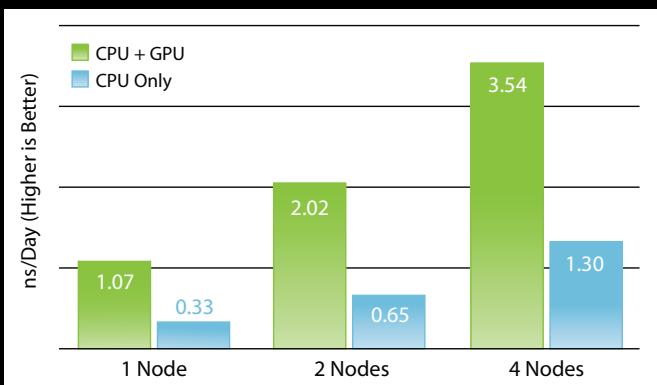
1.2%	Anjuta
.6%	MonoDevelop
.5%	Codewarrior
.3%	Zend Studio
.1%	Eric4

*CodeBlocks and Komodo Edit both received several write-ins.

Harness Microway's Proven GPU Expertise

**Thousands of GPU cluster nodes installed.
Thousands of WhisperStations delivered.**

- Award Winning BioStack - LS
- Award Winning WhisperStation Tesla - PSC with 3D



NAMD F1-ATP Performance Gain

Configure Your WhisperStation or Cluster Today!
www.microway.com/tesla or 508-746-7341



BEST OPEN-SOURCE CONFIGURATION MANAGEMENT TOOL

42.5% **Puppet**

15.7%	Other	8.1%	Cfengine
8.9%	Chef	4%	RHN Satellite
8.9%	OpenQRM	3.2%	Bcfg2
8.7%	Spacewalk		



BEST PLATFORM FOR DEVELOPING RICH INTERNET APPS

Remember back when HTML5 was little more than a fancy dream? It wasn't too long ago that the features of HTML5 seemed ridiculous and impossible, but here we are, in 2012, and HTML5 is the new standard. Web applications are becoming less and less "out there", and much more immersive, thanks to the features of HTML5. Perhaps next year the survey will be "What non-HTML5 platform do you use?", because otherwise, it's a no-brainer.

86.9% **HTML5**

4.7%	Other	2.5%	JavaFX	.2%	ZK Framework
3.1%	Gears (formerly Google Gears)	1.6%	OpenLaszlo		
		.9%	Mono Moonlight		

BEST PACKAGE MANAGEMENT APPLICATION

33.8% **apt**

17.3%	synaptic	3.4%	YaST	.6%	PiSi (Pardus)
12.9%	yum	2%	Software Center	.6%	Smart Package Manager
10.8%	pacman	1.8%	Zypper	.2%	InstallAnywhere
9.2%	aptitude	1.7%	Other		
3.9%	portage	1.7%	pkgtool		

BEST RSS READER

52.4% **Gmail RSS reader**

18.1% Other
16.8% akregator
12.7% liferea

BEST FILE ENCRYPTION

59.3% **GPG**

30.4% pgp
6.6% Other*
3.7% vim-openssl plugin

*TrueCrypt received several write-in votes.

BEST JAVA JRE

57.5% **Openjdk**

38.4% Oracle
4.2% Other

BEST JAVA APP SERVER

60.1% **tomcat**

21.9% jboss
10.9% Other*
7.1% glassfish

*jetty had many write-in votes.

BEST JOURNALING FILESYSTEM

78.8% **ext4**

10.9% ext3
6.2% xfs
4.1% Other*

*btrfs had many write-in votes.

BEST FILE MANAGER

49.8% **Nautilus**

26.9% Dolphin
13.4% Other*
9.9% Thunar

*PCMan File Manager, midnight commander and bash/command-line/vim/shell all received several write-ins.

BEST CONTENT MANAGEMENT SYSTEM

WordPress and Drupal and Joomla, oh my! WordPress takes the victory this year with its easy learning curve and plethora of themes and widgets. Drupal is right on its heels, however, with its flexibility and customization options. All of the CMSes in the survey do their jobs admirably well, and each has its niche to fill in the industry. The best thing to do with all these wonderful open-source options is to try them out and see!



34.9% WordPress

27.8%	Drupal	1.3%	Concrete 5	.2%	e107
15.3%	Joomla!	1.3%	Mambo	.2%	Silva
8.1%	Other	1%	TYPO3	.2%	Zotonic
4.4%	Alfresco	.8%	CMS Made Simple		
1.5%	Liferay	.8%	Documentum		

BEST LINUX-FRIENDLY WEB HOSTING COMPANY

It turns out that our favorite Linux-Friendly Web hosting company is also our favorite Linux game! "Other" is the victor this year, followed by a whole host of Linux-friendly providers. Amazon takes the top non-"Other" spot, but not by as much as I suspected. Linux users are truly an eclectic bunch.

18.9% Other

17.7%	Amazon	2.6%	OpenHosting	.7%	NearlyFreeSpeech.net
12.6%	GoDaddy.com	2.3%	OVH	.5%	Pair
11%	Linode	1.4%	Hurricane Electric	.5%	Slicehost
7.7%	Rackspace	1.2%	Site5	.3%	NeoSpire
5.6%	1&1	1.2%	Siteground	.3%	Server Beach
4.7%	Dreamhost	1%	SoftLayer Technologies	.2%	Arvixe
4.5%	HostGator	.9%	Host Monster	.2%	Codero
2.8%	Bluehost	.9%	Verio	.2%	Contegix

BEST LINUX LAPTOP VENDOR

It's great to see a big company like our top vendor, Dell, supporting Linux users, especially when it's obvious by the numbers that the company is doing a great job of it. Equally as exciting is to see companies like System 76 and ZaReason gain popularity by catering directly to the Linux community. Gone are the days when Linux-supported laptops were hard to find; welcome to the new world where we have choices, options and multiple brands!

24.9% Dell

20.5%	System 76	5.7%	ZaReason	.3%	R Cubed Technologies
19.1%	ASUS	5.2%	EmperorLinux		
12.5%	Other*	3.1%	LinuxCertified		<i>*Lenovo received many write-in votes.</i>
7.4%	Acer	1.4%	Linux Laptop Company		

BEST LINUX DESKTOP WORKSTATION VENDOR

Dell dominates this category as well this year. With more than a third of the vote, Dell is doing something right for our readers! The landscape of workstation vendors is even wider than the laptop world, and again, it's great to see familiar names like System 76, Penguin Computing and Microway making an impact as well.

37.2% Dell

26.9%	System 76	1.8%	HPC Systems	.7%	Open Sense Solutions
14.8%	Other*	1.6%	Pogo Linux	.5%	ASA Computers
8.5%	Penguin Computing	1.2%	Advanced Clustering	.4%	Recompute
3%	Microway		Technologies		<i>*As always, there were many write-in votes for "build my own".</i>
2.1%	R Cubed	1.2%	MadTux		

BEST LINUX SERVER VENDOR

Linux is nothing new in the server room, and the results in this category show it. IBM takes top spot again this year, proving that when it comes to servers, IBM is tough to match. Dell and HP both garner significant votes as well, showing that Linux in the data center isn't a novelty; it's the norm.

24% IBM

23.6%	Dell	2.6%	ZaReason	.4%	Microway
16.5%	Hewlett-Packard	1.6%	Sun Microsystems	.4%	Rackable Systems
10.3%	System 76	1.2%	iXsystems	.2%	Aberdeen
7.4%	Other*	.9%	HPC Systems	.2%	RackMount Pro
4.6%	Penguin Computing	.9%	Silicon Mechanics	.2%	SGI
4.2%	Super Micro Computers	.7%	Tyan Computers		
		.4%	ASA Computers, Inc.		

*“Build my own” had tons of write-in votes in this category as well.

BEST LINUX BOOK

As in previous years, this category was write-in only, and this year, we had a tie for first place between long-standing-winner *Linux in a Nutshell*, by Ellen Siever, Stephen Figgins, Robert Love and Arnold Robbins, and the newly published *The Debian Administrator’s Handbook*, by Raphael Hertzog and Roland Mas. Second place goes to *The Linux Programming Interface: A Linux and UNIX System Programming Handbook*, by Michael Kerrisk.

BEST LINUX JOURNAL COLUMN

I did my best this year to sneak up on Kyle Rankin’s popularity. Heck, the editorial staff even gave me a second column, but added together, my votes paled in comparison to Kyle’s. His relevant, timely, super-geeky column reminds us month after month why we love Linux. Hats off to you, my friend. Dave Taylor and I will happily take silver and bronze to your gold.



28.7% Hack and / by Kyle Rankin

15.4%	Work the Shell by Dave Taylor	9.4%	diff -u by Zack Brown	4.8%	Current_Issue.tar.gz by Shawn Powers
14.5%	The Open-Source Classroom by Shawn Powers	7.6%	At the Forge by Reuven M. Lerner	4.6%	New Products by James Gray
		7.6%	Other		
		7.4%	EOF by Doc Searls		

BEST BRAND OF VIDEO CHIPSET

48% NVIDIA

27.5% ATI/AMD
22.5% Intel

1% Other
.9% VIA



BEST LINUX SMARTPHONE MANUFACTURER



50.1% Samsung

16.2%	HTC	3.4%	Sony Ericsson
8.8%	Motorola	2.4%	LG
7.6%	Nexus	.7%	Dell
6%	Nokia	.3%	Kyocera
4.4%	Other		

BEST LINUX TABLET MANUFACTURER

Tablets have been around for a few years now, but the competition is getting more and more intense. Samsung takes the top spot this year with its passion for Android innovation. If there's one company trying to make sure Apple can't steal the tablet and phone market for itself, it's Samsung. The other company with a runaway number of votes is ASUS. If you add the write-in votes for "Google Nexus", which is manufactured by ASUS, you can see the company has a very healthy second-place position. The real winner in this race? Linux users!

45.2% Samsung

28.7% ASUS
7.1% Other*
6% Motorola
4.4% Acer
2.4% Archos

1.9% Dell
1.6% Toshiba
1.5% Nokia
.4% ViewSonic
.3% Le Pan

.1% Coby
.1% SuperPad
.1% Velocity Micro

*Google Nexus 7 (made by ASUS) received several write-in votes.

BEST OTHER LINUX-BASED GADGET

We're excited, and not at all surprised, to see the Raspberry Pi in the top spot this year. What a neat device, and what amazing potential it has as the core of cool projects everywhere.

47.9% **Raspberry Pi**

26.5%	Amazon Kindle	2%	Popcorn Hour
6.6%	Other*	1.7%	PogoPlug
6.4%	B&N NOOK	1.7%	Sheeva plug
5%	TomTom Navigation System	1.1%	Chumby
		.6%	Kangaroo TV

.3%	Ben NanoNote
.3%	BUG

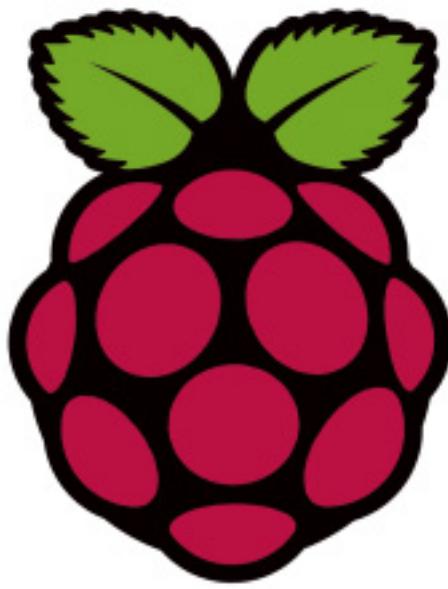
*BeagleBone and BeagleBoard both received several write-in votes.

BEST NEW OPEN-SOURCE PROJECT (RELEASED IN 2011 OR 2012)

Low price point, incredible design and nifty name, Raspberry Pi stole the show this year as the best new Open-Source Project. When the release day finally came, people were lined up outside the store waiting to buy their own Raspberry Pis. Of course, we're geeks, so "lining up" meant waiting at our computers to pre-order on-line, but the concept still was the same. To be fair, we should shine a little light on Cinnamon and OwnCloud, both getting a significant number of votes. Much like Fred Kaps, who followed the Beatles on *The Ed Sullivan Show*, it would be easy to overlook some really great projects!

BEST NEW COMMERCIAL APPLICATION (RELEASED IN 2011 OR 2012)

My fellow optimists, I'm happy to see you voted for Steam based on Valve's commitment to release a Linux client. We know Linux is a great platform for gaming, and if Valve can put its weight behind the support of our OS, it could mean the floodgates would open when it comes to available commercial games. Valve did it for OS X; let's see how it goes for Linux!



PRODUCT OF THE YEAR

Oh, Raspberry Pi, we love you so much! You're our favorite new Open-Source Project, and you're our favorite Linux-based gadget. It only stands to reason you're the Product of the Year! We're excited to see the projects, experiments, modifications and integrations you'll be involved in. When you combine "low price" with "amazing features", it makes for an equation that's hard to beat. We can't wait to see what the Raspberry Pi will do next! ■

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.

Help us make Readers' Choice 2013 even better. Please send suggestions for new categories or any ideas you have for improving the awards via <http://www.linuxjournal.com/contact>.

Introducing Grive

An independent open-source implementation of Google Drive—up in the cloud.

MEHDI POUSTCHI AMIN

Earlier this year, Google introduced its Google Drive cloud storage service. Cloud storage is a model of networked on-line storage where data is stored in virtualized pools of storage that third parties generally host. Hosting companies operate large data centers, and people who require hosting buy or lease storage capacity from them. The data-center operators, in the background, virtualize the resources according to their customers' requirements and expose them as storage pools, which the customers themselves can use to store files or data objects. Physically, the resources may span across multiple servers.

A Web service application programming interface (API) accesses cloud storage services through a cloud storage gateway or through a Web-based user interface. Google Drive is a place where you can create, share, collaborate and keep all of your stuff. You can upload and access all of your files, including videos, photos, Google

Docs, PDFs and more.

The advantage of the Google Drive is that it could become your centralized storage vault for all your documents in the cloud. It lets you create and collaborate in the same way as Google Docs did.

1) Create and collaborate: Google Docs is built right in to Google Drive, so you can work with others in real time on documents, spreadsheets and presentations. Once you choose to share content with others, you can add and reply to comments on anything (PDFs, images, video files and so on) and receive notifications when other people comment on shared items. You can upload up to 30 different file extensions, which many other cloud services don't allow.

2) Store everything safely and access it anywhere (especially while on the go): Synchronization is one of the coolest things ever. When you use a synchronizing storage product, like Google Drive, any file you put in

the drive, and anything you change that's stored in the drive, is updated automatically—not just in the cloud but on all the other devices that you have connected to the Drive. So you can start working on a file on one computer, close it and then open it on a second computer, and what you'll see is the version you closed on the first one. You can access your stuff from anywhere—on the Web, in your home, at the office or while running errands, from all of your devices. You can install Drive on your Mac or PC and can download the Drive app to your Android phone or tablet. And regardless of platform, blind users can

access Drive with a screen reader.

3) Search everything by keyword and filter by file type, owner and more: Drive even can recognize text in scanned documents using Optical Character Recognition (OCR) technology. Let's say you upload a scanned image of an old newspaper clipping. You can search for a word from the text of the actual article. Drive uses image recognition so that if you drag and drop photos from your Grand Canyon trip into Drive, you later can search for "Grand Canyon", and photos of it should pop up. This technology is still in its early stages, and it's expected to improve over time.

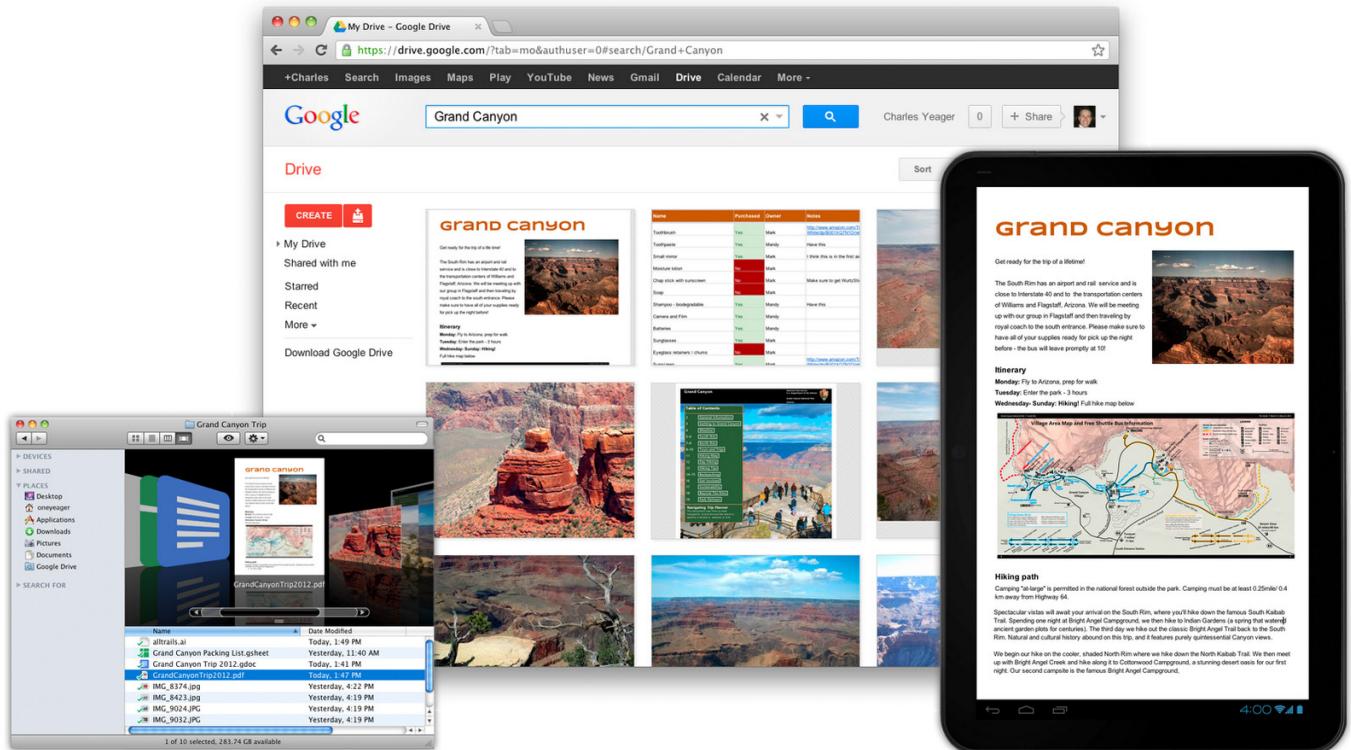


Figure 1. Google Drive on a Mac, PC and Android Phone or Tablet

4) Get your old files: Drive keeps a copy of every file you edit on it. It also tracks changes on documents and keeps revision copies of up to 30 days or the last 100 revisions.

5) Easy sharing to Google+:

Backing up photos is one of the most popular uses of cloud storage services, and with that in mind, it's no surprise to see that Google has made it easy for Google+ users to share images from its new Google Drive service. Google has integrated the two services to allow Google Drive users to share photos from the cloud service to Google+ from within the social network itself.

Google's new cloud storage solution features 5GB of free space. If you have never used a cloud-based storage service like Google Drive before, take a moment to consider the advantages of keeping your files on-line. Because files can be accessed from any computer with an Internet connection, Drive eliminates the need to e-mail or save a file to a USB drive. And because Drive allows you to share files, working with others becomes easier than ever before.

Google Drive and Privacy

Google's biggest problem with the Google Drive privacy policy is that there isn't actually a specific Google Drive privacy policy—there's just Google's new unified term of service and privacy policy.

Google states, "You retain ownership of any intellectual property rights that you hold in that content. In short, what belongs to you stays yours."

That means Google can't use your content for commercial purposes without your consent. However, the term of service also states:

You give Google (and those we work with) a worldwide license to use, host, store, reproduce, modify, create derivative works (such as those resulting from translations, adaptations or other changes we make so that your content works better with our Services), communicate, publish, publicly perform, publicly display and distribute such content. The rights you grant in this license are for the limited purpose of operating, promoting, and improving our Services, and to develop new ones.

For content that is yours, Google can't re-use it for its own purposes, but it can use content you upload in order to serve you. This can include integrating services together (like reading your scanned pictures in order to OCR them), and it can include analyzing your files to target advertisements to you. Google already does this in Gmail.

Google doesn't currently serve ads

in Google Docs (now called Google Drive), but it may, according to its license agreement, use data about the content you upload to target ads to you anywhere on the service.

Google also may give up your data in response to a legal demand, like a subpoena. If you want your cloud storage to be a little more out of touch, you might be interested in Wuala, which has no storage servers in the United States. Or, you might just want to keep your data off the Internet.

Google Drive and Security

Google Drive encrypts data between your computer and the Google servers. If you're using your Google Drive over the Web, the connection defaults to secure (HTTPS), and when you use the software that makes your Google Drive appear on your computer like a local hard drive, the data between your computer and Google is likewise encrypted. No casual hacker will be able to grab your files by monitoring or intercepting your Internet connection to Google. Your data also is stored under lock and key at Google itself, but it is not encrypted on the Google servers. You will have to encrypt your own files ahead of time. A Google rep explained why: encrypting files stored at Google would prevent you from previewing them on the Web, and it also would

prevent services like Google Goggles and its OCR engine from accessing files on your behalf. (I presume it also would prevent Google's ad-serving algorithms from scanning your data to serve you more targeted messages, and this is how Google makes its money.)

Linux Installation

If you install the Google Drive client app for your Windows or Mac PC, you can sync all your on-line Google Docs to your computer. You can be selective with your syncing—sync Google Docs, all of My Drive or individual folders, and items in Shared with me.

But for all the celebration, there is something missing: a Linux client. The Google Drive client is not yet available for Linux, and the old Google Docs FS does not work too well with Google Drive, so besides the Web interface, the best solution for getting Google Drive on Linux right now is using Grive, an unofficial, open-source, command-line Linux client for Google Drive. Grive has been put together by a third-party developer, and it looks quite nice. Thanks Google for making Drive an open platform.

The purpose of Grive Project is to provide an independent implementation of the Google Drive client. It uses the Google Document List API to talk to Google's servers. The code is written in

standard C++.

As of version 0.2.0, Grive can do two-sided synchronization between Google Drive and the local directory. It can download and upload changed files. New directories in Google Drive and the local directory also can be downloaded/uploaded. It cannot yet do the following:

- Wait for changes in the filesystem to occur and upload the files afterward. Sync is performed only when you run Grive.
- Symbolic link support.
- Sync all files/folders with multiple parents and download Google Documents.
- Support for files >2GB.

Those things will be added in the future, possibly during the next release.

At the time of this writing, you easily can install Grive in Ubuntu 11.10 or later using the following commands:

```
system :~$ sudo add-apt-repository ppa:nilarimogard/webupd8
system :~$ sudo apt-get update
system :~$ sudo apt-get install grive
```

On other Linux distributions, the installation is a little involved, as a

Grive package is not yet available. You can compile it from source.

Grive source code is available at GitHub if you want to tinker with it. Setup is a bit of a pain right now, and it requires some command-line work, which is not uncommon in Linux. But, I am here to show you how to install it on RPM-based Linux (CentOS, Fedora and Red Hat).

For starters, you need a dedicated machine or virtual server, as well as a reliable Internet connection.

In this scenario, let's use CentOS. As you may know, CentOS is an enterprise-class Linux distribution derived from sources freely provided to the public by the Upstream OS Provider (<http://www.redhat.com/rhel>).

I strongly recommend using CentOS 6.0, because CentOS 6.0 has been completely rebuilt with a newer build system and library checks to confirm upstream binary compatibility. It also brings a new kernel, new versions of key server components and improved virtualization support.

Before you install Grive, you need to do some preparatory work. Note: all the following commands must be run as root.

First, update your CentOS. This is optional, but I strongly recommend it. Run the yum command:

```
[root@system ~]# yum update
```

Next, disable SELINUX, and simply edit the /etc/selinux/config file:

```
[root@system ~]# vi /etc/selinux/config
```

Then add a line like:

```
SELINUX=disabled
```

Now, you need to enable an extra repository, ATrpms.

ATrpms is a third-party RPM repository. Its original focus was upon software used in natural sciences, especially in the field of high-energy physics, such as tools for numerical programming or for scientific publications. Since then, this repository includes many non-scientific software titles, like system tools or multimedia packages, resulting in a far more generic repository. Currently, packages are built only for Red Hat Linux flavors.

Install and enable ATrpms in CentOS by running the following commands:

```
[root@system ~]# cd /tmp
[root@system ~]# wget http://packages.atrpms.net/RPM-GPG-KEY.atrpms
[root@system ~]# wget http://dl.atrpms.net/el6Server-i386/atrpms/
➥stable/atrpms-repo-6-5.el6.i686.rpm
[root@system ~]# rpm --import RPM-GPG-KEY.atrpms
[root@system ~]# rpm -ivh atrpms-repo-6-5.el6.i686.rpm
```

Next, you must run the following commands to install

the required packages:

```
[root@system ~]# yum install automake autoconf \
openssl openssl-devel json-c json-c-devel curl \
curl-devel libcurl-devel libcurl libarchive \
libarchive-devel libidn libidn-devel expat \
expat-devel binutils binutils-devel
```

CMake is a cross-platform build system generator. CMake is used to control the software compilation process using simple platform- and compiler-independent configuration files. Check your CMake package version:

```
[root@system ~]# rpm -qa | grep -i cmake
```

If the version of CMake is <2.8, you need to update the CMake package by running the following commands:

```
[root@system ~]# yum remove cmake
[root@system ~]# yum install cmake \
--disablerepo=* --enablerepo=atrpms-testing
```

Boost provides free peer-reviewed portable C++ source libraries. Boost libraries are intended to be widely useful and usable across a broad spectrum of applications.

Now, check your Boost package version:

```
[root@system ~]# rpm -qa | grep -i boost
```

If the version of Boost is <1.46, you need to update the Boost package by running the following commands:

```
[root@system ~]# yum remove boost boost-devel
[root@system ~]# cd /tmp
[root@system ~]# wget http://sourceforge.net/projects/boost/
➥files/boost/1.49.0/boost_1_49_0.tar.gz/download
[root@system ~]# tar -xvf boost_1_49_0.tar.gz
[root@system ~]# cd boost_1_49_0
[root@system ~]# ./bootstrap.sh --prefix=/usr
[root@system ~]# ./b2 install
```

Installing the Boost libraries will take up to 15 minutes, depending on your hardware, so be patient.

Make sure your system time and date are correct. Run the following command if required:

```
[root@system ~]# ntpdate -b -p 8 -u pool.ntp.org
```

With all the pieces in place, you are ready to install Grive. The project's Web site offers access to the source code using Git. To download Grive, run:

```
[root@system ~]# cd ~
[root@system ~]# mkdir -p ~/.grive
[root@system ~]# cd /tmp
[root@system ~]# git clone https://github.com/Grive/grive.git
[root@system ~]# cd grive
[root@system ~]# cmake \
➥DCMAKE_INSTALL_PREFIX:/usr .
[root@system ~]# make all install
```

The first time you run Grive, use the -a argument to grant permission to Grive to access to your Google Drive. The -a option is needed only the very first time you run Grive:

```
[root@system ~]# cd ~/.grive/
[root@system ~]# /usr/bin/grive -a
```

After running the command above, a URL should be displayed in the terminal—copy that URL and paste it in a Web browser.

You will need to log in to your Google account if you have not done so. In the newly loaded page, you will be asked to give Grive permission to access your Google Drive, and after clicking "Allow access", an authentication code will be displayed. Copy this code and paste it in the terminal where you ran Grive:

Please go to this URL and get an authentication code:

```
https://accounts.google.com/o/oauth2/auth?scope=
➥https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email
➥+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.profile
➥+https%3A%2F%2Fdocs.google.com%2Ffeeds%2F
➥+https%3A%2F%2Fdocs.googleusercontent.com%2F
➥+https%3A%2F%2Fspreadsheets.google.com%2Ffeeds
➥%2F&redirect_uri=urn:ietf:wg:oauth:2.0:oob
➥&response_type=code&client_id=
➥22314510474.apps.googleusercontent.com
```

Please input the authentication code here

That's it. Now, each time you want to sync Google Drive with your local grive folder, navigate to the .grive folder and run /usr/bin/grive (this time without the -a, because you already have authenticated Grive with Google Drive). You can set up a cron job if you want to do this on a regular basis.

If everything works fine, Grive will create a .grive file in your home directory. It also will start downloading files from your Google Drive to your current directory. After all this business is done, all you need to do to refresh your Google Drive files is run Grive from the terminal.

Is it as convenient as Google's official desktop solutions? No, but you are not running Linux because you are looking for easy solutions.

Ending Notes

If you get a "crash" (a bunch of error messages on the screen), you should run:

```
[root@system ~]# /usr/bin/grive -l log.txt
```

Because the file will contain a log of the sync operation, you may want to edit the file first to remove personally sensitive information and then send it to Grive developers.

Keep up with the posts at <https://plus.google.com/116411649000013553715/posts> and <https://github.com/grive/grive/issues> for the latest information, especially regarding news on newly discovered bugs and when you need to download and build again! ■

Sayed Mehdi Poustchi Amin is currently a PhD research student in computer science at SIU. His research is focused on developing honeypot decoys for analysis of Internet attacks. His professional certifications include CCNA, MCTS, MCITP and MCSE. He welcomes comments on this article and beyond at mehdi.poustchi@gmail.com.

Resources

The Grive Project—an Open-Source Linux Client for Google Drive: <https://github.com/Grive/grive>

The Official Google Blog, "Introducing Google Drive": googleblog.blogspot.in/2012/04/introducing-google-drive-yes-really.html

"Grive: Open Source Google Drive Client for Linux": www.webupd8.org/2012/05/grive-open-source-google-drive-client.html

The Google Drive FAQ: news.cnet.com/8301-1023_3-57420402-93/the-google-drive-faq

GIMP 2.8

After years of exhaustive and sluggish development, the venerable raster image editor finally has a new release. Laden with new features and a completely new UI, the open-source image editor tries to appeal to graphics designers. Let's take an in-depth look at the new version of the popular editor, GIMP.

SHASHWAT PANT

Professional graphics programs always have been proprietary OS turf, but with time and massive development, the Open Source community has made the ecosystem an artist's resort. Ranging from boggling 3-D processing elements to nifty vector processing, Linux graphics editing now is mature enough to sustain artists' enthusiasm for Linux,

Until recently, GIMP's interface was very much a mixed-bag experience—too perplexing with a lack of efficient window control. Besides, its meager 8-bit channel support was a major hurdle in mass adoption. Finally, after four years of development following GIMP 2.6, developers closed the curtain on the next major release. GIMP 2.8 surpasses the older iteration with a more stable and feature-rich approach.

On June 5, 2012, taking a huge four-year window for development,

developers finally presented the much awaited version of GIMP 2.8. Unlike 2.6, where major changes were made in the core base of the software, 2.8 brings an overall makeover of the software stack. It features a brand-new UI with single window mode, many nifty cosmetic changes and much awaited changes in the core, such as support for OpenCL hardware acceleration for improved and faster results.

The main idea behind the new interface is to provide smoother, more intuitive and better manageability while working with multiple images, along with support for higher-bit raw files and hardware acceleration, paving the way for enhanced editing without sacrificing performance.

So without further ado, let's take an in-depth look at the newly refurbished GIMP. Since there are no bug releases at the time of

GEGL and BALB

GEGL (Generic Graphical Library) is the new image processing core that debuted with GIMP 2.6. GEGL provides power graphics-based array calculations that make editing more powerful, allowing developers to leverage high-bit-depth processing along with efficient color channels. Support for non-destructive editing and hardware acceleration makes GEGL a perfect library for GIMP to leverage. In future releases, GEGL will support 16-bit and 32-bit high image editing solutions.

BALB is a support library to GEGL that allows seamless color conversion, making photographers and editors feel at ease when working with new versions of GIMP.

this writing, I am using the vanilla variant, 2.8.0. The hardware I used for testing is based on AMD's first-generation APU—that is, Llano 3850 with 4GB of RAM.

Getting GIMP 2.8

GIMP is one of the premier software packages in the Linux ecosystem, and it comes pre-installed on almost all distributions. To get the latest version of GIMP, check your package manager. If your distribution doesn't ship with the latest version, you can compile it from the tarballs provided at the GIMP Web site.

Users looking for the latest build should grab the source from the GIMP Git repository at <http://developer.gimp.org/git.html>; however, in order to maintain compatibility, you must ensure that all dependencies, such as GEGL and BALB,

also are updated.

The best part about GIMP is its light nature. The application is uniquely positioned and designed so that it can run on any hardware without any glitches, and the measly install and package size makes it pocket-friendly software, so if you want to carry it around, you can have an installation of GIMP in your portable driver.

The first notable change in the GIMP is the splash screen. Developers have now opted for a wider and bigger screen, much different from the long flat splashes that debuted with older iterations. Once the application opens though, you will notice the same old floating windows and docks. The developers have refrained from switching completely to single window mode



Figure 1. GIMP 2.8 Splash Screen

with this release.

Single Window Mode

The most touted feature that initially was set to debut with GIMP 2.10, finally arrives for mass adoption and testing. Single window mode is disabled by default, but you can enable it from Windows→single window mode. Once you have enabled the option, GIMP will wrap all the floating boxes into a single entity making it smoother and more efficient to work with.

To position GIMP more uniquely in the competitive software ecosystem, developers have opted for a more intuitive approach, which is reminiscent of other major software solutions. The single window mode is more efficient and works across all desktop environments, unlike the Utility Window that debuted with GIMP 2.6, which had issues with the KDE Kwin windows manager. Even in single window mode, you can customize the GIMP toolboxes the way you like, so the liberty to

change GIMP according to your tastes still holds true.

Tabbed Navigation

The other major UI change is the tabbed view. Often photographers and

designers work with multiple images, so it usually is cumbersome for an artist to work with multiple windows. To tackle this issue and provide an easy way to switch back and forth when dealing with multiple images,

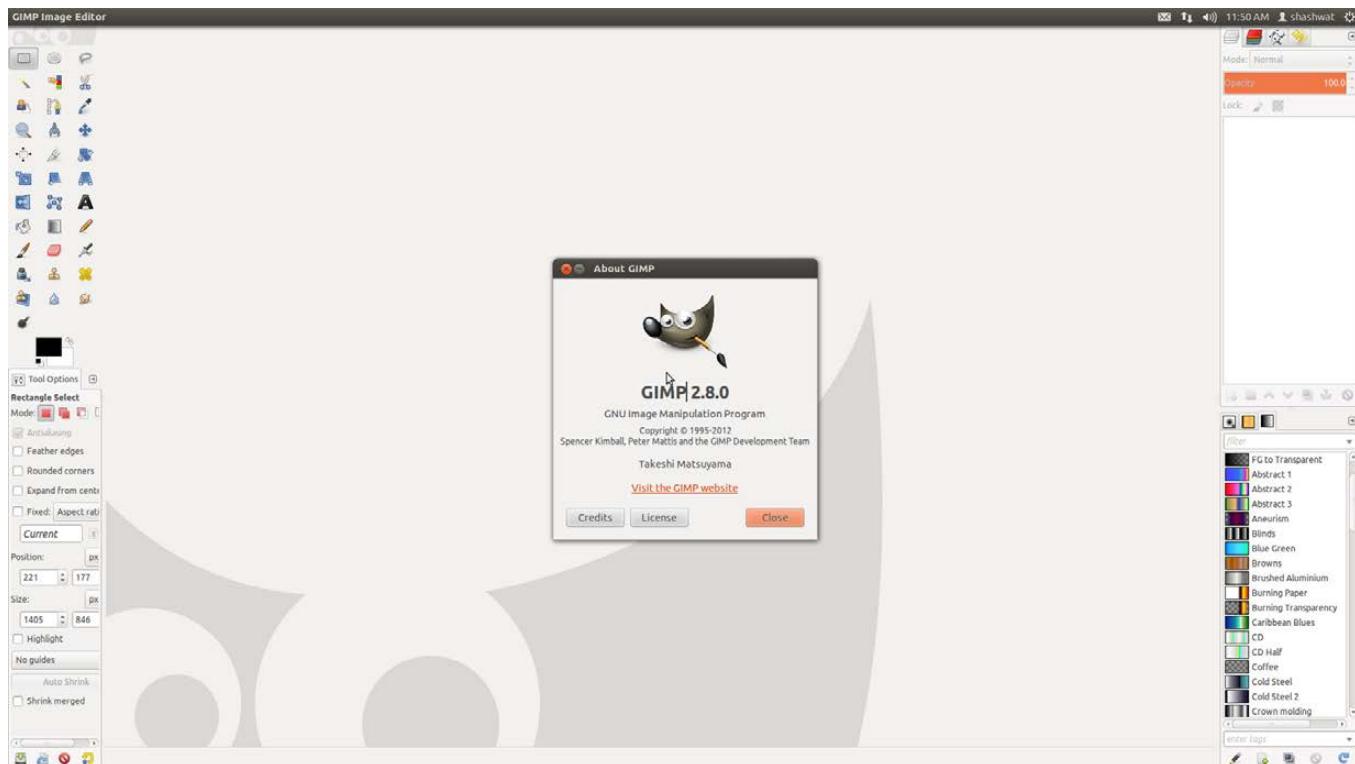


Figure 2. Single Window Mode in GIMP 2.8

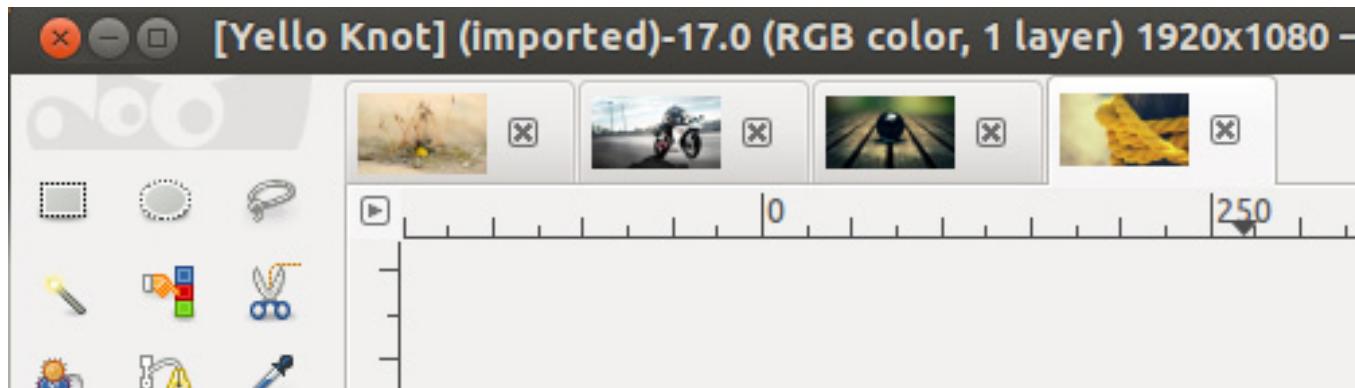


Figure 3. Tabbed Image View in GIMP

GIMP has come up with a tabbed view of all the active windows.

The tabbed view sits on top of the editing canvas and displays a small thumbnail preview of all the active windows—an efficient way to toggle between multiple images. It's a feature that isn't very new among graphics editors, but it still is a worthy add-on for multitasking with images easily.

Other UI Changes

Major UI changes have overshadowed most of the new release's makeover; however, the developers have paid close attention and fine-tuned some smaller areas and made changes where they matter most:

- Multi-column dock: with previous versions, it was possible to group multiple docks together in a

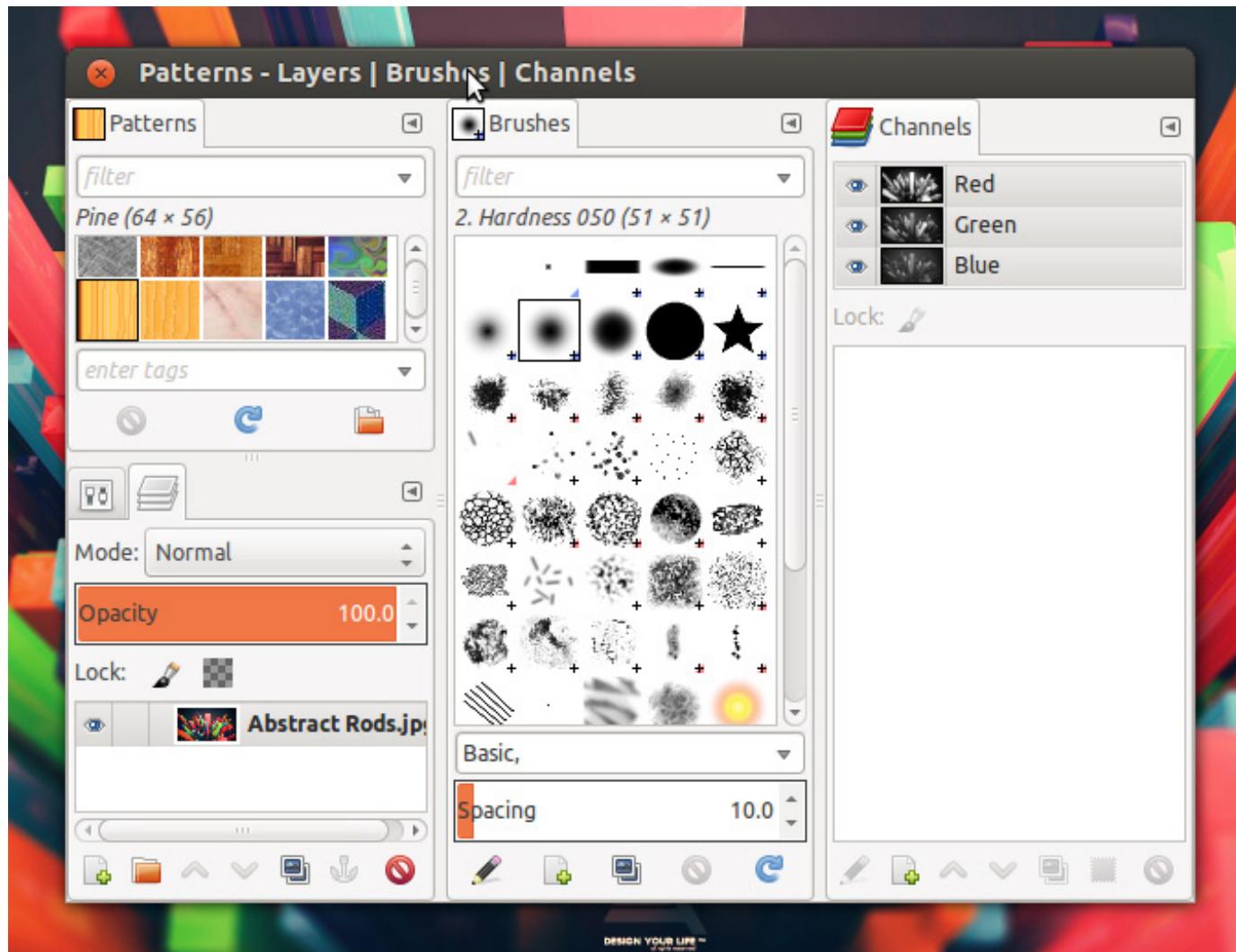


Figure 4. Multiple Tool Bars Docked Sideways

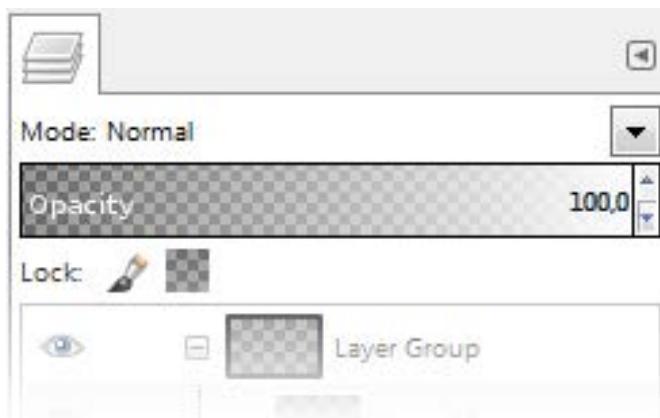


Figure 5. New Opacity Slider Enhanced for Touch Devices

tabbed view, but with GIMP 2.8, developers are allowing users to dock the window sideways or in vertical view. The ability to stack multiple docks is a nice way to let users modify the look and feel freely according to their own needs and working area.

- Transparency slider: GIMP 2.8 brings a new transparency slider to the table. Initially suggested with the GIMP usability project, the new idea saw daylight with the final release. The changes are tailor-made to target users with tablets and those who use a stylus or pen for drawing and painting with the GIMP.
- Additional space for toolboxes: the docking bars now have more area and less unused space. Developers

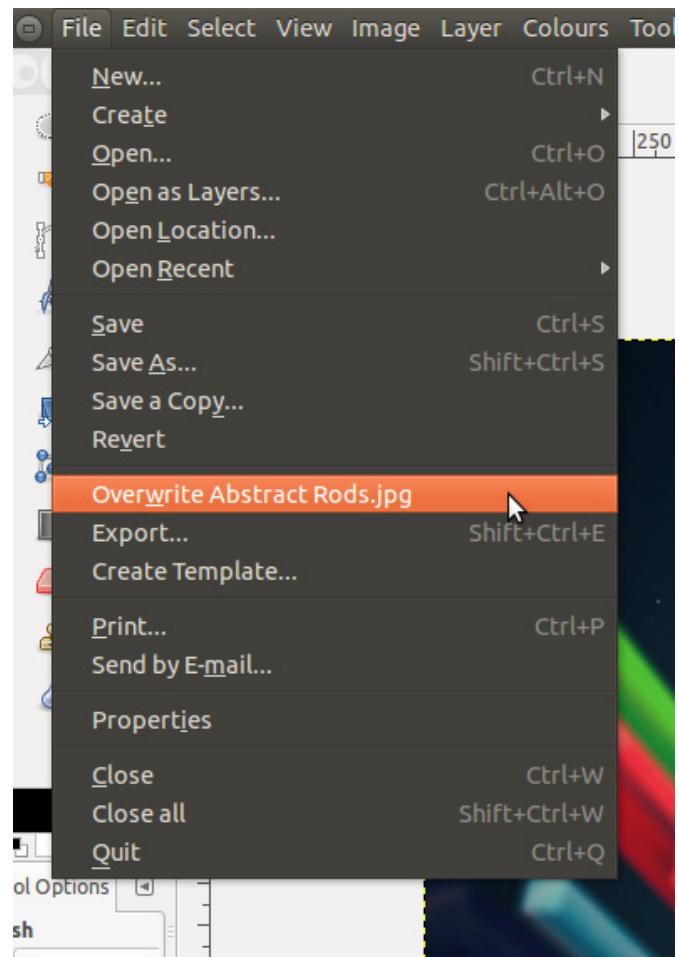


Figure 6. New Save and Export System

- have removed the top unused area and done away with the tab information that took up a lot of space. As a result, toolboxes are more efficient and have less clutter.
- Save and export features: GIMP's latest release has two new ways to save files. The previous save feature worked for saving both GIMP's native .xcf files (that save all the detailed information)

along with native MIME image formats, such as .png, .jpg and so on. With GIMP 2.8, the save feature will save images only in GIMP's native .xcf format. To save images in MIME image formats, you have to choose the Export option from the file menu.

That sums up GIMP's changes in the user interface department. However, many bold changes in the new

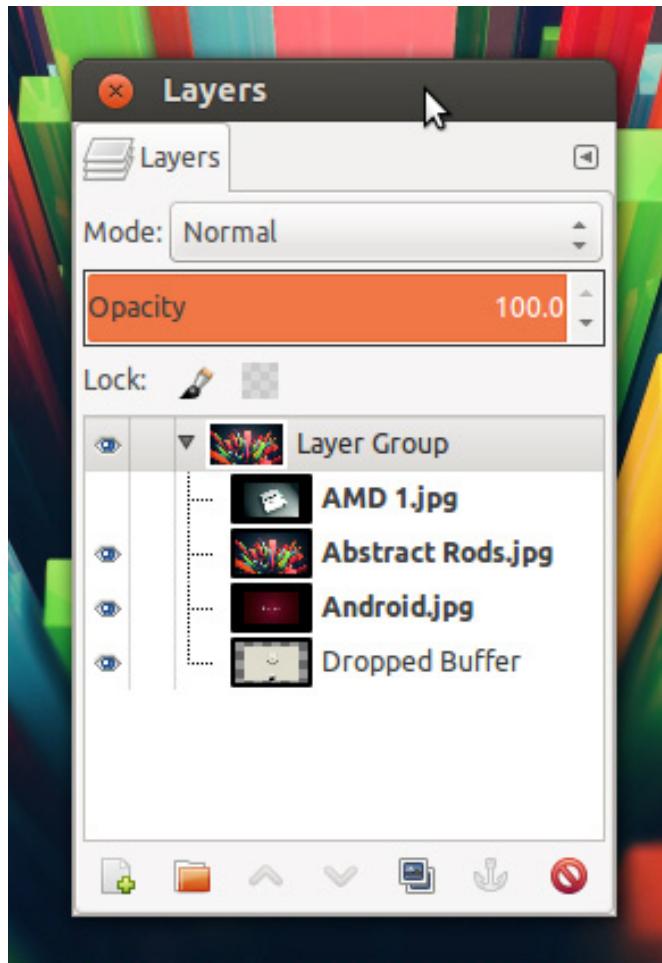


Figure 7. Layer Grouping with Multiple Layers Grouped

release not only enhance the editing experience but also pave the way for future GIMP development.

Layer Grouping

The most painful part in editing and creating huge images are layers. An image is an amalgamation of many layers, and managing images with many layers can be a daunting task. To provide an easy workaround, GIMP 2.8 boasts layer grouping features. You simply create a new group, and if you want to add layers, simply drag and drop into the layer group.

Inside the layer group, you can carry out all the generic layer-related tasks, so these new groups don't limit your editing.

Resource Tagging

Tagging is an efficient way to locate data, especially if you have several plugins, brushes and gradients installed. GIMP 2.8 lets you easily tag resources that allow you to navigate quickly through the massive resources at your disposal.

On-Canvas Text Editing

This feature created great hoopla with the GIMP 2.8 release. On-canvas editing, which was part of a project from Google's Summer of Code 2011,

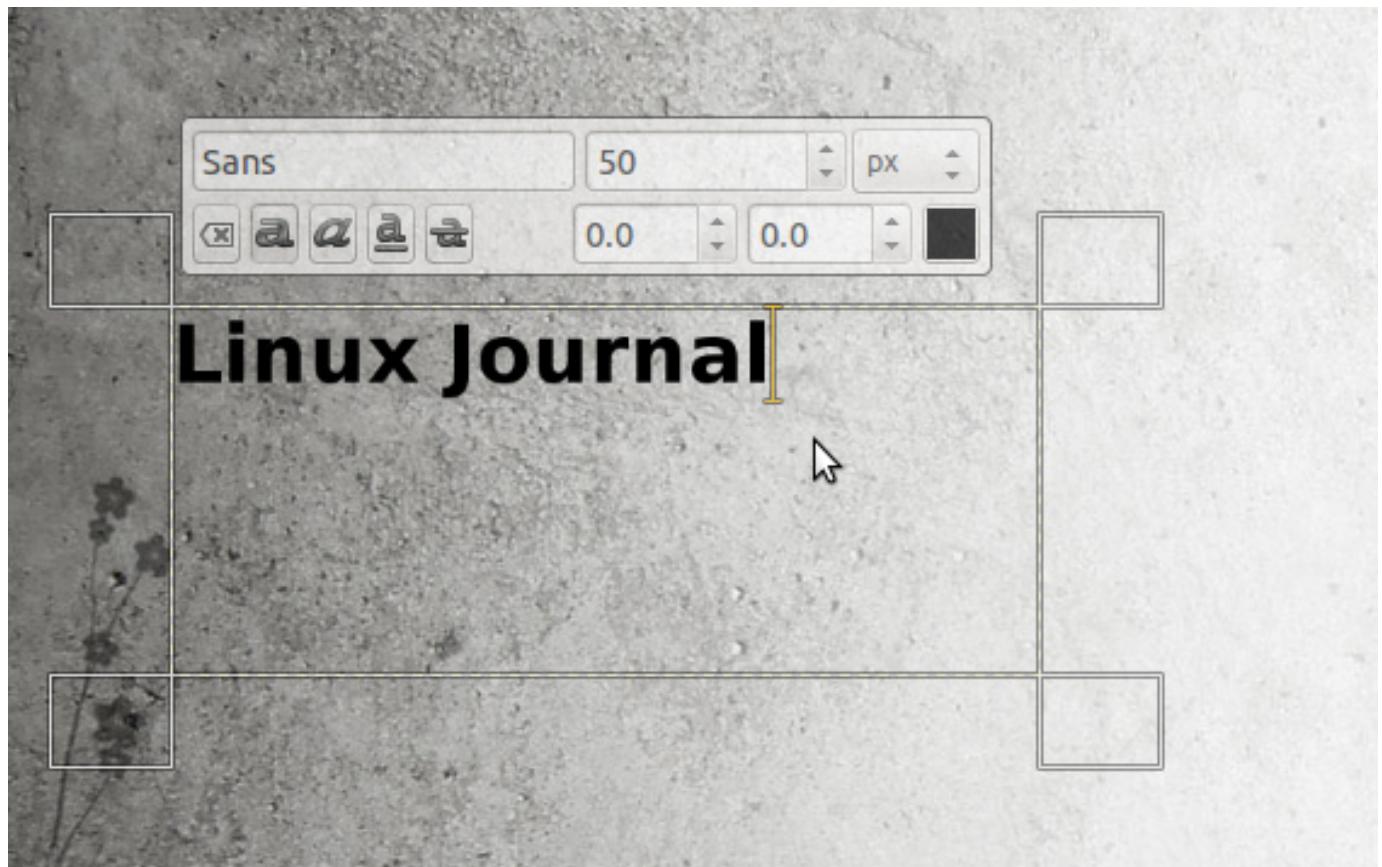


Figure 8. On-Canvas Editing in Action

allows you to edit text directly in the canvas—no more pop-up boxes for adding text to images. The main advantage of the on-canvas editor is to provide a clutter-free and more precise experience.

Cairo Adoption

All on-canvas tools, such as perspective transformation, shear, rotate and scale tools, are redrawn using the Cairo graphics library. As a result, the tools look more professional, and they carry a new on-canvas status indicator and no

longer use the ubiquitous status bar at the bottom.

GEGL and OpenCL Implementation

Starting with GIMP 2.6, GEGL has come a long way with major improvements. The most important is support for OpenCL. With support from AMD and a GSOC project, GIMP now scales with graphics processors for image processing, which makes a normal image function much faster.

GIMP, being an image editor, requires you to work with array

variables and even multiprocessing capabilities are needed to complete tasks quickly; however, with the advent of SIMD-based processors (graphics card/processors) and OpenCL, it is now much easier for graphics-intense applications to leverage their work on SIMD engines for faster processing.

However, the current state of GEGL/OpenCL is dismal in Linux. GEGL doesn't scale well, and in fact, it performs much slower when compared to plain CPU-based processing. This could very well be the AMD driver's fault, as its OpenCL support and performance in Linux is still flaky.

GEGL also has developed support for higher-bit image processing, which is still in development and should debut with GIMP 2.10. Developers have already laid out a plan for fully functional and complete GEGL integration with GIMP 2.10, so in future releases, all the functions will be done in parallel using OpenCL code stack.

Closing Thoughts

With many nifty add-ons, such as mathematical calculations in resize windows, improved cage transformation tools and API changes for simplified script writing,

GIMP 2.8 has come a long way and is a far cry from its older versions.

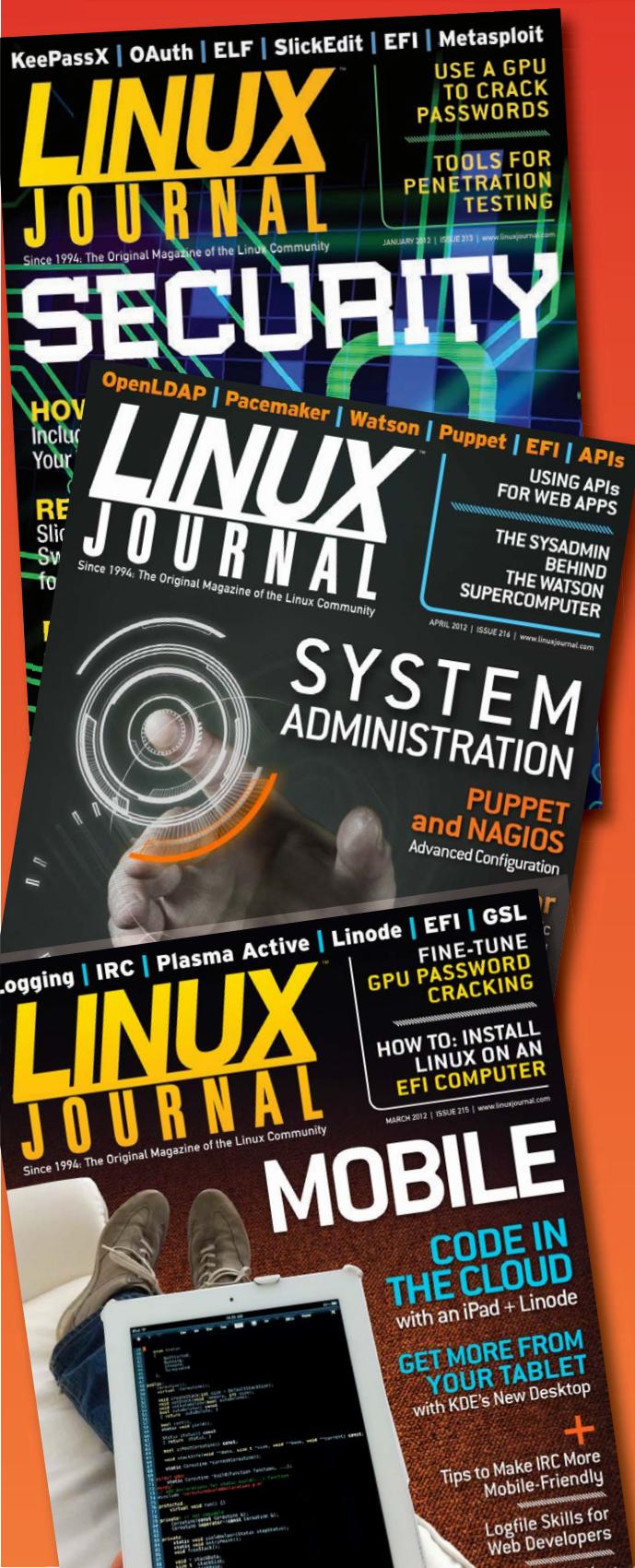
The revamped UI and new underlying code have given GIMP a much-needed momentum. With GIMP 2.8, developers have covered many milestones, providing a more enhanced interface along with bleeding-edge OpenCL acceleration.

However, GEGL support and the dismal OpenCL performance, at least under Linux, have kept me thinking. With refinements every day, I believe that with GIMP 2.10, we will have full-blown OpenCL-accelerated image editing.

Looking forward, GIMP 2.10 and 3.0 will bring an even more massive changelog, making it a worthy competitor to proprietary solutions. With GIMP 2.8, developers have cleared many hurdles that hindered its mass adoption. The clunky interface is more usable and intuitive than ever. GIMP 2.8 establishes itself as one of the premier open-source alternatives to proprietary solutions, providing bleeding-edge solutions with an intuitive interface and an ever-improving ecosystem. ■

Shashwat Pant is a FOSS enthusiast and hardware freak. He likes to tweak his hardware for optimum performance. He is an IT Engineering student studying in India and likes to review the latest FOSS software and distributions.

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



- » 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 & Kindle formats, on-line and through our Android & iOS apps. Wherever you go, *Linux Journal* goes with you.

SUBSCRIBE NOW AT:
WWW.LINUXJOURNAL.COM/SUBSCRIBE



DOC SEARLS

Biggest Data

Our old enemy was proprietary platforms. Should it now be proprietary data?

Turns out maps matter.

That's always been the case for me. I'm a map freak. I own hundreds of paper maps in various specialties, plus many atlases, books on geography, geology and other geo-obsessions. But I'm no longer an edge case, because maps are proving to be essential on smartphones, which today approaches a billion or more people. Digital maps on phones are now among the core portfolio of smartphone apps, alongside voice, text, calendar and contacts. What could be more mobile about a phone than a map to help the user look things up and get around?

In appearance and function, smartphone maps evolved from standalone GPS units, which were limited to location lookup and turn-by-turn directions in both text and voice form. But maps on phones grew much deeper, because boundless amounts of memory, intelligence and usage-based heuristics could be located in the cloud, and delivered to the app and the user through an API

over a live data connection.

Nobody has done more with cloud-based mapping than Google. By the middle of this year, Google-made or -based map apps were the primary sources of location and navigation information for users carrying all kinds of mobile devices—especially Android and Apple mobile devices. Among smartphones, that pair alone comprised about a 90% market share in the US.

In September 2012, however, Apple released iOS 6 and the iPhone 5, both with a new Maps app that did not use Google as a data source. The new Google-free Maps app quickly turned into Apple's worst fail since the Newton. Initial sales of the iPhone 5 were good, as were installs of iOS 6. But then came reports of the new Maps app's disabilities, which were extreme. Missing were countless major points of interest, such as the entire subway systems of New York, London and Paris. Museums were moved into rivers. Dead businesses came to life, and live businesses went away. At the time of this writing

(in early October 2012), iPhone 5 sales projections have been revised downward, and Apple's stock is dipping as well. The map debacle may not be the only reason, but it's still a big one.

Why would Apple do something so obviously harmful to itself? Because Google was clearly withholding essential mapping features from iOS to favor Android, and Apple needed to rid itself of a clearly hostile supplier. As an owner of both Android and Apple phones, that much was obvious to me. But news of it didn't

make the mainstream press (at least as far as I know) until David Pogue reported as much in his *New York Times* column of September 27, 2012 (http://www.nytimes.com/2012/09/27/technology/personaltech/apples-new-maps-app-is-upgraded-but-full-of-snags-review.html?_r=0):

After poking around, here's what I've learned.

First, why Apple dropped the old version: Google, it says, was saving

LINUX JOURNAL

now available
for the iPad and
iPhone at the
App Store.



Available on the
App Store

linuxjournal.com/ios

For more information about advertising opportunities within *Linux Journal* iPhone, iPad and Android apps, contact Rebecca Cassity at +1-713-344-1956 x2 or ads@linuxjournal.com.



all the best features for phones that run its Android software. For example, the iPhone app never got spoken directions or vector maps (smooth lines, not tiles of pixels), long after those features had come to rival phones.

Apple has a variety of replacement data sources, starting with TomTom, the Netherlands-based maker of navigation (mostly GPS) systems. TomTom's primary data source is its Tele-Atlas subsidiary, which it acquired for 2.9 billion euros in 2008. The prior year, Nokia acquired Navteq, Tele-Atlas' main competitor, for 5.7 billion euros. Google used Tele-Atlas as a source until October 2009, when it became its own primary data source.

Google, Navteq and Tele-Atlas all deploy special vehicles on streets to create, correct and enhance mapping. In Google's case, this also has included mapping of Wi-Fi access points to fine-tune a user's location further. Microsoft, Google and Apple also have aerial views provided both by satellites and low-flying aircraft. And all are feeding their clouds and crunching up data gathered by watching what users do.

At this point, it would be easy to digress into vendor sports and handicap

which of the map data sources and mobile device suppliers will win in the marketplace. But instead, I want to look at what all this says about dependencies. What we see here is the replacement of the platform with the Big Data API—let's call it a BiGDAPI—as a single source of market-making and market-breaking dependency.

Google's passive-aggressive map app game with Apple, and Apple's refusal to keep playing it, are both evidence of Google's huge leverage through its map BiGDAPI.

The term "Big Data" has been around for a long time, but has obtained buzzphrase status only in the last two years. Although much that can be said about Big Data is positive and harmless (better medicine, better science, better analytic fodder for countless good purposes), one unspoken motivation behind the buzz is obtaining high degrees of market leverage. And much of that leverage is not in harmony with the constructive motivations and practices behind free software, open source and Linux. Because, behind many of the big APIs are vast jungles of exclusive and patent-protected functionalities and restrictions around use. Such as, for example, the spoken turn-by-turn directions Google wouldn't allow Apple to use. It can be dispiriting

to see platform leverage exceeded by large proprietary databases and exclusive services made available through APIs. But it's important to bring attention to what's going on, so here we are.

We also should remember that good things come from APIs. For exposing organizational competencies in purely useful ways, nothing beats them. (For a sample, take a look at the growing list of APIs at ProgrammableWeb.com: <http://www.programmableweb.com/apis/directory>.)

One hopeful force on the mapping front is OpenStreetMap.org, which has been crowdsourcing maps data for years. It has an API too (<http://wiki.openstreetmap.org/wiki/API>). Perhaps in time, free developers and users can best the sums of data currently locked up in proprietary mapping bases, and the analytics as well. But until then, it's helpful just to watch what's happening in the mapping space, and how dependent we remain on proprietary companies and BiGDAPs over which we have little control. ■

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.

Advertiser Index

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

ADVERTISER	URL	PAGE #
1&1	http://www.1and1.com	17
EMAC, INC.	http://www.emacinc.com	19
IXSYSTEMS	http://www.ixsystems.com	7
MICROWAY	http://www.microway.com	80, 81
SCALE	http://www.socallinuxexpo.org	55
SILICON MECHANICS	http://www.siliconmechanics.com	3
USENIX LISA	https://www.usenix.org/conference/lisa12	2

ATTENTION ADVERTISERS

The *Linux Journal* brand's following has grown to a monthly readership nearly one million strong. Encompassing the magazine, Web site, newsletters and much more, *Linux Journal* offers the ideal content environment to help you reach your marketing objectives. For more information, please visit <http://www.linuxjournal.com/advertising>.