

PL/R | DNS | Arduino | Squeezebox | NVM | T

LINUX JOURNAL™

Since 1994: The Original Magazine of the Linux Community

TELNET
TROUBLESHOOTING

BECOME A
KNOWLEDGABLE
DNS USER

FREE TO
SUBSCRIBERS
EPUB, Kindle, Android, iPhone & iPad editions

SEPTEMBER 2012 | ISSUE 221 | www.linuxjournal.com

EMBEDDED

AN ARDUINO-INSPIRED
HARDWARE PROJECT

THE NEXT
BIG THING IN
MAIN MEMORY
IS GOING TO CHANGE
EVERYTHING

STREAM YOUR MUSIC WITH
LOGITECH
SQUEEZEBOX'S
OPEN PLATFORM

Reviewed
ZaReason's
ZaTab

Bash
Notational
Shortcuts

Embed the R Language
in PostgreSQL
for Powerful Statistical Analysis

SAVE THE DATE!



LISA '12

26th Large Installation
System Administration Conference
sponsored by USENIX
in cooperation with LOPSA

DECEMBER 9–14, 2012 | SAN DIEGO, CA

**Come to LISA '12 for training and face time
with experts in the sysadmin community.**

LISA '12 will feature:

6 days of training on topics including:

- Virtualization
- Security
- Configuration Management
- Cloud

Plus a 3-day Technical Program:

- | | |
|-----------------------------------|-----------------------|
| • Invited Talks | • Guru Is In Sessions |
| • Paper Presentations | • Vendor Exhibition |
| • Practice and Experience Reports | • Workshops |
| | • Posters and WIPs |

The LISA '12 keynote address will be delivered by Vint Cerf, Google.

Registration opens mid-September.

www.usenix.org/lisa12

STAY CONNECTED:



[FACEBOOK.COM/USENIXASSOCIATION](https://www.facebook.com/usenixassociation)



[TWITTER.COM/LISACONFERENCE](https://twitter.com/lisaconference)



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

SEPTEMBER 2012
ISSUE 221

EMBEDDED



FEATURES

- 70 Logitech Squeezebox Platform: Livin' in the Land of (Open-Source) Hi-Fi**

Stop worrying about uploading gigabytes of music files to the cloud, and start enjoying your music with the Logitech Squeezebox.

Craig Maloney

- 84 Arduino Teaches Old Coder New Tricks**

An Arduino-inspired hardware project using the gEDA open-source Linux software suite for printed circuit development.

Edward Comer

- 106 The Radical Future of NVM**

Think there has to be a trade-off between high performance and storage persistence? Not anymore.

Richard Campbell

COLUMNS

32 Reuven M. Lerner's At the Forge

PL/R

40 Dave Taylor's Work the Shell

Bash Notational Shortcuts: Efficiency over Clarity

44 Kyle Rankin's Hack and /

Troubleshooting with Telnet

50 Shawn Powers' The Open-Source Classroom

A Domain by Any Other Name...

118 Doc Searls' EOF

Making the Case to Muggles

REVIEW

62 ZaTab: ZaReason's Open Tablet

Kevin Bush

IN EVERY ISSUE

8 Current_Issue.tar.gz

10 Letters

18 UPFRONT

30 Editors' Choice

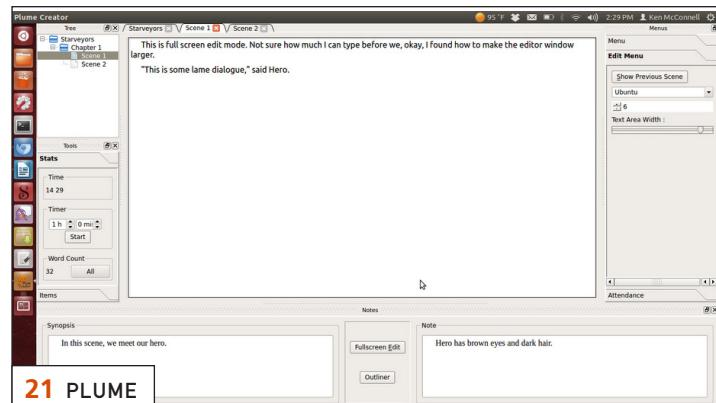
58 New Products

123 Advertisers Index

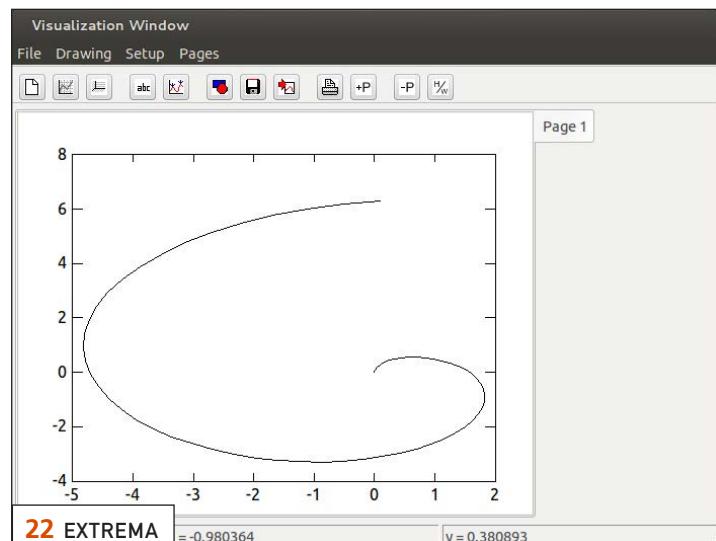
ON THE COVER

- Telnet: a Handy Troubleshooting Tool, p. 44
- Become a Knowledgeable DNS User, p. 50
- An Arduino-Inspired Hardware Project, p. 84
- The Next Big Thing in Main Memory Is Going to Change Everything, p. 106
- Stream Your Music with Logitech Squeezebox's Open Platform, p. 70
- Bash Notational Shortcuts, p. 40
- Reviewed: ZaReason's ZaTab, p. 62
- Embed the R Language in PostgreSQL for Powerful Statistical Analysis, p. 32

Cover Image: © Can Stock Photo Inc. / silvertiger



21 PLUME



22 EXTREMA



62 ZAREASON'S ZATAB

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

Proofreader	Geri Gale
--------------------	-----------

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 Faloutsos • Dennis Franklin Frey • Alicia Gibb
Victor Gregorio • Philip Jacob • Jay Kruizenga • 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 320 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

The Borg Ran Windows

I was watching *Star Trek the Next Generation* the other day with my 13-year-old daughter, and I began to ponder what operating system the Borg used. Based on shape and available systems in the early 1990s, you might think the Borg ships ran NeXTstep. That big cube certainly reminded many of us of the NeXT cubes of the time, and the drones walked slow enough to explain the 25MHz processors. If you dwell on it a little more, which of course I did, embedded Linux starts to make sense. The Borg's hardware was widely variant, was collected from many different planets (manufacturers), and it all needed to work together. Linux certainly fits the bill. All of that falls apart, however, when you consider how the Borg replicated themselves. They used little nanoprobes to "infect" people with their virus-like systems. If the Borg are that virus-ridden, they must be running Windows!

All joking aside, that's what we focus on this month—not the Borg, but embedded Linux. Reuven M. Lerner

starts off the issue by embedding the R language into PostgreSQL. Statistical analysis is a complicated beast, and Joe Conway's PL/R functions make things a little easier. Reuven shows how. Dave Taylor follows up with Bash notational shortcuts. Statistically speaking (har har), using shortcuts in your scripts can save time, but it's often at the expense of clarity. Dave discusses how and when to use shortcuts.

Next, Kyle Rankin takes a little time-travel adventure to the days of Telnet. Of course for Kyle, the days of Telnet are yesterday and today. He explains how to use the old standby Telnet protocol for doing some pretty helpful things when troubleshooting anything from big-metal hardware to tiny embedded systems. He even teaches how to send an e-mail with Telnet, which is worth at least ten geek points. My Open-Source Classroom column follows Kyle with a primer on DNS. DNS is usually something you don't think about—until it quits working. This month, I walk through some neat uses

Kevin Bush reviews the ZaTab from ZaReason, which is a fully open tablet computer running CyanogenMod. With this tablet, rooting isn't a bad word.

for DNS and maybe teach you a few things along the way.

Nowadays, when people think of embedded systems, Android is one of the first things that comes to mind. Kevin Bush reviews the ZaTab from ZaReason, which is a fully open tablet computer running CyanogenMod. With this tablet, rooting isn't a bad word. Craig Maloney follows right up with another embedded system, namely Squeezebox. Logitech has created a completely open platform for streaming music around your house, and it uses Linux to do it. Craig shows off this cool system and teaches how to set up your own.

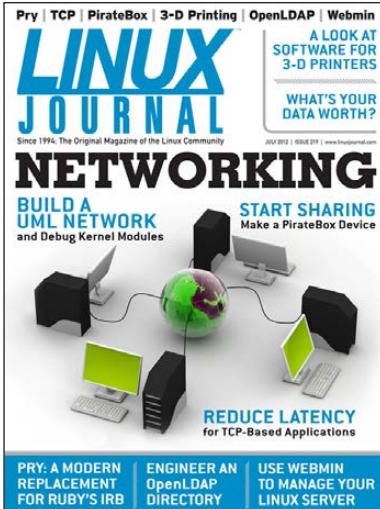
If your idea of embedded Linux looks a little more like wires, solder and printed circuit boards, Edward Comer knows just how you feel. This month, he goes in depth with Arduino. Whether you want to program the embedded code or etch your own circuit board with vinegar and salt, this article is for you. Edward walks through the whole process from planning to implementation, and he proves that a project like this is possible for anyone with the interest and dedication.

Richard Campbell finishes off the issue with an article on NVM. If you've ever considered your fancy new SSD to be too slow, you'll want to read his article. Nothing beats the speed of RAM, so what if RAM were a persistent storage device? Richard explores that idea and talks about the next big thing in the world of storage.

So, whether you want to program your own Borg cube full of Arduino drones or just want to stream some music into your shuttlecraft from the holodeck, this issue is for you. Like every month, however, we also have an issue chock full of things for every flavor of Linux enthusiast out there. We have tech tips, product reviews, kernel news and even a few things just for fun. We hope you enjoy this issue as much as we enjoyed putting it together. Until next month, live long and prosper. ■

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



Dave Taylor's Work the Shell, July 2012

I just wanted to say thanks to Dave for the subshell article!

I have to work

with a lot of .csv files and used to take a lot of time manipulating them with a spreadsheet program, but after learning just a small bit about shell scripting (and a bit of Perl too), I've found that I can trim those large chunks of time to mere seconds and automate out the tedium and drudgery.

Since then, I continually look for as many tips as I can find about shell scripting, so please continue!

—Jeff Shutt

Dave Taylor replies: Thanks, Jeff!

Appreciate hearing from you.

NOOK Subscription

This isn't a complaint, as I know there have been many regarding the switch to the all-digital edition. As a NOOK Color owner, I was delighted to find issues available for purchase in the

NOOK store at a low price. My question is why is there no subscription available through the NOOK store? I would be very happy to purchase a subscription through there. Until then, I'll continue purchasing individual issues, but a subscription through the NOOK store would be preferred.

—Jeffery Mathis

Thanks for your support! We're definitely looking into the NOOK subscription option. Keep in mind though that if you order a subscription through LinuxJournal.com, you will receive a monthly download link that contains .epub, .pdf and on-line versions as well as .mobi. If you order through B&N, you receive only the .epub version.—Ed.

Android App

I have been reading your marvelous magazine for more than 10 years.

I recently got myself a new Android (Samsung Galaxy S2 4G) smartphone, and one of the first applications I installed was yours, so I could read your magazine anytime, day or night. But, I have been having a few issues with the app.

1) I have to enter my e-mail address (40

characters in total) every time I try to access the magazine on-line!

2) I often get the error "Subscription options are unavailable: device is off-line or service is unavailable."

I live in the Federal Capital of Australia - Canberra and have ADSL 2+ at home, and I work for Unisys on federal government accounts, and they have very high-speed Internet access (I can't discuss further for security reasons).

The device is definitely not off-line. It shows 4–5 bars of signal strength at work and at home. I have alternated between Wi-Fi and the phone network for access, but this is really starting to bug me. I have never had any other app issues—LinkedIn, Foursquare, Groupon, Gmail, eBay, Telstra (phone carrier), Amazon, RealEstate.com.au and heaps of other apps all work perfectly.

I can't find a "debug" setting in your app or setup options, such as "save your e-mail address".

How can I "save" my e-mail address? How can I get some debug details from the app to try to resolve the connectivity issue?

—Nigel Grant

Wow, Nigel, I'm really sorry you've had so much trouble with the app. It is certainly supposed to keep your information, and although there were some on-line/off-line issues with early versions of the app, they should be largely solved now. My first recommendation would be to troubleshoot like any other oddly acting app. Try clearing all the cached data (in settings/applications), and if that doesn't help, try deleting and re-installing it. If you've moved the app to the SD card, make sure permissions are correct and so on. Hopefully, it will straighten out for you soon. Be sure to check out the .epub and .mobi versions too; they render nicely on the S2 as well. (I have the same phone myself.)—Ed.

LJ .mobi to Kindle

At Pragmatic Bookshelf, they offer their e-publication in .mobi format (among others). By giving them your Kindle's e-mail address and allowing e-mail from them on your Amazon account, they can send your publications directly to your Kindle. Is this a possibility for LJ?

—Jes D. Nissen

Jes, although our distribution method doesn't currently support a direct e-mail, some fellow readers have scripted automated solutions. Check the past couple issues, or read the next letter from Ward.—Ed.

Script to Send the .mobi Version of LJ to an @kindle.com Address

I've written a Python script that fetches the monthly *LJ* e-mail from an IMAP server, uses the URL inside to download the .mobi version of *LJ* and send that to an @kindle.com address, so that when you sync your Kindle, *LJ* is added automatically. Just add the script to your cron job to run monthly. Feel free to share the script. You can find it at <http://goo.gl/C2IE3>.

—Ward Poelmans

Ward, you rock my face off (which my teenage daughter assures me is a good thing, and not the nightmare-inducing horror scene it sounds like). Seriously, thanks a ton.—Ed.

Transition to All-Digital

I greatly miss *Linux Journal* in hard copy. There is no substitute for having a magazine around that you can just pick up and read anytime, anywhere. The electronic version, despite having some advantages, simply fails to match the hard-copy experience.

Now, I have passed through the grieving process—denial, anger, bargaining, depression—and have come to acceptance. As such, I will continue to subscribe, because *Linux Journal* is part of the Linux ecosystem, and I want the

ecosystem to grow and evolve, even if occasionally some changes displease me.

Since you have gone all electronic, please adapt your style to the new media. It's no longer print, so lose the print format and capitalize on the possibilities of digital. Two columns don't work on displays that don't show a complete page; some of the images need higher resolution (the photos of Reuven and the Silicon Mechanics and iXsystems advertisements in the July issue, for example); cut the publishing cycle to be more timely; don't go overboard with multimedia, but include it when it enhances the exposition and works creatively.

It is going to be hard going for you and your loyal readers, but I am prepared to continue to support *Linux Journal* through this difficult time.

—Gordon Garmaise

Gordon, I understand your process, and even though I had to put on a happy face, I may have had some similar emotions! We have modified the PDF version significantly to look better on devices, yet still retain the magazine look and feel. If you've seen the PDF on a 10" tablet, it's pretty stunning. Hopefully, the .epub and .mobi versions with their flowing text help with other devices.—Ed.

Love the E-Format

I've read numerous rants and raves about the digital format of *Linux Journal*, and I felt the need to chime in with some positive feedback. Personally, I love the digital format. It's the reason I renewed my *LJ* subscription. Almost all the magazines I used to read in paper, I now look for in digital format. It's much more portable, and I no longer have to worry about having stacks of magazine copies piled under the coffee table or lined up on my bookcase. The electronic version is also much more interactive. If a person wants to learn more about a particular article topic or an interesting product in an ad, there's usually a hyperlink to click on. I would like to see all magazines at least offer a digital option for those of us who prefer bytes over tree bark.

—Jim Vaughn

Thanks Jim! It's certainly been a polarizing topic, that's for sure.—Ed.

Regarding My Letter to Dave Taylor in the Last Issue

I just read Dave Taylor's July 2012 column. Keeping in the same vein as before, I want to see if these comments to Dave help.

You refer to how you can build up a command and then save it by saying:

```
!! > new-script.sh
```

This will not save the command. It will run the last command and save its output. If you want to save the last command, you want to say either:

```
!!:p > new-script.sh
```

or:

```
echo "!!" > new-script.s
```

They are almost the same but not quite. By the way, the !! syntax is from the C shell and is a part of Bash as well. It is not a part of the Bourne shell or ksh.

You show how useful subshells can be:

```
newname=$(echo $filename | sed 's/.JPEG/.jpg/')
```

In fact, this is massive overkill. The \$() is a subshell. The echo is built in, so no subshell there. Then, you use sed with all of its footprint that is capable of powerful regex manipulations, for a total of two subprocesses. Instead, how about a simpler approach:

```
newname="${filename/.JPEG/.jpg}" # Look Ma! No subprocesses.
```

How graceful is the continuation of a multi-line command? But, there's no

mention of PS2. Anyone who uses it will not see the default that you have.

Telling people to use a for file in \$(grep -l stuff...) is bad practice. The construct will fail if the filenames have embedded whitespace. Also, it will fail if the list is large, because you will violate the maximum length of a command. (Commands do have a max length.)

The proper way to do it is either to use a while read loop, or to use find | xargs. It's almost always a bad idea to use find -exec:

```
find . -type f -print0 | xargs -0 grep -l stuff
```

The same thing using while read and process substitution might be:

```
while read fn
do
  do_something_with $fn
done <(< find | xargs...)
```

This is exactly two subprocesses.

You mentioned a 250-line script (scale), but you don't tell us where it is so we can see it. But, it bears mentioning that there is a big difference between options and their possible option arguments, and command arguments.

Instead of:

```
scale {args} factor [file or files]
```

your terminology should be:

```
scale [options] factor [file-list]
```

The use of square brackets is to denote optional use. I strongly encourage you to read the Bash man page.

Looking forward to the next one.

—**Steven W. Orr**

Thanks for Covering the Basics

I just read Dave Taylor's column in the July 2012 *Linux Journal* ("Subshells and Command-Line Scripting") and wanted to say thanks. Although Dave's other shell articles provide insight into using the shell, I usually end up planning to read those in depth sometime later, and then don't. I enjoy articles that explain the basics in more depth. While I am not a newbie (I started more seriously when Fedora Core 1 came out), I am not very advanced. My occupation was as a toolmaker at an R&D facility, and my family consumed (I'm now retired) much of my time and thought. I am probably still, at best, a beginning-intermediate Linux adherent/user. I want to develop a much deeper understanding of all aspects of how Linux works, and

how to put it to work, but since my stem-cell transplant, I find it a little more difficult to learn and retain knowledge. I have found this article very helpful, as are those of many of the other *Linux Journal* contributors. Thanks to Dave and all of the others for your efforts. By the way, I used e-mail because I am not really a Web 2.0 kind of guy. My wife and I don't use social media. Thanks for providing this avenue of communications too.

—**jjerome1**

Dave Taylor replies: Thanks for your kind note. It's nice to know people are a) reading and b) appreciating what I write.

Texterity Android App

When *LJ* came out in .epub format, I immediately put it on my Kobo and my Android phone. The Kobo is a tad underpowered, so it's hard to flip through a magazine quickly, so I've been using my Android phone more lately, on a 4.3" screen.

The built-in .epub reader is good, and the .epub reflows nicely, but I decided to try the *LJ* app to see if I like it.

Pros: 1) annotations allow one to jump to a specific article, and 2) pure text mode helps the content fit the screen, which is good for small displays.

Con: it doesn't remember where I left off!

This "con" is huge, an obviously needed feature, and a terrible oversight unless I'm missing some hidden setting in the app. The built-in .epub reader opens right to where I left off. In the *LJ* app, if I jump to another app temporarily and then go back in, the book is closed, and it doesn't know where I was. Even if I go into an annotation, and then click the back button, it doesn't remember where in the page I was. Web browsers work better than this, so this is simply not acceptable. Find a Texterity developer and smack them with a cluebat.

Also, if the Wi-Fi is enabled, when I go back into the app after briefly looking elsewhere, it tries to reload all the books again!

I was in a coffee shop with spotty Wi-Fi and I had to disable my Wi-Fi because every time I looked elsewhere, like my calendar, and went back into the *LJ* app, it insisted on trying to download the issue again!

I'm running it on Gingerbread, so I don't know if these issues are fixed already elsewhere, but I'm amazed that anyone let the software out the door in this condition, as the user experience is not

good compared with a simple .epub reader, which is what I'll be using.

Keep up the good work and sorry for ranting. Maybe it's just me, since I would think that these issues would be serious enough to prevent the release of the app.

—Mike

Thanks for the feedback, Mike. We'll make sure to get the info to the Texterity folks. Sadly, cluebats aren't indigenous to my area, so I may have to smack them with a fruit bat instead.—Ed.

Why Not Have a Happy Funeral for the Paper Version of LJ?

All right, I must admit that during the past few months I have attempted to try to guess which page the Letters to the Editor would end and start reading from there. The only other place I have heard more complaining has been at church—if people thought the song service was too long, too short, not enough traditional songs, not enough new songs, not my newly written song, and I haven't even touched the minister's message critiques!

I have been reading an interesting book titled *How to Change Your Church (Without Killing It)*. An intriguing chapter in the book describes how leaders need to honor areas of ministry that have run

their course before phasing them out. This got me looking back at all of the writers over the months that have been angry about the removal of the paper version of their favorite magazine. I understand the economics of business and am not suggesting you do something that would jeopardize your bottom line, but I do think there is a solution to help those who are angry and disappointed about the absence of their physical magazine. Why not rejoice in the articles of the past? Summon all past subscribers of the print edition to tell you which paper article was the best in their opinion. Rejoice and give credit to those who got their fingers dirty at the printing press. It just may cause some disgruntled folks to come back just because you are allowing them the opportunity to thumb through their old magazines and relive the moments of breaking off work early to get to the mailbox for the magazine. They may start to await the e-mail prompt: "Your Issue of *Linux Journal* has arrived and is ready for download".

If people have a chance to rejoice and relive their positive experiences of the past, they may just come back!

—Dean Anderson

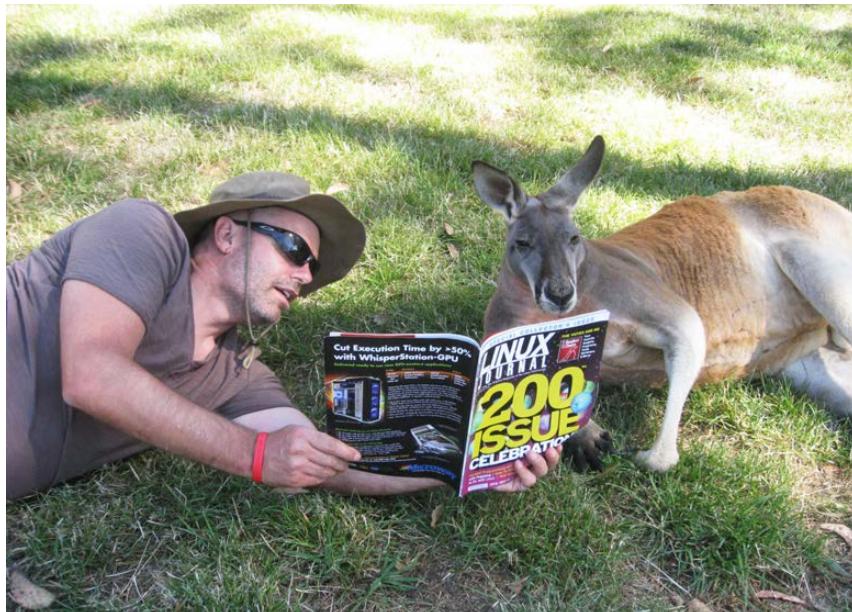
Dean, that's an interesting idea. We'll toss it around a bit and see if we can

come up with something. I still have quite a stack of Linux Journal issues I can't bear to part with. (The arcade machine article in the August 2007 issue was the first time I was ever published. Issue 160 will always have a home on my coffee table! See <http://www.linuxjournal.com/article/9732>.)—Ed.

Photo of the Month

I'd like to share this photo with readers.

—Tim O'Grady



Are you an emacs or vi roo?

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



Go to <http://www.linuxjournal.com/rc2012> to vote in this year's Readers' Choice Awards!

Voting ends September 16, 2012.

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.

diff -u

WHAT'S NEW IN KERNEL DEVELOPMENT

Apparently, **CloudLinux** recently accidentally released a non-GPLed driver that still used GPL-only kernel symbols, and that claimed to the kernel to be GPLed code.

Matthew Garrett noticed this first, and he submitted a patch to cause the kernel to treat the module as non-GPL automatically and deny it access to the GPL-only symbols. Because the violation was so blatant, a number of kernel developers seemed to favor potentially taking legal action against CloudLinux. **Greg Kroah-Hartman** was particularly dismayed, since the driver in question used kernel symbols he himself had created for GPL use only.

At one point the CEO of CloudLinux, **Igor Seletskiy**, explained that this just had been an engineering mistake. They had decided to release their GPLed driver under a proprietary license, which as the copyright holders they were allowed to do, but they mistakenly had failed to update the code to stop using GPL-only symbols. He

promised to fix the problem within the next few weeks and to provide source code to the binary driver that had caused the fuss.

Chris Jones recently asked what the etiquette was for giving his own personal version numbers to his own customized kernels. Greg Kroah-Hartman replied that he was free to do whatever he wanted in that arena without offending anyone. But, Greg suggested that in order for Chris' users to be able to understand the features and capabilities of the particular customized kernels, Chris probably would be served best by relying on the official version numbers and just appending his own afterward—for example, Linux version 3.4-cj2.

Apparently, the number of **patch submissions** coming into the Linux kernel is in the process of exploding at the speed of light, like a new Big Bang. **Thomas Gleixner** remarked on this trend and suggested that the kernel development process might have to be modified to be able to handle such an E=mc²-esque

Greg Kroah-Hartman even said he thought the increase in patch submissions might be an intentional denial-of-service attack....

situation. Greg Kroah-Hartman even said he thought the increase in patch submissions might be an intentional denial-of-service attack, designed to interfere with the ability of the kernel developers to keep developing. Thomas didn't think this was likely, but he did point out that a number of companies assessed the performance of their kernel-hacking employees by the number of patches accepted into the kernel, or the total number of lines of code going into the tree. He speculated that this type of performance evaluation, implemented by companies around the world, might account for the DoS-seeming nature of the situation.

Thomas Gleixner pointed out that there was a problem with kernel code maintainers who pushed their own agendas too aggressively and ignored their critics just because they could get away with it. Greg Kroah-Hartman agreed that this was a tough problem to solve, but he added that in the past, problem

maintainers seemed to go away on their own after a while. **Alan Cox** said that in his opinion, maintainers who seemed to be pushing their own agendas were really just doing development the way they thought it should be done, which was what a maintainer should be doing.

But, Alan did agree that maintainers sometimes were lax on the job, because of having children or illnesses or paid employment in other fields. He suggested that having co-maintainers for any given area of code might be a good way to take up some of the slack.

Another idea, from **Trond Myklebust**, was to encourage maintainers to insist on having at least one "Reviewed-By" tag on each patch submission coming in, so that anyone sending in a patch would be sure to run it by at least one other person first. That, he said, could have the effect of reducing the number of bad patches coming in and easing the load on some of the maintainers.

—**ZACK BROWN**

Steam, Now with Less Wine!



If you've ever run Valve Software's Steam client on your Linux box using Wine, you know that even if you pretend it works well,

it really doesn't. Wouldn't it be great if Valve finally would release a native client for Linux? Thankfully, Valve agrees! On a recent blog post (<http://blogs.valvesoftware.com/linux/steamd-penguins>), the Valve folks verify that they're creating a native Ubuntu 12.04 Steam client. It's bad news for zombies, however, because the first game they're porting to the penguin platform is *Left 4 Dead 2*.

Granted, the Steam client alone doesn't mean the Linux game library will explode overnight, but it does mean game developers will have one more reason to take Linux users seriously. There have been rumors of Steam for Linux for years, but this time, it looks like it really will happen! Stay tuned to the Valve blog for more details.—**SHAWN POWERS**

They Said It

I don't believe in e-mail.
I rarely use a cell phone
and I don't have a fax.

—**Seth Green**

I was just in the middle
of singing a song
about how broke we
were and now my cell
phone rings.

—**Joel Madden**

To be happy in this
world, first you need a
cell phone and then you
need an airplane. Then
you're truly wireless.

—**Ted Turner**

You have to take into
account it was the cell
phone that became
what the modern-day
concept of a phone call
is, and this is a device
that's attached to
your hip 24/7. Before
that there was "leave
a message" and
before that there was
"hopefully you're
home".

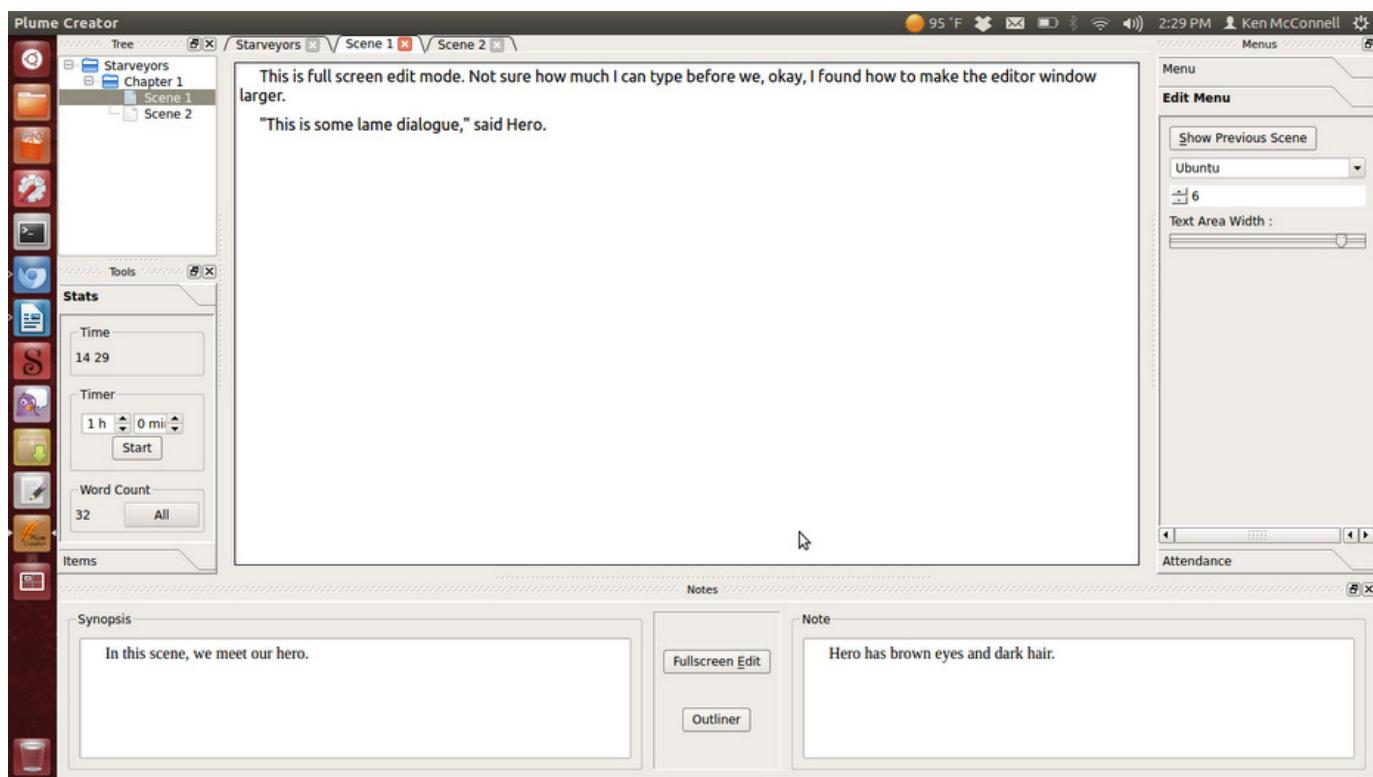
—**Giovanni Ribisi**

You want to see an
angry person? Let me
hear a cell phone go off.

—**Jim Lehrer**

Pluck Out a Novel with Plume

I often discuss the Linux port of Scrivener with my writer friend Ken McConnell. We both like Scrivener's interface, and we both prefer to use Linux as our writing platform. Unfortunately, the Linux port of Scrivener just doesn't compare to the OS X version. The other day, Ken told me about Plume Creator.



(Image Courtesy of <http://www.ken-mcconnell.com>)

With a very similar interface, Plume Creator will feel quite familiar to any Scrivener user. It's very early in development, but it already behaves much nicer than the Linux port of Scrivener. If you've ever wanted to write a novel, or even considered giving NaNoWriMo (<http://www.nanowrimo.org>) a try, Plume Creator is worth a look. Get it today at <http://plume-creator.sf.net>.

—SHAWN POWERS

Extreme Graphics with Extrema

High-energy physics experiments tend to generate huge amounts of data. While this data is passed through analysis software, very often the first thing you may want to do is graph it and see what it actually looks like. To this end, a powerful graphing and plotting program is an absolute must. One available package is called Extrema (<http://exsitewebware.com/extrema/index.html>). Extrema evolved from an earlier software package named Physica. Physica was developed at the TRIUMF high-energy centre in British Columbia, Canada. It has both a complete graphical interface for interactive use in data

analysis and a command language that allows you to process larger data sets or repetitive tasks in a batch fashion.

Installing Extrema typically is simply a matter of using your distribution's package manager. If you want the source, it is available at the SourceForge site (<http://sourceforge.net/projects/extrema>). At SourceForge, there also is a Windows version, in case you are stuck using such an operating system.

Once it is installed on your Linux box, launching it is as simple as typing in **extrema** and pressing Enter. At startup, you should see two windows: a visualization window and an analysis

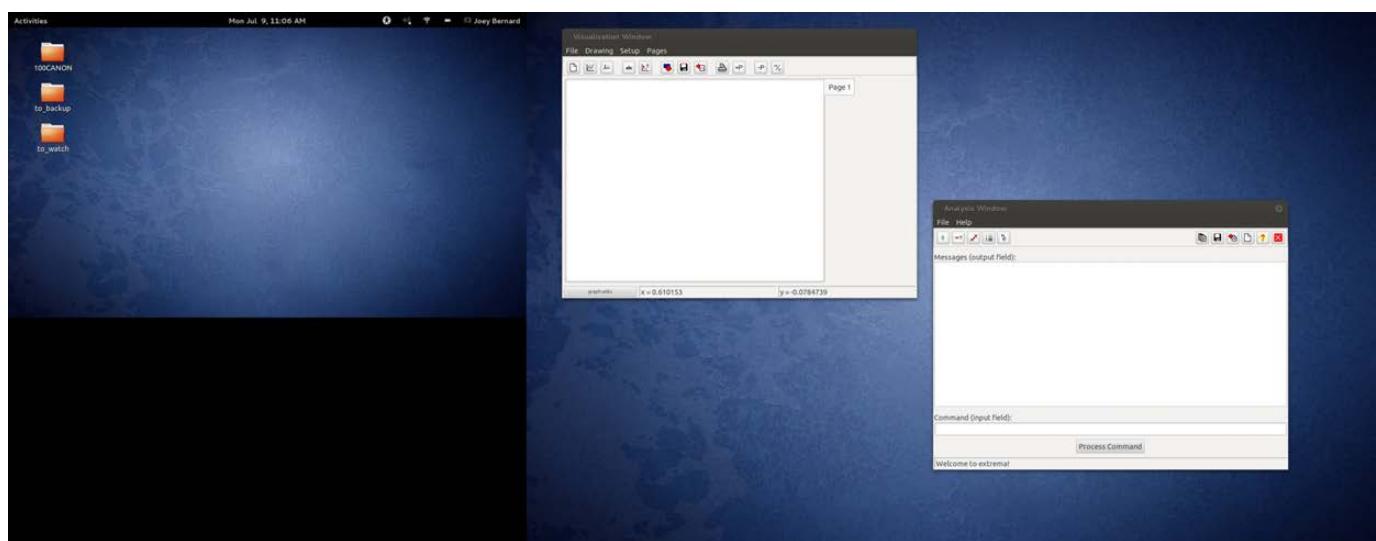


Figure 1. On startup, you are presented with a blank visualization window and an analysis window.

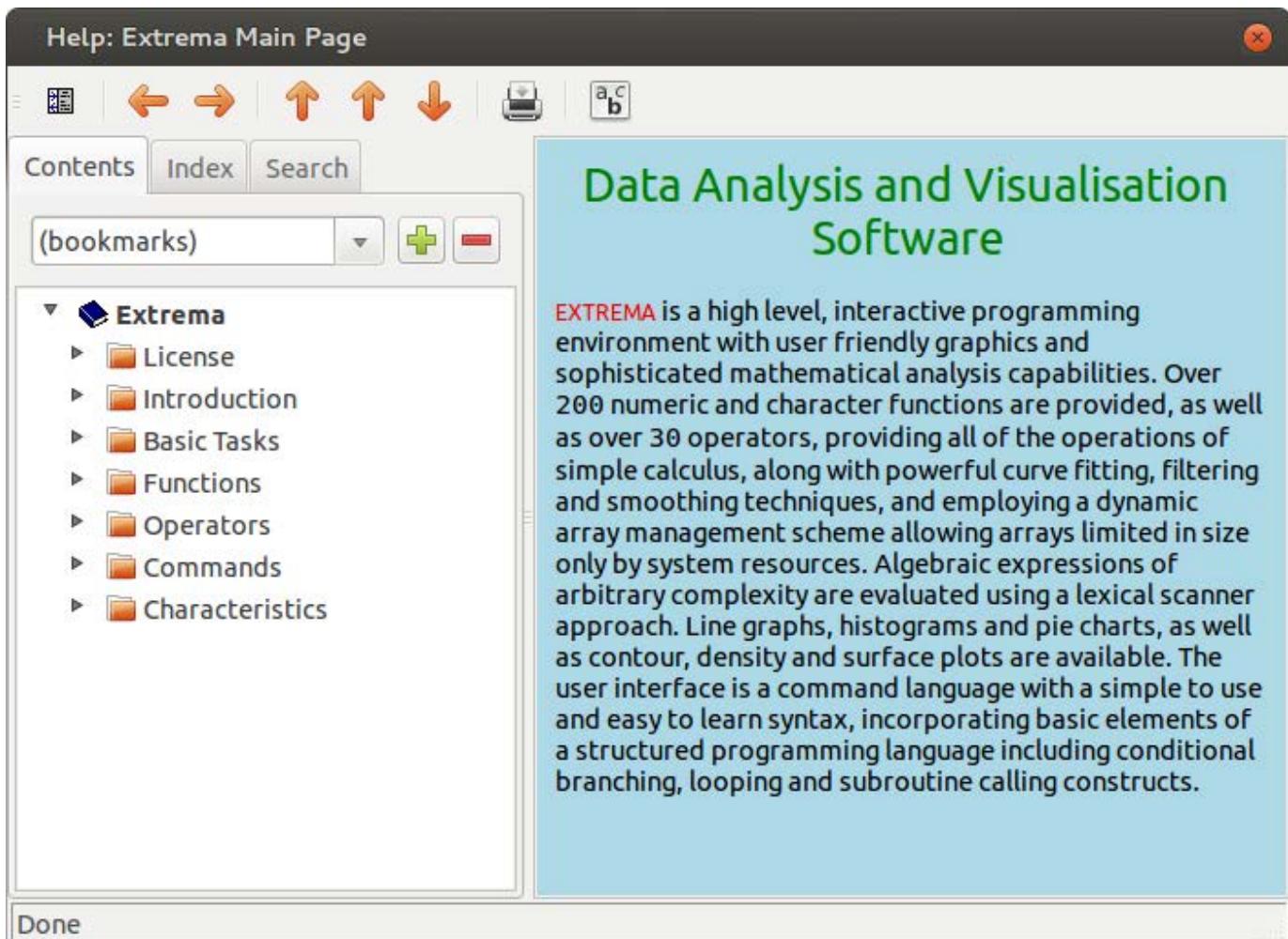


Figure 2. The help window gives you information on all of the available functions, operators and commands.

window (Figure 1). One of the most important buttons is the help button. In the analysis window, you can bring it up by clicking on the question mark (Figure 2). In the help window, you can get more detailed information on all the functions and operators available in Extrema.

Extrema provides 3-D contour and density plots. For 2-D graphing, you can control almost all the features, like axes, plot points, colors, fonts and legends.

You also can do some data analysis from within Extrema. You can do various types of interpolation, such as linear, Lagrange or Fritsch-Carlson. You can fit an equation to your data with up to 25 parameters. Extrema contains a full scripting language that includes nested loops, branches and conditional statements. You either can write out scripts in a text editor or use the automatic script-writing mode that translates your point-and-click actions to

the equivalent script commands.

The first thing you will need to do is get your data into Extrema. Data is stored in variables and is referenced by the variable's name. The first character of a variable name must be alphabetic and cannot be any longer than 32 characters. Other than these restrictions, variable names can contain any alphabetic or numeric characters, underscores or dollar signs. Unlike most things in Linux, variable names are case-insensitive. And remember, function names are reserved, so you can't use them as variable names.

String variables can contain either a single string of text or an array of text strings. Numeric variables can contain a single number, a vector (1-D array), a matrix (2-D array) or a tensor (3-D array). All numbers are stored as double-precision real values. Unlike most other programming languages, these arrays are indexed starting at 1, rather than 0. There are no limits to the size of these arrays, other than the amount of memory available on your machine. Indexing arrays in Extrema can be interesting. If you want the eighth element of array `x`, you simply can reference it with `x[8]`. You can grab elements 8, 9 and 10 with `x[8:10]`. These indices can be replaced with expressions, so you could get the eighth element with `x[2^3]`.

There also are special characters that you can use in indexing arrays. The statement `x[*]` refers to all the values in

the vector. If you want the last element, you can use `x[#]`. The second-to-last element can be referenced with `x[#-1]`.

You likely have all of your data stored in files. The simplest file format is a comma-separated list of values. Extrema can read in these types of files and store the data directly into a set of variables. If you have a file with two columns of data, you can load them into two variables with the statement:

```
READ file1.dat x y
```

You also can read in all of the data and store it into a single matrix with:

```
READ\matrix file1.dat m nrows
```

In order to do this, you need to provide the number of rows that are being read in. You also can generate data to be used in your analysis. If you simply need a series of numbers, you can use:

```
x = [startval:stopval:stepsize]
```

This will give you an array of numbers starting at `startval`, incrementing by `stepsize` until you reach `stopval`. You can use the `GENERATE` command to do this as well. The `GENERATE` command also will generate an array of random numbers with:

```
GENERATE\RANDOM x min max num_points
```

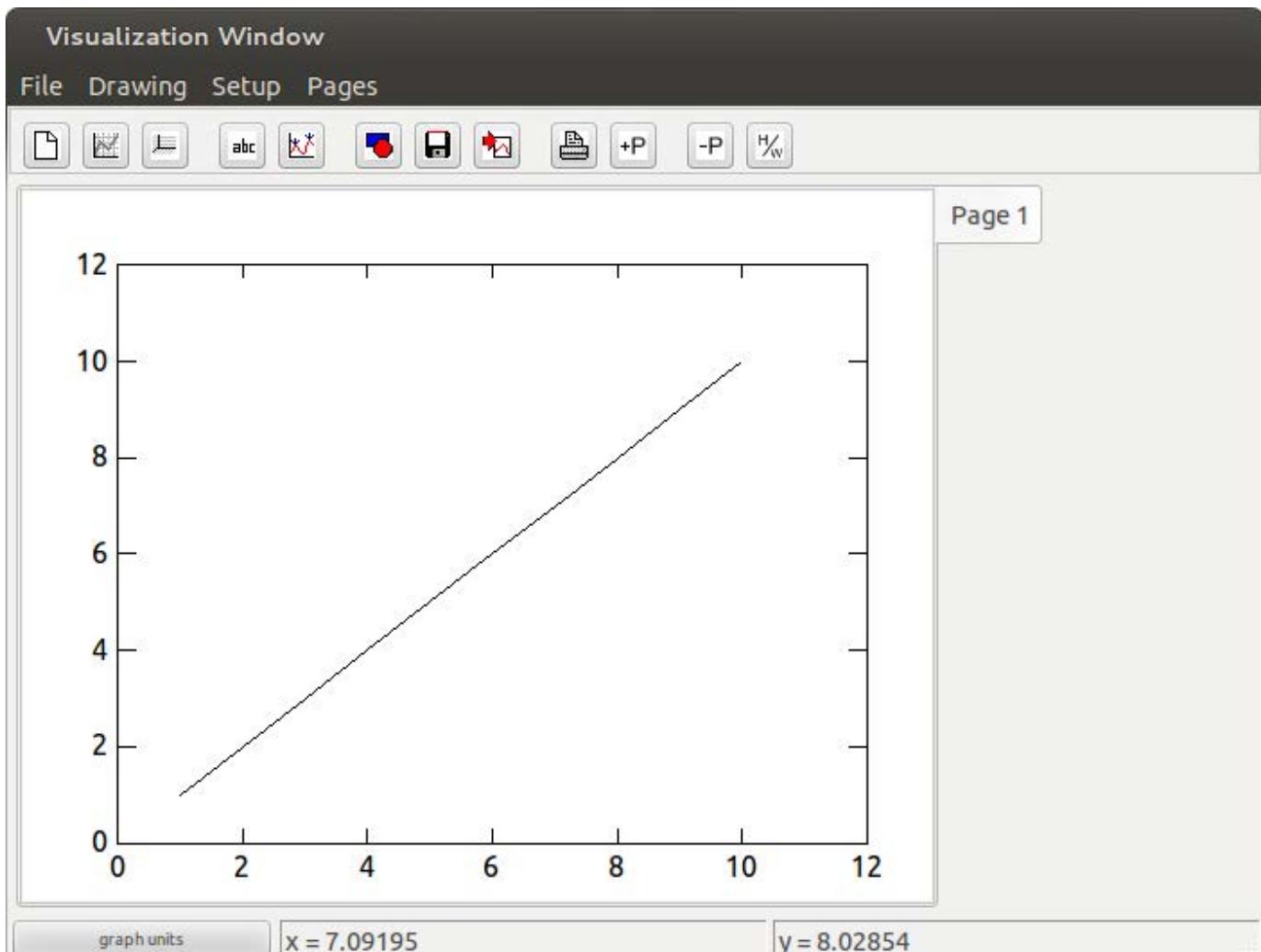


Figure 3. Plotting a Vector of Values

Extrema has all of the standard functions available, like the various types of trigonometric functions. The standard arithmetic operators are:

■ + — addition

■ - — subtraction

■ * — multiplication

■ / — division

■ ^ — exponentiation

■ () — grouping of terms

There also are special operators for matrix and vector operations:

■ >< — outer product

■ <> — inner product

■ <- — matrix transpose

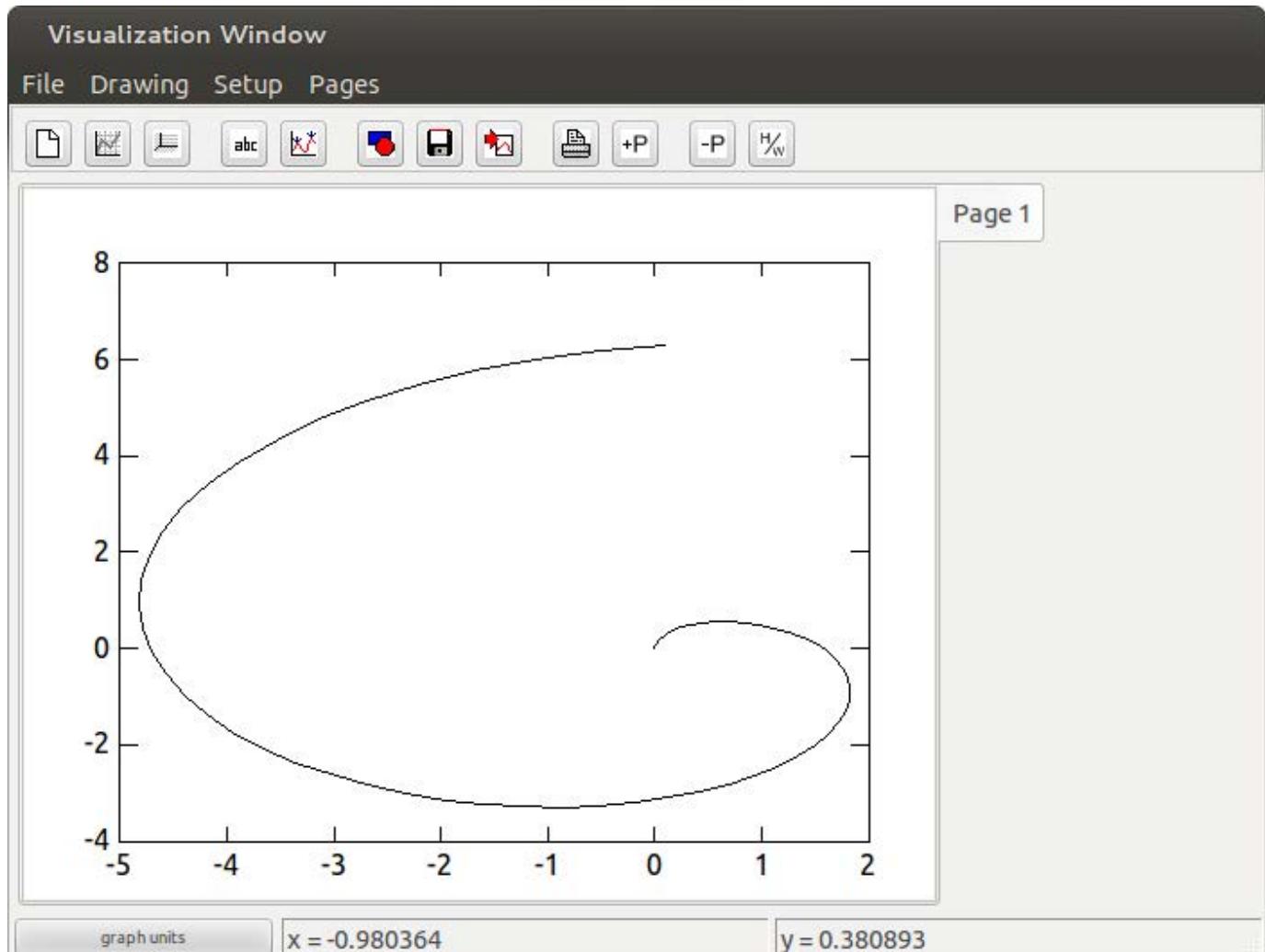


Figure 4. Graphing a Parametric Plot

- >- — matrix reflect
- /| — vector union
- /& — vector intersection

There also is a full complement of logical Boolean operators that give true (1) or false (0) results.

Now that you have your data and have seen some of the basic functions and operators available, let's take a look

at graphing this data and doing some analysis on it. The most basic type of graph is plotting a one-dimensional array. When you do this, Extrema treats the data as the y value and the array index as the x value. To see this in action, you can use:

```
x = [1:10:1]
GRAPH x
```

This plots a fairly uninteresting straight line (Figure 3).

To plot two-dimensional data, you can use:

```
GRAPH x y
```

where x and y are two vectors of equal length. The default is to draw the data joined by a solid line. If you want your data as a series of disconnected points, you can set the point type to a negative number, for example:

```
SET PLOTSYMBOL -1
```

Then you can go ahead and graph your data.

Parametric plots also are possible. Let's say you have an independent variable called t that runs from 0 to 2*Pi. You

then can plot $t * \sin(t)$ and $t * \cos(t)$ with:

```
t = [0:2*pi:0.1]
x = t * sin(t)
y = t * cos(t)
graph x y
```

This will give you the plot shown in Figure 4.

In scientific experiments, you usually have some value for error in your measurements. You can include this in your graphs as an extra parameter to the graph command, assuming these error values are stored in an extra variable. So, you could use:

```
graph x y yerr
```

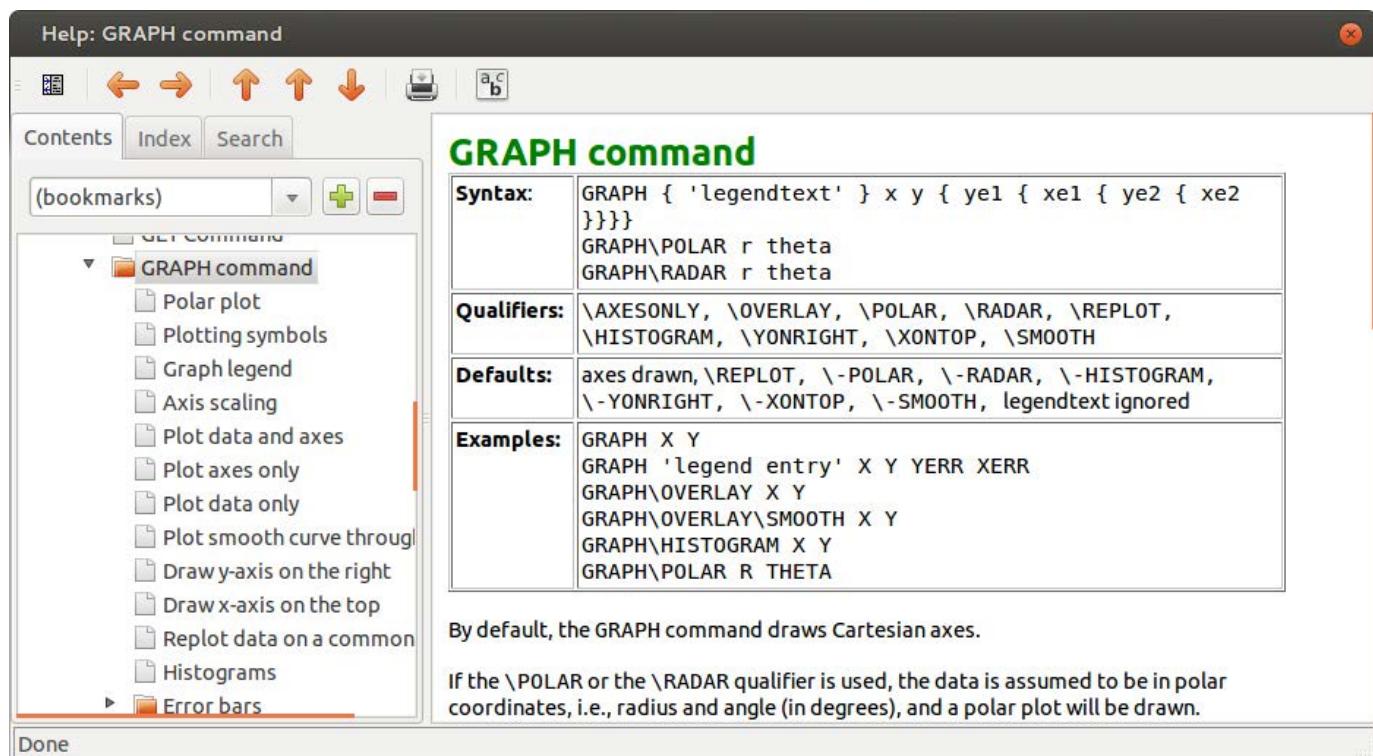


Figure 5. The graph command has many available options.

to get a nice plot. Many options are available for the graph command (Figure 5).

More complicated data can be graphed in three dimensions. There are several types of 3-D graphs, including contour plots and surface plots. The simplest data structure would be a matrix, where the indices represent the x and y values, and the actual numbers in the matrix are the z values. If this doesn't work, you can represent the separate x, y and z values with three different vectors, all of the same length. The most basic contour graph can be made with the command:

CONTOUR m

where m is the matrix of values to be graphed. In this case, Extrema will make a selection of "nice" contour lines that create a reasonable graph.

You can draw a density plot of the same data with the density command, where the values in your matrix are assigned a color from a color map, and that is what gets graphed. Unless you say differently, Extrema will try to select a color map that fits your data the best. A surface plot tries to draw a surface in the proper perspective to show what surface is defined by the z values in your data.

Let's finish by looking at one of the more important analysis steps, fitting an equation to your data. The point of much of science is to develop equations that describe the

data being observed, in the hope that you then will be able to predict what you would see under different conditions. Also, you may learn some important underlying physics by looking at the structure of the equation that fits your data. Let's look at a simple fitting of a straight line. Let's assume that the data is stored in two vectors called x and y. You'll also need two other variables to store the slope and intercept. Let's call them b and a. Then you can fit your data with the command:

```
SCALAR\FIT a b
FIT y=a+b*x
```

Then, if you want to graph your straight line fit and your data, you can do something like:

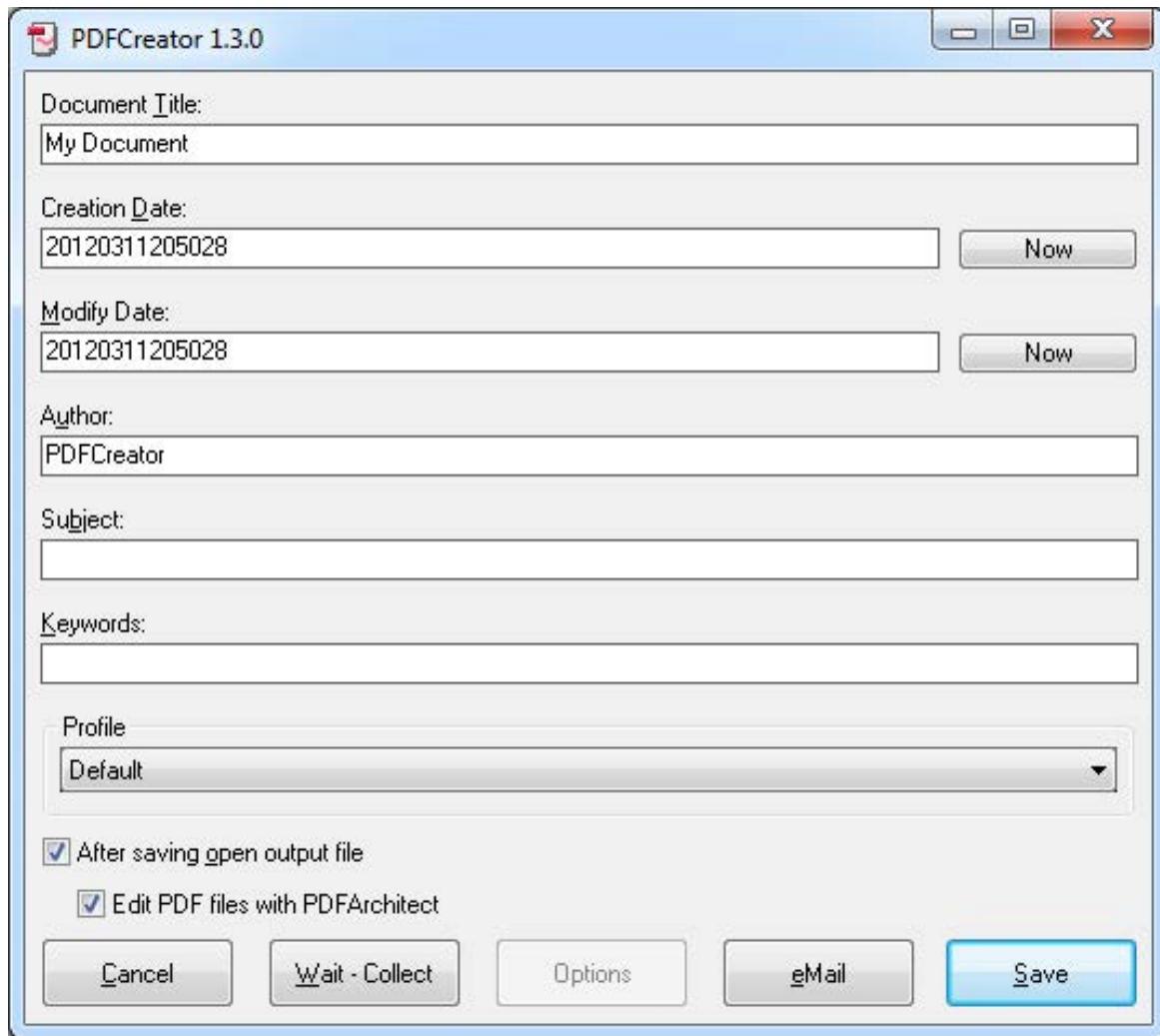
```
SET PLOTSYMBOL -1
SET PLOTSYMBOLCOLOR RED
GRAPH x y
SET PLOTSYMBOL 0
SET CURVECOLOR BLUE
GRAPH x a+b*x
```

Now that you have seen the basics of what Extrema can do, hopefully you will be inspired to explore it further. It should be able to meet most of your data-analysis needs, and you can have fun using the same tool that is being used by leading particle physicists.

—JOEY BERNARD

Non-Linux FOSS

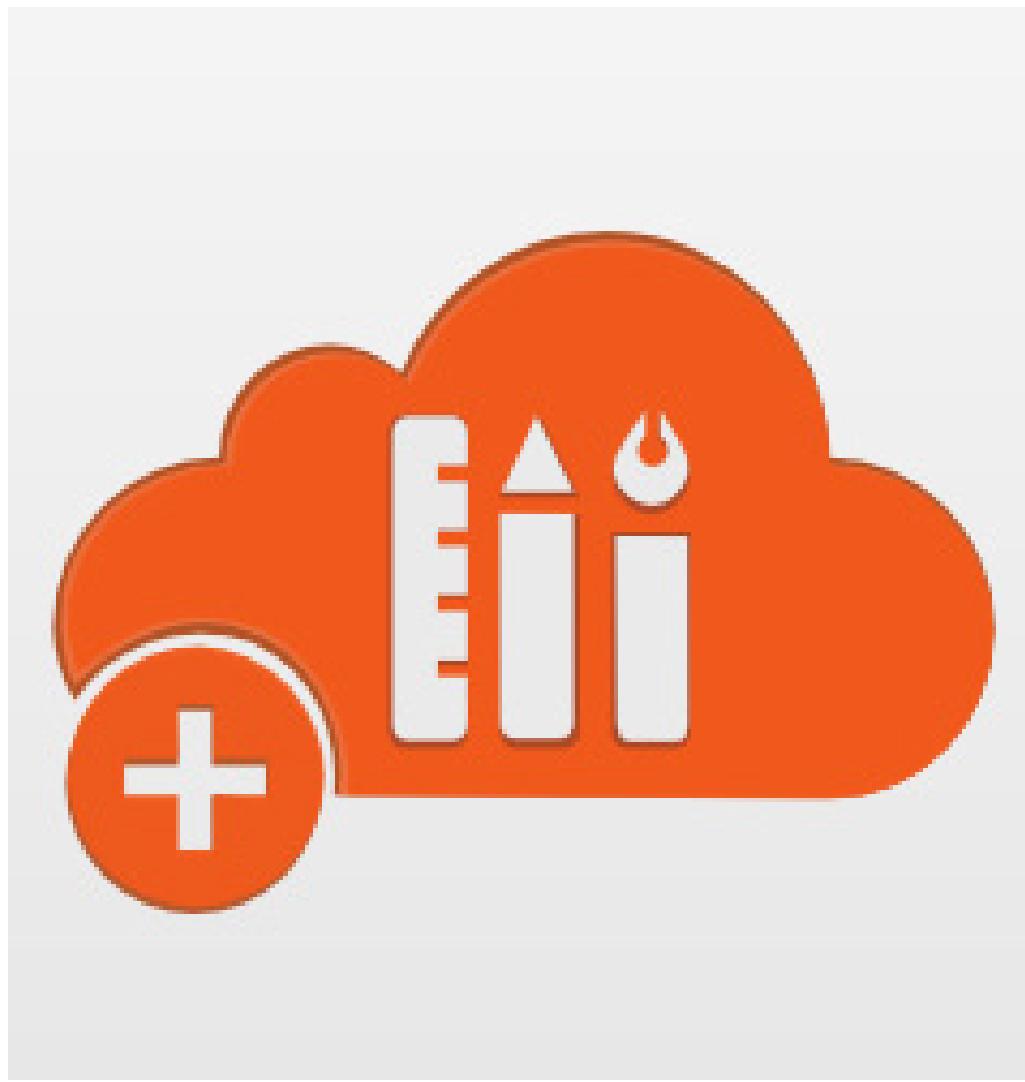
As a LibreOffice user, and an OpenOffice.org user before that, the idea of printing to a PDF is nothing new. If you're stuck on a Windows machine, however, it's not always easy to "be green" by printing to a digital file. Thankfully, there's the trusty PDFCreator package. PDFCreator installs like any other program in Windows and then creates a virtual printer that any program can use to generate PDF files.



(Image from <http://www.pdfforge.org>)

As it has matured, PDFCreator has gained a bunch of neat features. Whether you want to e-mail your PDF files directly, sign them digitally or even encrypt them, PDFCreator is a great tool. Check it out at <http://www.pdfforge.org>.—SHAWN POWERS

There's an App for That



The concept of standalone Web apps isn't new. Anyone using Prism with Firefox or Fluid with OS X understands the concept: a browser that goes to a single Web site and acts like a standalone application—sorta.

With Fogger, however, Web applications take on a whole new

meaning. Using a variety of desktop APIs and user scripts, applications created with the Fogger framework integrate into the Linux desktop very much like a traditional application. If you've ever found Web apps to be lacking, take a look at Fogger; it makes the Web a little easier to see: <https://launchpad.net/fogger>.

—SHAWN POWERS

MATCH YOUR SERVER TO YOUR BUSINESS. ONLY PAY FOR WHAT YOU NEED!



With a 1&1 Dynamic Cloud Server, you can change your server configuration in real time.

- Independently configure CPU, RAM, and storage
- 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 10 for unlimited domains, reseller ready
- Up to 99 virtual machines with different configurations

1&1 DYNAMIC CLOUD SERVER

SAVE \$180

\$34.99/ month first year, base configuration only (regularly \$49.99).



- **NEW:** Monitor and manage your cloud server through 1&1 mobile apps for Android™ and iPhone®.



www.1and1.com

*Offer valid for a limited time only. First year \$34.99/ month only applies to base configuration. Base configuration includes 1 processor core, 1 GB RAM, 100 GB Storage. 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.



REUVEN M.
LERNER

PL/R

Perform powerful statistical analysis by embedding the R language in PostgreSQL.

I took two introductory statistics classes in graduate school and found that I really liked the subject. It wasn't always intuitive, but it always was interesting, and it really helped me to put research, polling and many newspaper stories in a new light. I don't do statistical analysis every day, but it's a powerful tool for organizing and working with large sets of data, and for finding correlations among seemingly disparate pieces of information.

For the courses I took, my university told me to buy SPSS, a commercial program that helps you perform statistical calculations. Being an open-source kind of guy, I discovered R—a programming language aimed at helping people solve problems involving calculations and statistics. R is a full-fledged language, and it theoretically can be used in a wide variety of situations. But, it was designed primarily for use in mathematics and statistical work, and that's where it really shines.

I managed to get through the class just fine using R instead of SPSS. The

quality of R's documentation and the intuitive feel of the language, especially for someone experienced with Ruby and Python, meant that when my instructors demonstrated how to do something in SPSS, I managed to find the appropriate parallel in R, and even get a result before they had finished their explanation.

I have continued to use R during the past few years, both as I've progressed with my dissertation research and even on some client projects. I've used R to analyze data from text files (typically in CSV format), and I've used R to analyze data from databases, using the client packages available for various databases.

Perhaps the most intriguing place I've recently seen R, and where I've started to experiment with it in my own work, is inside PostgreSQL. PostgreSQL, long my favorite relational database, has for many years allowed for the creation of user-defined functions, similar to "stored procedures" in other databases. Whereas most databases provide a single language in which people can write their functions, PostgreSQL makes it possible to connect

If you ever have used R, the possibility of using such a powerful statistics package inside your database should seem like a natural and powerful combination.

nearly any language to the database. By default, PostgreSQL lets you write server-side functions in PL/PgSQL, and there long has been support for PL/Perl and PL/Python.

Since 2003, developer Joe Conway has maintained PL/R, allowing you to write server-side functions in PostgreSQL, using R. If you ever have used R, the possibility of using such a powerful statistics package inside your database should seem like a natural and powerful combination. Rather than having to read the data into R outside PostgreSQL, you suddenly can have R work directly on the results of queries, without needing to use a separate client application or process.

In this article, I introduce the basics of PL/R. This combination isn't for everyone, but with the growing (and welcome) popularity of PostgreSQL among Web developers, and with the increasing need for analysis of information gathered from Web users, it seems to me that PL/R could be an important tool for many developers.

Introduction to R

The home page for R is <http://r-project.org>. From that site, you can download versions

of R for a variety of operating systems, including Linux. (I was able to install R on my server running Ubuntu with `apt-get install r-base-core`, which installed a large number of dependent packages.) New versions of R come out every few months and normally are installed in two versions: the R language and environment and the runtime necessary to execute programs written in R, known as "Rscript". To use the language interactively, just type `R` at the shell prompt.

R is an interpreted, dynamic language. It has some object-oriented features as well. At the core of R is the vector data type, which can contain any number of elements of the same type (known as "mode" within R). Even items that are seemingly scalar values actually are vectors of length 1. Thus, you don't really have integers in R, but rather one-element vectors of type `integer`. You can create multi-element vectors with the `c()` function, handing it the values you would like it to contain:

```
> c(1,5,9,3)
[1] 1 5 9 3
```

The beauty of a vector is that mathematical operators are applied to all of its elements.

This returns a vector, but now the value is lost. If you want to capture it, you must do so in a variable, using the assignment operator <-:

```
> x <- c(1,5,9,3)
```

Note that <- is the standard assignment operator in R. You can, in many places, use the more traditional = operator instead, but it is frequently recommended in the documentation that you use <- to avoid problems in certain circumstances. (I must admit, I've never experienced any problems, although I do try to use <- whenever possible.)

The beauty of a vector is that mathematical operators are applied to all of its elements. Thus:

```
> x + 5  
[1] 6 10 14 8
```

```
> x * 3.5  
[1] 3.5 17.5 31.5 10.5
```

```
> y <- x * 8.6  
> y  
[1] 8.6 43.0 77.4 25.8
```

If you're wondering what the [1]

means on the left side of the vector output, that's an indication of the starting index of the data you're looking at. R, like FORTRAN, but unlike most other languages I use, uses 1 as the index of the first element of an array. Thus, if I want to retrieve the value 43 from y, I need to use index 2:

```
> y[2]  
[1] 43
```

Note that retrieving the element at index 2 doesn't give me a scalar value, but rather a one-element vector.

Vectors are nice, but they can hold only one data type at a time. Because R is dynamically typed, it cannot stop you from entering inappropriate values. Rather, it'll cast all of the values to the best possible common type, if they're different. So if you say:

```
> x <- c(1,2,"abc",3)  
> x  
[1] "1"    "2"    "abc"  "3"
```

notice how all the values in this vector have been turned into strings, in order to ensure that the vector's mode is of type "character".

R allows you to create multidimensional vectors, known as matrices. To create a matrix, just invoke the `matrix()` function, giving it a vector as a first parameter and either the `nrow` parameter or the `ncol` parameter:

```
> m <- matrix(c(1,2,3,4,5,6), nrow=2)
> m
 [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

If you want to grab the first column—a vector, of course—you can do so:

```
> m[,1]
[1] 1 2
```

You also can grab the second row:

```
> m[2,]
[1] 2 4 6
```

Similar to a matrix, but containing vectors of different types, is a “data frame”. When you create a data frame, you assign a name (given as the parameter `name`) to a vector. The vectors then are displayed in parallel, such that index 2 in each vector can be read across, much like a spreadsheet or database table. For example:

```
> names <- c('tom', 'dick', 'harry')
```

```
> scores <- c(90, 60, 99)
> height <- c(180, 160, 190)
> d <- data.frame(names=names, scores=scores, height=height)
> d
   names scores height
1  tom     90    180
2  dick    60    160
3 harry    99    190
```

You can think of a data frame as almost a database table. Not surprisingly, when you execute a PL/R function inside PostgreSQL, you can retrieve the contents of a table into a data frame and then manipulate it.

Now, in this example, you can see that the scores of the three students would appear to be correlated with their heights—such that the taller the students, the higher their scores. One of the first things you learn in statistics, of course, is that correlation doesn’t imply causality, so I should note that I’m definitely not trying to say taller people are smarter! But you can find, at least in the data sample here, a correlation between height and score. This is the sort of thing that R does, and does very well. The easiest way to find the correlation is to run a simple regression—meaning, to find the best possible line that will connect these dots, if “height” is the independent (x) variable and “scores” is the dependent (y) variable. In R, you

would express this as:

```
> lm( scores ~ height, data=d)
```

Call:

```
lm(formula = scores ~ height, data = d)
```

Coefficients:

(Intercept)	height
-151.714	1.329

You can do even better than this though. You can assign the output of your call to `lm()` into a variable. This variable, like everything in R, will then be an object on which you can perform additional calculations:

```
score.lm <- lm( scores ~ height, data=d)
```

This object contains the information you need to know in order to predict people's scores based on their heights (assuming there is a correlation, of course, which I'm not at all claiming there is, outside this contrived example). You then can do this:

```
> intercept <- coefficients(score.lm)[1]
> slope <- coefficients(score.lm)[2]

> new.student.height <- 157
> predicted.score <- intercept + (slope * new.student.height)
> predicted.score
(Intercept)
56.87143
```

Now, if you're trying to predict test scores based on student height, you're likely to be disappointed. But, perhaps you're trying to predict other things—for example, the number of pages people will click on if they came from a particular search keyword, or the likelihood users will purchase something from you, if they initially came to your site during lunch hour rather than at night. These are the simplest sorts of questions you can try to answer with a statistical regression, and as you begin to work with such data, you see more and more opportunities for analysis.

Now, it's possible to do this analysis in many different ways. Google Analytics is a highly popular (and powerful) platform for making certain correlations. And, of course, you always can dump your database into CSV format and then read it into R or another package for analysis. But what PL/R lets you do is run all of this analysis on your database itself, with a language (unlike PL/PGSQL) that is optimized for fast mathematical analysis.

One of the most important parts of R is CRAN, an analog to Perl's CPAN, Python's PyPi and RubyGems—an extensive set of open-source packages on a wide variety of subjects, which implement functionality you might well want to use. For example, my dissertation research

involves understanding what sorts of social networks were created among users of the software I created; using a CRAN project called statnet, such analysis becomes relatively easy to do.

Installing PL/R

Installing PL/R is possibly the hardest part of working with PL/R, although it has gotten significantly easier since the creation of the “extension” system in PostgreSQL. First, make sure you have installed the latest versions of both R and PostgreSQL. Earlier versions will work, but particularly in the case of PostgreSQL, a modern version will be better, thanks to the extension system. I assume in this article that you are using PostgreSQL 9.1.

Now, you need to set the R_HOME environment variable. This variable will tell the PL/R compilation and extension mechanism where to find R’s header and library files. On my Ubuntu server, after installing R via apt-get, I set R_HOME to /usr/share/R:

```
export R_HOME=/usr/share/R
```

Once you’ve set that up, you can download the PL/R source code. At the time of this writing, the latest version is 8.3.0 and is available from the PL/R home page. Then, as the instructions indicate, go into the plr directory that results from

opening the .tar.gz, and type:

```
USE_PGXS=1 make  
USE_PGXS=1 make install
```

Note that this doesn’t install PL/R into any of your PostgreSQL databases. Rather, it makes the PL/R extension available, such that you then can create the PL/R extension inside any database that would like to benefit from it.

After installing the extension, I went into my personal PostgreSQL database

System on Module

New - SoM-3517

- TI ARM Cortex-A8 600 MHZ Fanless Processor
- Up to 512 MB of DDR2 SDRAM
- Up to 1GB of NAND Flash
- Up to 2GB of eMMC Flash
- 2 High Speed USB 1.1/2.0 Host ports
- 1 High Speed USB 2.0 OTG port
- 4 Serial Ports, 2 I2C and 2 SPI ports
- Processor Bus Expansion
- 10/100 BaseT Fast Ethernet
- CAN 2.0 B Controller
- Neon Vector Floating Point Unit
- 24-bit DSTN/TFT LCD Interface
- 2D/3D Accelerated Video w/ Resistive Touch
- Small, 200 pin SODIMM form factor (2.66 x 2.375")



Windows CE



2.6 KERNEL

The SoM-3517 uses the same small SODIMM form-factor utilized by other EMAC SoM modules and is the ideal processor engine for your next design. All of the ARM processor core is included on this tiny board including: Flash, Memory, Serial Ports, Ethernet, SPI, I2C, I2S Audio, CAN 2.0B, PWMs, Timer/Counters, A/D, Digital I/O lines, Video, Clock/Calendar, and more. The SoM-3517 additionally provides a math coprocessor, and 2D/3D accelerated video with image scaling/rotation. Like other modules in EMAC’s SoM product line, the SoM-3517 is designed to plug into a custom or off-the-shelf Carrier board containing all the connectors and any additional I/O components that may be required. The SoM approach provides the flexibility of a fully customized product at a greatly reduced cost. Contact EMAC for pricing & further information.

<http://www.emacinc.com/som/som3517.htm>



EMAC, inc.
EQUIPMENT MONITOR AND CONTROL

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

(named “reuven”, same as my user name) and invoked:

```
SELECT * from pg_available_extensions;
```

I could tell that the extension had been installed correctly, because one of the output rows from this query contained `plr`. Thus, I was able to install it with:

```
CREATE EXTENSION plr;
```

PostgreSQL responded with `CREATE EXTENSION`, meaning that the query was successful.

Using PL/R

Now that PL/R has been installed, what can you do with it? Since you installed it for the purpose of writing functions, the natural thing to do is...write a function. For example, here’s a PL/R function that multiplies two numbers:

```
CREATE OR REPLACE FUNCTION mult(num1 INTEGER, num2 INTEGER)
RETURNS INTEGER AS
$$
return(num1 * num2);
$$ LANGUAGE plr;
```

If you ever have written a PL/PgSQL function before, you’ll recognize the general outline of the function-creation syntax. But between the `$$` quotation symbols, instead of PL/PgSQL, you

instead have an R function. Because you’re not in the normal R environment, you don’t have the normal R function assignment or parameters, and you do need to specify a return type. But the function works just fine, and someone using this function doesn’t need to know that it was written in R:

```
reuven=# select mult(50, 20);
mult
-----
1000
(1 row)
```

Where PL/R really comes into its own is when you have data that needs R-type analysis. For example, let’s put the same score-height data into a database table:

```
CREATE TABLE Students (
name TEXT,
score INTEGER,
height INTEGER
);
```

```
INSERT INTO Students (name, score, height)
VALUES ('tom', 90, 180),
       ('dick', 60, 160),
       ('harry', 99, 190);
```

If you can get this data from a PostgreSQL table into an R data frame, you can perform a regression on the data, returning the slope of the intercept line:

```

CREATE OR REPLACE FUNCTION score_height_slope() RETURNS TEXT AS
$$
students <- pg.spi.exec("select name, score, height FROM
➥students");
score.lm <- lm(score ~ height, data=students);
return(score.lm[[2]]);
$ LANGUAGE PLR;

```

Now, note that in this case, you're not running a regression directly on the data in the table. Rather, the table data is read into R, which creates a data frame, on which you run the regression. However, the ease with which you can do this, and the way in which the SQL query (using the pg.spi.exec function) can retrieve database information and stick it in a data frame, makes all the difference. If retrieving all of the data in one fell swoop would be a problem, you might prefer to use PL/R's other implementations of the SPI (server programming interface) API for PostgreSQL, including support for

working with cursors.

Conclusion

PL/R is one of those ideas I never would have understood if I had encountered it years ago, but now, given my exposure to (and use of) statistics, there are many ways I can foresee using it. There are some limitations; PL/R functions cannot easily call other PL/R functions, and data types don't always match up as you might expect with their PostgreSQL counterparts. But PL/R offers support for such advanced PostgreSQL features as aggregates and window functions, allowing you to produce all sorts of sophisticated reports and analysis.■

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.

Resources

The home page for PL/R is <http://www.joeconway.com/plr>. This site includes downloadable source code, documentation and even a small wiki.

The home page for PostgreSQL is <http://postgresql.org>. The home page for the R language, including the CRAN repository of third-party R packages and the twice-yearly R journal, is at <http://r-project.org>.

If you are interested in learning more about R, there are a number of good tutorials on-line. One printed book that I enjoyed reading, and which taught me a great deal, is *Art of R Programming*, written by Norman Matloff and published by No Starch Press. If you're interested in R, and have experience in other programming languages, I recommend reading this book.



DAVE TAYLOR

Bash Notational Shortcuts: Efficiency over Clarity

Are shell scripts inevitably antiquated? Is Dave writing Bourne shell scripts for UNIX, not even writing about Linux? Read on to find out about his latest letter from a reader and subsequent explanation of his philosophy of writing scripts for publication.

I get letters. Well, I don't get very many letters, truth be told, but I do occasionally get interesting dispatches from the field, and a recent one took me to task for writing about UNIX, not Linux, and for focusing on the Bourne shell, not Bash. Which is odd.

When you're on the command line or writing a shell script, things are pretty darn similar across Linux and UNIX due to the POSIX standard that defines syntax, notational conventions and so on. And in terms of the wealth

of commands available? My experience is that if you rely on "nonstandard" Linux commands like some of the more-sophisticated GNU utilities, you might find yourself in a right pickle when only the more lobotomized versions are available on a job site or with particular hardware that, yes, might be running a flavor of UNIX. It happens.

Same with Bash versus Bourne shell. Although because I do write about shell functions and various other more-advanced features, and because I never

When a script or program looks like your cat ran across the keyboard, it might be very efficient, but it's sure hard to debug later, even if it's your own code.

test the scripts in this column against Bourne Shell (not being Jason, after all), and well, just because I'm not using your favorite Bash features and shortcuts, that doesn't mean I'm using that "other" shell, does it?

The most valuable part in the letter was to remind me that there are some slick notational conventions that are added to modern Bash shells that can clean up some of our conditional statements and structures. It was a good reminder—old dog, new tricks, and all that. Let's have a look.

Shortening Conditional Tests

One of the first programming languages I learned was APL. You probably haven't even heard of it, but it was a remarkably powerful language characterized by lots of special notations that gave you the ability to produce sophisticated results in just a line or two. The problem was, no one could debug it, and the common belief was that it was faster to rewrite a program than to figure out what

another programmer was thinking.

This recurred the first time I looked at Perl, and I even said so to Larry Wall when we bumped into each other years ago. When a script or program looks like your cat ran across the keyboard, it might be very efficient, but it's sure hard to debug later, even if it's your own code.

And onward to Linux. When working on shell scripts, you're used to seeing single brackets for conditional expressions, like this:

```
if [ -n $value ] ; then
```

What I haven't explained is that every time you write a conditional in this form, it actually invokes a subshell process to solve the equation. Write it with double brackets, however:

```
if [[ -n $value ]] ; then
```

and you'll force the test to remain within the shell itself, which will make your scripts faster and more efficient.

There's also some benefit in terms of strict quoting of arguments in expressions too, because they don't have to be handed to a subshell, you can often get away with sloppier quoting using the `[[]]` notation.

The question is, how much faster is it, and is it worth making your scripts just a bit more obfuscated, particularly for us old dogs who are used to the `[]` notation? On the vast majority of systems, in the vast majority of cases, I don't think it speeds things up much at all. By their very nature, shell scripts aren't written to be maximally efficient. If you need lightning performance, there are better—albeit more complicated—languages you can use, like C++ or even Perl. Just keep your cat away from the keyboard.

The same goes for another notational convention that I eschew in the interest of writing maximally clear and readable script code. It turns out that a conditional statement like:

```
if [ -n $value ] ; then
    echo value $value is non-zero
fi
```

also can be written more succinctly as:

```
[ -n $value ] && echo value $value is non-zero
```

In this situation, `&&` means "if the

previous command had a 'true' exit status, do the next one" and `||` means the opposite, as in:

```
[ -n $value ] || echo value $value has a length of zero
```

More efficient? Certainly if we use `[[]]` instead of the single brackets we have now, but is it worth the obfuscation? Perhaps in code that you're delivering to a client or that you are writing as a fast throwaway script for a specific task, but the code I publish here needs to be easily understood. Then we weave in efficiency.

To get a sense of how long I've been chewing on how to write legible, easily understood code, I'll just say that when I first started coding in Fortran-77, I loved that you could have spaces in variable and function names, letting me write code that was even more like an algorithm instead of a complicated program.

Variable Expansion Tricks

Speaking of tricks and cats running across keyboards, I've also avoided some of the really complicated `{}$` notational options in the interest of having my scripts be as widely portable as possible. For example, I tend to write:

```
length=$(echo $word | wc -c) ; length=$(( $length - 1 ))
```

Indeed, across all of these shortcuts and modern tweaks to the Bash shell, which are better?

It's clunky and admittedly inefficient. A smarter way to do it is:

```
length=$(( ${#word} ))
```

It turns out that the \${# notation produces the number of characters in the value of the specified variable. That's easy!

If you look at the Bash man page, you'll know that there are dozens of different syntactic shortcuts like this. Remembering which is which, when for the majority of you readers shell script programming is a useful additional skill, not your main job, is probably more trouble than it's worth.

Don't believe me? All right, what does this do?

```
echo ${value^^}
```

I'd never seen this notation before this particular reader sent me his message, but it turns out that in Bash4 (not earlier versions of Bash), it transliterates lowercase to uppercase. That's something I'd usually write like this:

```
$(echo $value | tr '[:lower:]' '[:upper:]')
```

Or, a slight variation that taps into the modern <<< notation:

```
$(tr '[:lower:]' '[:upper:]' <<< $value)
```

Which is better? Indeed, across all of these shortcuts and modern tweaks to the Bash shell, which are better?

I'll let you tell me, dear reader. 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?

And, needless to say, keep those letters coming, whether you agree with what I'm writing or vehemently disagree. We have asbestos inboxes here at *Linux Journal* and always want to hear what you're thinking! [Send your Letters to the Editor via <http://www.linuxjournal.com/contact.>■

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>.



KYLE RANKIN

Troubleshooting with Telnet

Dust off that telnet command and communicate with a server with raw plain-text commands—it's good for the soul.

Poor telnet, it used to be the cool kid on the block. It was the program all sysadmins turned to when they needed to connect to a remote server. Telnet just wasn't that good at keeping a secret—all communication went over plain text—so administrators started switching to SSH for encrypted remote shell sessions. Of course, along with the switch came a huge stigma against administrators who still used telnet. Eventually, telnet became an outcast—the program you used if you were an out-of-touch old-timer who didn't care about security.

I for one think telnet isn't all bad. Sure, it can't keep a secret, but it still can do a lot of useful things around the server room. Really, telnet just provides you a convenient way to connect to a network port and send commands. Telnet can work well to diagnose problems

with one of the many services out there that still accept plain-text commands in their protocol. In fact, it's one of my go-to command-line programs when I'm troubleshooting. In this column, I'm going to give telnet a second chance and describe how to use it to perform some common troubleshooting tasks.

Test Remote Ports

There are many different ways to test whether a network port is listening on a system, including GUI port scanners, Nmap and nc. Although all of those can work well, and even I find myself using Nmap more often than not, not all machines end up having Nmap installed. Just about every system includes telnet though, including a lot of embedded systems with BusyBox environments. So if I wanted to test whether the SMTP port (port 25) was listening on a server with

While you are connecting to port 80, you might as well actually throw some HTTP commands at it and test that it works.

the IP 192.168.5.5, I could type:

```
$ telnet 192.168.5.5 25  
Trying 192.168.5.5...  
telnet: Unable to connect to remote host: Connection refused
```

In this case, the remote port is unavailable, so I would fall back to some other troubleshooting methods to figure out why. If the port were open and available though, I could just start typing SMTP commands (more on that later).

As you can see from the above example, the syntax is to type the command `telnet`, the IP or hostname to connect to, and the remote port (otherwise it will default to port 23—the default port for telnet). So if I wanted to test a Web server instead, I would connect to the HTTP port (port 80):

```
$ telnet www.example.net 80
```

Troubleshoot Web Servers

While you are connecting to port 80, you might as well actually throw some HTTP commands at it and test that it works. For starters, you want to make sure you actually are connected:

```
$ telnet www.example.net 80  
Trying 192.168.5.5...  
Connected to www.example.net.  
Escape character is '^]'.  
[REDACTED]
```

Once you are connected, you can pass a basic HTTP GET request to ask for the default index page followed by the host you want to connect to:

```
GET / HTTP/1.1  
host: www.example.net
```

The GET request specifies which page (/) along with what protocol you will use (HTTP/1.1). Since these days most Web servers end up hosting multiple virtual hosts from the same port, you can use the host command so the Web server knows which virtual host to direct you to. If you wanted to load some other Web page, you could replace `GET /` with, say, `GET /forum/`. It's possible your connection will time out if you don't type it in fast enough—if that happens, you always can copy and paste the command instead. After you type your commands, press Enter one final time, and you'll get a lot of headers you don't normally see along

with the actual HTML content:

```
HTTP/1.1 200 OK
```

```
Date: Tue, 10 Jul 2012 04:54:04 GMT
Server: Apache/2.2.14 (Ubuntu)
Last-Modified: Mon, 24 May 2010 21:33:10 GMT
ETag: "38111c-b1-4875dc9938880"
Accept-Ranges: bytes
Content-Length: 177
Vary: Accept-Encoding
Content-Type: text/html
X-Pad: avoid browser bug
```

```
<html><body><h1>It works!</h1>
<p>This is the default web page for this server.</p>
<p>The web server software is running but no content
has been added, yet.</p>
</body></html>
```

As you can see from my output, this is just the default Apache Web server page, but in this case, the HTML output is only one part of the equation. Equally useful in this output are all of the headers you get back from the **HTTP/1.1 200 OK** reply code to the modification dates on the Web page, to the Apache server version. After you are done sending commands, just press **Ctrl-]** and **Enter** to get back to a telnet prompt, then type **quit** to exit telnet.

I usually just use telnet to do some basic HTTP troubleshooting, because once you get into the realm of authentication, following redirects and other more-complicated parts of the protocol, it's

much simpler to use a command-line tool like curl, or I guess if you have to, even a regular GUI Web browser.

Send an E-mail

Although I just use telnet for basic Web server troubleshooting, telnet ends up being my preferred tool for e-mail troubleshooting, mostly because it's so simple to send a complete e-mail with only a few telnet commands.

The first step is to initiate a telnet connection with the mail server you want to test on port 25:

```
$ telnet mail.example.net 25
Trying 192.168.5.5...
Connected to mail.example.net.
Escape character is '^].
220 mail.example.net ESMTP Postfix
```

Unlike the blank prompt you may get when you connect to an HTTP server, with SMTP, you should get an immediate reply back. In this case, the reply is telling me I'm connecting to a Postfix server. Once I get that 220 prompt, I can start typing SMTP commands, starting with the **HELO** command that lets me tell the mail server what server is connecting to it:

```
HELO lappy486.example.net
250 mail.example.net
```

The nice thing about the interactive SMTP

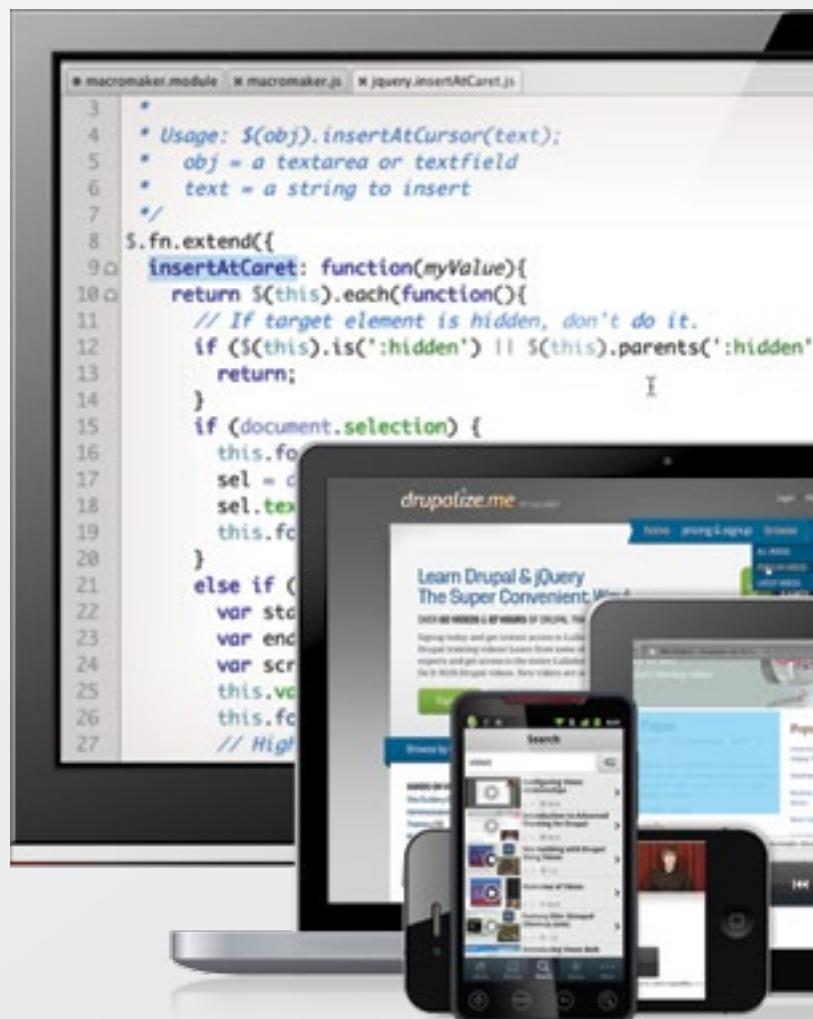


The Most Convenient Way to Learn Drupal!

Have hundreds of hours of Drupal training right at your fingertips with the Drupalize.Me app. Learn while you're on the bus, in line at the bank, on the couch, or anywhere! New videos are being added every week to help you stay up to date on the latest Drupal knowledge.

Learn about our latest video releases and offers first by following us on Facebook and Twitter (@drupalizeme)!

Go to <http://drupalize.me> and get Drupalized today!



connection here is that if I do somehow make a typo in a command or make a mistake, it should let me know; otherwise, I should get a 250 reply. After HELO, you use the MAIL FROM: command to list what e-mail address the e-mail should appear to be from. I say “appear to be from”, because you can put just about any e-mail address you want here, which is a good reason not to blindly trust FROM addresses:

```
MAIL FROM: <root@example.net>
250 Ok
```

In the past, I used to type in the e-mail address directly without surrounding it with <>. My personal Postfix servers are fine with this, but other mail servers are more strict and will reply with a syntax error if you don't surround the e-mail address with <>. Since this FROM address was accepted, you can follow up with RCPT TO: and specify who the e-mail is addressed to:

```
RCPT TO: <postmaster@example.net>
250 Ok
```

The fact that the mail server responded with 250 should mean that it accepted the TO address you specified here. Finally, you can type DATA and type the rest of your e-mail, including any extra headers you want to add, like Subject, then finish up with a single period on its own line:

```
DATA
```

```
354 End data with <CR><LF>.<CR><LF>
```

```
Subject: Give Telnet a Chance 1
```

```
Hi,
```

```
All we are saying is give telnet a chance.
```

```
.
```

```
250 Ok: queued as 52A1EE3D117
```

When I'm testing e-mails with telnet, I usually put a number in the subject line so I can continually increment it with each test. This way, if some e-mail messages don't get delivered, I can tell which ones went through and which ones didn't.

Once you are done with the DATA section and the e-mail is queued, you can type quit to exit:

```
quit
```

```
221 Bye
```

```
Connection closed by foreign host.
```

Now that you have some ways to troubleshoot with telnet, hopefully you won't relegate telnet to the junk drawer of your Linux systems. Sure, you may not want to use it for remote shells, but now that just about everyone uses SSH anyway, maybe you can break out telnet on your terminal for all of your other plain-text network needs without your friends scolding you. ■

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.



9th Annual High Performance Computing For Wall Street

September 19, 2012 (Weds) Roosevelt Hotel, NYC

Madison Ave and 45th St, next to Grand Central Station

High Performance Computing, Cloud, Big Data, Low Latency, Network Systems, Data Centers, Optimization, Linux, Speed and Cost Savings. All-in-One.

Join the new speakers announced and their programs covering HPC, Big Data, Cloud, Data Centers, Networks, Switches, Optimization, Linux, Speed and Cost Savings. All-in-one in New York City.

Wall Street and the global financial markets will be coming to the Roosevelt for the largest meeting of HPC in New York.

Come to this convenient one-day meeting place. Network with your friends. See and hear the latest systems to reduce cost, increase speed and productivity.

Transformation is what Wall Street IT is all about. Wall Street is constantly moving towards zero latency, zero net gain, and faster moving parts with less staff.

Save \$100. Advance registration \$295. \$395 onsite. Conference includes general sessions, drill down sessions, the exhibits, handouts, an industry luncheon, special sponsor functions.

Don't have time for the Conference? Register for the free Show from 8 am-4 pm, but you must go online.

Exhibit and Sponsor information from Flagg Management:

www.flaggmgmt.com/hpc

Platinum Sponsor



Gold Sponsors



Show & Conference:
Flagg Management Inc
353 Lexington Avenue
New York 10016
(212) 286 0333
fax: (212) 286 0086
flaggmgmt@msn.com

Show Hrs: Wed, Sep 19 8:00-4:00
Conference Hrs: 8:30-4:50

Cisco
Keynote
Speaker 9am



Paul Perez
Chief Technology Officer
Data Center Group
Cisco



Jeffrey M. Birnbaum
Founder, CEO
60East Technologies



Antonio Hallak, Head
GLL Electronic Trading
Engineering, Credit
Suisse Securities (USA)

Partial list of speakers 7/27/12



Bob Gaines
Senior Sales Director
DataDirect Networks



Dino Vitale (invited)
Director, Morgan Stanley
Cross Technology Svcs.



Dave Weber
IBM Global Prog Exec,
Wall St Ctr for Excellence



Jan Mark Holzer
Sr Consulting Engineer
Red Hat



Tom Steinert-Threlkeld
Editor-in-Chief, Securities
Technology Monitor



Joseph Curley
Dir. Technical
Computing Mktg, Intel



Nan Boden
CEO
Myricom



Nick Werstiuk, IBM Corp.
Product Mgmt
Platform Computing



Raymond Russell
CTO, Co-Founder
Corvil



Thomas Isakovich
CEO
Nimbus Data



Geoffrey Noer
Sr. Director Product Mktg
Panasas



Pat Greene
Solution Architect
Hewlett Packard



Nikita Ivanov
CEO
GridGain Systems



Vikram Mehta
VP System Networking
IBM Corporation



Emile Werr, VP & Head
of Enterprise Architecture
NYSE Euronext



Nick Werstiuk
Product Mgmt
Platform Computing



Anshul Sadana
SVP of Customer
Engineering
Arista Network



Jason Stowe
CEO
Cycle Computing



Alex Tabb, Partner
Crisis & Continuity Svcs
Tabb Group

www.flaggmgmt.com/hpc



SHAWN POWERS

A Domain by Any Other Name...

Don't let DNS get you into a BIND; read on to sort out port 53.

In this article, I cover DNS, arguably the most "Rube Goldberg" of all services. (Well, except for Sendmail, but that's really just one application, not an entire service.) DNS (Domain Name Services) quite simply maps domain names to IP addresses. For some reason, it's easier for humans to remember words than strings of numbers, so rather than remembering 12.34.56.78, we remember www.linuxjournal.com.

Using DNS instead of remembering IP addresses not only helps prove Linux users aren't really cyborgs, it also allows some pretty cool magic in the server department. Instead of one server per Web site, a single server with a single IP address can host multiple Web sites. Unfortunately, the way DNS works on a global scale means it's not without its faults and frustrations. G.I. Joe always said, "knowing is half the battle", so in this article, I walk you through being a knowledgeable DNS user, without ever delving into the

complexities of the underlying system.

The Un-DNS

For most Linux distributions, you configure "how" the computer looks up URLs by setting the options in /etc/nsswitch.conf. If you look at your nsswitch.conf file, you'll probably find the line:

`hosts: files dns`

This line tells the computer that before it asks its DNS server for the IP address, it should look into its /etc/hosts file for a domain mapping. By default, there probably is a line defining the localhost address and possibly an entry defining whatever hostname and IP you set for the computer. This file has the following format:

# IP Address	Domain Name
127.0.0.1	localhost
192.168.1.1	router
192.168.1.10	homeserver
192.168.1.20	xbmc

Using DNS instead of remembering IP addresses not only helps prove Linux users aren't really cyborgs, it also allows some pretty cool magic in the server department.

Once entered into the file, you can use the names defined in place of an IP address. This is truly domain name resolution in its simplest form. If you need only to map an address for your local computer, this is the ideal way to configure your computer. Keep in mind that the order specified in `nsswitch.conf` is the order your computer will search, so if you put an entry like this:

192.168.1.20 www.google.com

it won't ever look up the proper address for Google. It will query the server at 192.168.1.20 as if it were www.google.com. This is a feature rather than a bug. Although it certainly allows for some easy pranking (you didn't hear that from me), it also can be used to block specific sites. If you wanted to block Facebook on your home computer, for example, you could add:

127.0.0.1 www.facebook.com

Then, when the user tries to access Facebook, it will fail. Note that this is not a foolproof way to block Internet sites, but it works in most situations. This method often is used to block ads on Web pages. It's possible to find a list of ad servers and then put the list into your `/etc/hosts` file. Again, it's not foolproof, but the logic is sound. The hosts file is also useful for other purposes too, which I'll come back to later.

DNSMasq—the Super Simple Server

If you're using an off-the-shelf router for your home network, chances are it's running DNSMasq as a DNS server. DNSMasq is a DNS forwarder that queries a remote DNS server and returns the value to the client requesting the information. It has the handy feature of querying the router's `/etc/hosts` file first, and most routers have a way to add DNS entries. It's not always simple, like the case of DD-WRT. In order to add an address to the DNSMasq server on DD-WRT, you need to add lines to the DNSMasq

config section like this:

```
address=/homeserver/192.168.1.10  
address=/xbmc/192.168.1.20
```

See Figure 1 for a DD-WRT screenshot. By adding addresses to your router's DNSMasq server, you effectively can make an /etc/hosts file for every computer on your network. DNSMasq also is tied into the DHCP server, allowing for automatic mapping of hostnames to DHCP assignments, but that's a bit outside the scope of this column.

Getting into a BIND

When it comes to DNS on the greater Internet, BIND is the de facto standard server. Unfortunately, BIND also is where the vast complexities of the Domain Naming System come into play. Don't get me wrong, the complexities aren't frivolous, just frustrating at times. If you are managing a DNS server

for a business, chances are you need to work with BIND. Because BIND supports every facet of the DNS concept, I think it's important for a little terminology lesson before we dig into configuration. See the sidebar for some DNS terms you should be familiar with before delving into BIND configuration.

First off, if you're going to configure BIND, I recommend using a tool like Webmin for your first time. There are some quirks when you edit the BIND configuration file that aren't apparent at first. For example, when you edit a zone file (usually stored in /var/named), you need to increment the serial number at the top, or other servers won't see your information as the most recent.

Assuming your zone files are created, it's fairly easy to see how to add or modify records. The only thing I'll specifically mention here about the zone file is the TTL setting. This setting tells clients how many seconds

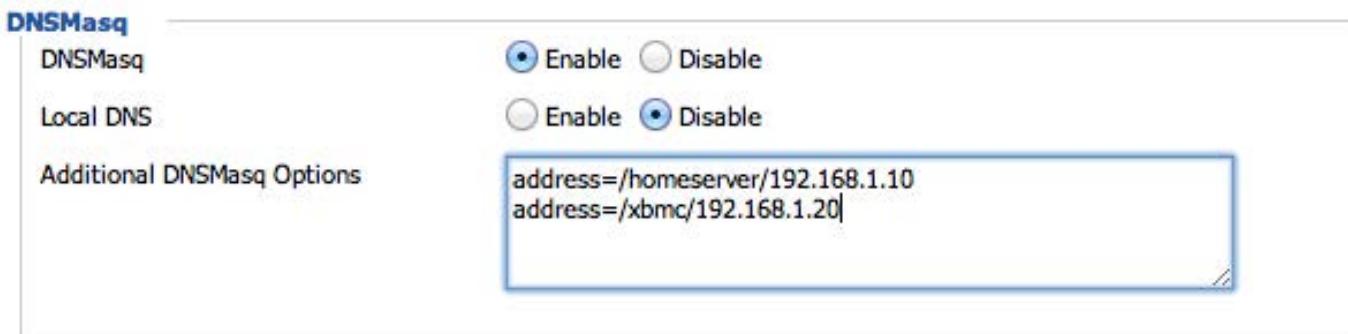


Figure 1. DD-WRT allows you to add DNS entries, but it's not terribly user-friendly.

Common DNS Terms

- **A Record (Address Record):** this type of record directly maps a name to an IP address. Originally, no two A Records were supposed to point to the same IP address. (This is no longer practical, but is considered “best practice” where appropriate.)
- **Authoritative:** a server is considered authoritative when it is hosting the domain in question itself rather than querying another server for the information. A server is considered authoritative by domain; it’s not a boolean server setting like with DHCP. The same server can be authoritative for one domain, and not for another.
- **BIND (Berkeley Internet Name Domain):** the most common DNS server on the Internet.
- **Caching (or Namecaching):** locally stored copy of name resolution from an authoritative DNS server. The caching duration is based on the TTL settings from the authoritative server (see TTL below).
- **CNAME (Canonical Name Record):** this creates an alias to another DNS entry that inherits the properties of the original.
- **Forward Zone:** a “zone” is used to define the section of DNS space where a server is responsible for mapping names to IP addresses.
- **Reverse Zone:** a DNS server also can supply reverse lookups, mapping names to queried IP addresses. This often is used for security to verify DNS information.
- **FQDN (Fully Qualified Domain Name):** this is the entire DNS name, including a period at the end.
- **MX Record (Mail Exchange Record):** this specifies a mail route for a particular domain. Multiple MX Records are possible (and recommended!) with priority levels.
- **NS Record:** declares what server serves a given zone. This is where the server would declare itself authoritative for a particular zone.
- **PTR Record (Pointer Record):** a PTR record often is called the reverse record, and it associates an IP address with a domain name.
- **Propagation:** the period of time between when a DNS change is made on the authoritative server and the time all servers on the Internet have current information. This propagation time can be several hours or several days depending on the TTL settings for a particular record.
- **Root Server:** there currently are 13 root servers on the Internet, which host the top-level domains. Through very complex routing and redundancy, these servers are all over the globe and are placed with fault tolerance in mind.
- **SOA Record (Start of Authority Record):** the first record in a zone file, containing information about the zone itself, including whether or not the server is authoritative.
- **SRV Record (Service Record):** provides information about what services are available for a domain.
- **Top-Level Domain:** any zone hosted by the 13 root servers. These are domains like com, edu, org, gov and country codes like us, ca and uk.
- **TTL (Time To Live):** this is a number set by the authoritative server for a domain that tells DNS servers how long to cache information before querying again.

the domain information is “good” for. If you plan to change Web hosts, it’s good practice to set this TTL setting to something low about a week before the change, so when you do make the change, propagation time across the Internet goes much more quickly. Some DNS hosting companies set this to DAYS by default, so your well-planned host migration could take a week to propagate instead of a couple hours.

Tools of the Trade

Although messing with the DNS servers is how you adjust settings, testing them isn’t as straightforward as you might think. If you change a setting on one server, it takes a while to propagate to other servers, so testing can be frustrating. Thankfully, it’s not hard to query specific servers.

For years, nslookup was the tool for doing DNS lookups. For some reason, several years ago, nslookup became deprecated, and no one bothered to tell me about it. I’m telling you now, so you don’t look foolish trying to use nslookup on your machine. The new dog in town is the dig command, and it’s pretty cool.

Issuing the dig command on its own will query whatever DNS server is assigned to your system. So typing:

```
dig www.linuxjournal.com
```

will yield something like this:

```
spowers@server:~$ dig www.linuxjournal.com
```

```
; <>> DiG 9.8.1-P1 <>> www.linuxjournal.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 50038
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.linuxjournal.com.           IN      A

;; ANSWER SECTION:
www.linuxjournal.com.      388     IN      A      76.74.252.198

;; Query time: 34 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Wed Aug  1 10:32:23 2012
;; MSG SIZE  rcvd: 54
```

You’ll notice at the bottom that the server responding is my local router at 192.168.1.1. If I want to query a different DNS server, however, it’s as easy as typing:

```
dig @8.8.8.8 www.linuxjournal.com
```

which gives this:

```
spowers@server:~$ dig @8.8.8.8 www.linuxjournal.com
```

```
; <>> DiG 9.8.1-P1 <>> @8.8.8.8 www.linuxjournal.com
; (1 server found)
```

```

;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27150
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.linuxjournal.com.          IN      A

;; ANSWER SECTION:
www.linuxjournal.com. 270     IN      A      76.74.252.198

;; Query time: 30 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Wed Aug  1 10:34:21 2012
;; MSG SIZE  rcvd: 54

```

Notice that the response is the same, but at the bottom you can see the server I queried was 8.8.8.8 (Google's public DNS server, which is terribly easy to remember).

One important thing to note is that the dig command will *not* honor entries in your /etc/hosts file. It's strictly a DNS lookup tool, so it only knows to query servers.

Some Cool DNS Tricks

Now that you have at least a loose understanding of how DNS works, I want to share a few of the really nifty things you can do with it in your network.

Rudimentary Load Balancing

Simply by adding multiple A Records for

a single domain, it's possible to create a load-balancing situation for your servers. The BIND server will respond to queries in a round-robin fashion when there are multiple A Records. If you have a service you'd like to split across servers without configuring actual load balancing, round-robin DNS is a viable solution.

It's important to keep in mind that this solution has several failings. It doesn't actually load balance, it just alternates DNS responses. So, luck of the draw might mean one server is far busier than another. Round-robin DNS also breaks reverse lookups. If your application requires reverse DNS to function, a round-robin scenario will not work. Also, with a low TTL, constantly changing IP information likely will have an adverse effect on most services.

Split DNS

I touched on this concept earlier with regard to putting false entries in the /etc/hosts file. It's sometimes called "fake DNS" or "DNS hijacking" or any of several nefarious-sounding titles. Basically, the concept is that you host a zone locally and declare yourself authoritative for your clients. Because you've told your BIND server it was authoritative for a zone, it will serve only the IP information you configure and not query the greater Internet for actual IP mapping. This is very useful if you want

to map internal domain names to private IPs. This is also useful if you want certain domains (like `intranet.example.com`) to exist only inside your company and not even resolve from the Internet.

It's important to note that this is a hack of the protocol and can have its failings too. In fact, some routers will detect split DNS as a DNS-spoofing attack (which technically it is) and not allow you to use those false addresses.

Virtual Domains

This isn't a hack at all, but rather a feature of most Web servers. Because the number of Web sites far surpasses the number of available IP addresses, Web servers now allow separate Web sites using the same IP address. For example, let's say three friends share a server. The DNS entries for their blogs might be:

12.34.56.78	<code>shawnblog.example.com</code>
12.34.56.78	<code>julieblog.example.com</code>
12.34.56.78	<code>frankblog.example.com</code>

You'll notice the three different domains share the same IP address. If you try to access `http://12.34.56.78`, the Web server won't know whose blog you're trying to access. By configuring virtual domains, however, the Web server running at 12.34.56.78 can differentiate between the blogs by which address you are querying. So `http://shawnblog.example.com` will get one Web site, and

`http://julieblog.example.com` will get an entirely different one.

This ability for DNS entries to share a common IP address and Web servers to serve pages based on which domain name was requested has allowed for the modern Internet to work. Without that feature, shared hosting wouldn't be possible.

Virtual Hosting for One

While Internet Web hosting relies on virtual hosts for most sites, a Web developer or sysadmin often can become frustrated waiting for propagation time when a DNS change is made. Let's say you've created a fancy new Web site on a new Web server, and you want to make sure it's working once the DNS change propagates through the Internet. Since Web servers reply based only on the domain request they receive, until DNS propagates, it's impossible to make sure the new Web site is working.

Thankfully, the Web server itself doesn't do DNS lookups; it knows only the domain names it's supposed to respond for. If you change your local `/etc/hosts` file with the new DNS information *before* you make the change on the Internet, you can test your new server using your local DNS information. In fact, without the ability to make this local bogus DNS change, testing your new server would be close to impossible!

At the End of the Day

DNS is a complex system of root servers, zone transfers, propagation time and reverse lookups. As you can see, however, it does a lot more than just map names to IP addresses. Without DNS, the Internet would be a lot less useful, and company Web sites would be far more difficult to remember. ("Come check us out at <http://224.143.77.155> !!" doesn't really roll off the tongue very well.)

By using DNS on your local network, you can save a lot of time and make future changes far less painful. If BIND overwhelms you, don't let that bother

you too much. You certainly don't have to be an expert on zone transfers in order to grasp DNS. The easiest way to start is to play with your /etc/hosts file. Playing with someone else's /etc/hosts file also can be fun, but be sure to use your super powers only for good! ■

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.

Powerful: Rhino



Rhino M4700/M6700

- Dell Precision M4700/M6700 w/ Core i7 Quad (8 core)
- 15.6"-17.3" FHD LED w/ X@1920x1080
- NVidia Quadro K5000M
- 750 GB - 1 TB hard drive
- Up to 32 GB RAM (1866 MHz)
- DVD±RW or Blu-ray
- 802.11a/b/g/n
- Starts at \$1190
- E6230, E6330, E6430, E6530 also available

- High performance NVidia 3-D on an FHD RGB/LED
- High performance Core i7 Quad CPUs, 32 GB RAM
- Ultimate configurability — choose your laptop's features
- One year Linux tech support — phone and email
- Three year manufacturer's on-site warranty
- Choice of pre-installed Linux distribution:



Tablet: Raven



Raven X230/X230 Tablet

- ThinkPad X230/X230 tablet by Lenovo
- 12.1" HD LED w/ X@1366x768
- 2.6-2.9 GHz Core i7
- Up to 16 GB RAM
- 750 GB hard drive / 180 GB SSD
- Pen/finger input to screen, rotation
- Starts at \$2050
- T430, T530, W530 also available

Rugged: Tarantula



Tarantula CF-31

- Panasonic Toughbook CF-31
- Fully rugged MIL-SPEC-810G tested: drops, dust, moisture & more
- 13.1" XGA TouchScreen
- 2.4-2.53 GHz Core i5
- Up to 8 GB RAM
- 320-750 GB hard drive / 512 GB SSD
- CF-19, CF-52, CF-H2 also available

EmperorLinux
...where Linux & laptops converge

www.EmperorLinux.com
1-888-651-6686

Model specifications and availability may vary.

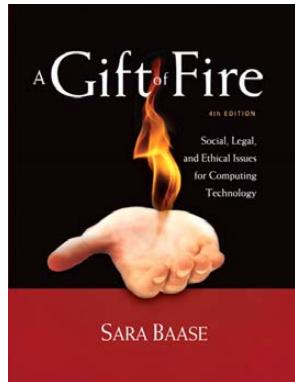




The Linux Professional Institute's Linux Essentials Program

With youth unemployment around the world holding at crisis levels, the new Linux Essentials Program from the Linux Professional Institute (LPI) couldn't come at a more auspicious time. The new certification program, consisting of an exam and Certificate of Achievement, introduces new users and youth to Linux and open-source software. The program's main topics include the Linux community, careers in open source, popular operating systems, important applications, licensing issues and the basics of the command line, files and scripts. Other program elements include regional links to employment and apprenticeship programs; support for skills competitions, such as Worldskills International; and support for teacher collaboration and sharing of learning exercises. A "low-stakes" exam is available on-line either through the LPI or Internet-based testing via partners. Linux Essentials currently is available at select LPI affiliate locations and IT events in Europe, the Middle East and Africa.

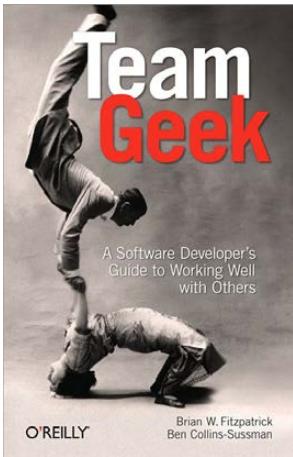
<http://www.lpi.org/linuxessentials>



Sara Baase's *A Gift of Fire*, 4th ed. (Prentice Hall)

During its relatively brief history, our dynamic Linux community has had an outsized interest in and influence on the public dialogue concerning the role of technology in society. An outlet for us Linuxers to scratch our inner philosophical itch is to chew on the ideas found in the new 4th edition of Sara Baase's book *A Gift of Fire: Social, Legal, and Ethical Issues for Computing Technology*. As indicated by the title, Baase's work explores the social, legal, philosophical, ethical, political, constitutional and economic implications of computing and the controversies they raise. With a computer scientist's perspective, and with historical context for many issues, Baase covers the issues we face both as members of a technological society and as professionals in computer-related fields. Baase's primary goal is to encourage current and future computer professionals to understand the implications of what they create and how it fits into society at large.

<http://www.informit.com>



Brian W. Fitzpatrick and Ben Collins-Sussman's *Team Geek* (O'Reilly Media)

People are rational and predictable. [Editor's note: Let's try that again.] Sometimes people can be irrational and unpredictable. As a trained software engineer, you're probably a whiz with computer languages, compilers, debuggers and algorithms. But how much real training did you get in dealing with the human side of software development? If your answer is a predictable "not much", crack open Brian W. Fitzpatrick and Ben Collins-Sussman's new book *Team Geek: A Software Developer's Guide to Working Well with Others*. Authors Fitzpatrick and Collins-Sussman, producers of the popular video series, "Working with Poisonous People", cover the basic patterns and anti-patterns for working with other people, teams and users while trying to develop software. Readers learn how to deal with imperfect people and discover why playing well with others is at least as important as having great technical skills. By internalizing the techniques in this book, publisher O'Reilly says that readers will get more software written, be more influential and happier in their careers.

<http://www.oreilly.com>

TalentSoft

Enterprises frequently experience a gulf between managerial needs and the ability of the human resources department to deliver appropriate talent to address those needs. "Problem solved", says TalentSoft, thanks to the new TS Spring'12 release of its SaaS-based Integrated Talent and Competencies Management application, which now incorporates the best social networks practices into its collaborative talent management solution. Employees gain control over their careers; managers benefit from greater autonomy, and HR managers increase their productivity by focusing on high value-added tasks. TS Spring'12 adds two new core features: TS SocialConnect and TS TalentOffice. TS SocialConnect is a solution that helps employers leverage more than 300 social networks toward strengthening brands and attracting the best profiles. TS TalentOffice allows HR departments to draw upon their existing resources to generate customized spreadsheet-based reports and word-processing documents.

<http://www.talentsoft.com>

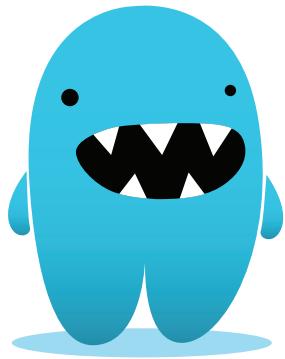


Logic Supply's LGX CT100 Open Chassis

Logic Supply designed the new LGX CT100 Open Chassis to solve the space and installation challenges involved with securing Mini-ITX (or 3.5") mainboards and other system components within larger enclosures, such as cabinets and kiosks. The CT100 is a "case option" designed to be an open mounting plate solution fitting into an existing electrical cabinet, kiosk or piece of manufacturing equipment. The result, says Logic Supply, is a compact, enclosure-free home for all the IT equipment with the benefit of convenient accessibility to those who need to maintain the IT over time. Absence of physical enclosure means that many different system configurations are available and the full range of Mini-ITX and 3.5" motherboards and power-supply options are supported. Logic Supply also designed the CT100 with bench-top configuration testing in mind: it holds all parts safely during system debugging and prototyping. A range of mounting provisions are available, with only a DIN rail or screw studs required for attachment of the plate to the enclosure.



<http://www.logicsupply.com>



Belongs Inc.'s Belon.gs Global Lost & Found Service

Losing stuff has long been accepted as a frustrating yet quintessential part of the universal human condition. A team of resourceful Finns—and who loves resourceful Finns more than we do?—is launching the Belon.gs Global Lost & Found Web and mobile service to help unfortunate folks worldwide recover lost important belongings.

To utilize the service, customers register at Belon.gs and attach QR code tag stickers to their valuables. If an item should go missing, the finder can scan the tag's QR code with a smartphone or access the Web address on the tag. The owner is automatically notified and anonymous chat is established between the two parties to arrange the return of the lost item. To promote the returning of valuables, Belon.gs supports setting rewards for found items through PayPal, and the Belongs technology will streamline the transfer of the reward from owner to finder. The aforementioned Finns envision a better world with Belong.gs—a world without the unnecessary grief that comes from losing your valuables.

<http://www.belong.gs>



Cryptzone's AppGate MOVE

James Bond would effortlessly foil many a cyber threat if Q would slip him an AppGate MOVE (My Own Virtual Environment) from Cryptzone, a USB Flash drive that provides a portable way to access information and applications securely from virtually any computer. By working independently of the host device's operating system, the bootable AppGate MOVE allows secure remote working and eliminates risk from malware infection. Cryptzone says that the trend for increased telecommuting calls for a low-cost solution that allows for trusted access to corporate information from an untrusted computer at home or in a public space. Working in combination with an AppGate Security Server, this drive contains a full operating system, the AppGate client, a Web browser, a Microsoft-compatible Office Suite, e-mail client, a firewall and other applications required to complete daily tasks. There is no need for an expensive, dedicated corporate laptop. AppGate MOVE allows users to work securely because the configuration of the PC is irrelevant and untouched. Because the local PC hard drive is not used, no residue or evidence is left when the session to the AppGate server is closed. The AppGateUSB Factory allows production of unlimited copies of AppGate MOVE.

<http://www.cryptzone.com>

Crossrider

Multiple Web browsers are convenient for users but can be a nightmare for developers, who historically have had to create extensions according to each browser's unique requirements. A new company, Crossrider, with its eponymous, cloud-based cross-browser extension development platform, seeks to make life more pleasant for extension developers. Crossrider lets developers create cross-browser extensions quickly with a single JavaScript code and publish to potentially millions of end users immediately. All major browsers are supported, including Chrome, Firefox, Internet Explorer and Safari. The platform also provides the tools needed to build and manage apps, including JQuery support, a powerful API and app-boosting Crossrider plugins, which allow developers to create custom code to extend the Crossrider API and enable code sharing within the Crossrider developer community. Crossrider also features a cloud-based IDE that allows developers to create extensions in real time without downloading a development package. The IDE is equipped with IntelliSense to enable autocompletion for all Crossrider API methods. Tools for monetizing and tracking extensions also are included in the platform.

<http://crossrider.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

ZaTab: ZaReason's Open Tablet

Open, rooted, warranty included—we take ZaReason's new open tablet, the ZaTab, for a spin. KEVIN BUSH

Quite a few options exist as far as Android tablets go. Some of them are great choices for personal entertainment and media consumption. Google's new Nexus 7 is a powerful little beast designed to serve up media from Google Play. Amazon's Kindle Fire is a great device for tapping Amazon's extensive content offerings. Undoubtedly, these tablets were designed to direct more of your money to the tablet-maker's on-line content marketplaces. The glaring lack of SD card expansion on these devices confirms this. The ZaReason team designed a tablet that can be what the user wants it to be—one that supports users' own content, that is not necessarily tied to a particular content store and that can be used as far more than a simple consumption device. Have they

succeeded in creating the world's first open tablet? Let's find out.

STATS:

- CyanogenMod 9 Android 4.04.
- Allwinner A10 SoC.
- 9.7" IPS 1024x768 display.
- Five-point capacitive touchscreen.
- 16GB internal storage + microSD for additional storage.
- 1GB of RAM.
- Wi-Fi (802.11 b/g/n).
- Front-facing and back cameras.
- Sturdy metal back.
- High-capacity 8000 mAh battery.
- Ultra-light (630 grams).

PORTS:

- Headphone.
- MicroSD card slot.
- Micro-HDMI video out.
- 2x micro-USB ports.



The ZaTab's Generic OEM Retail Box

As you read over the stats, a few things should catch your attention. First, the ZaTab comes loaded with CyanogenMod 9, based on Android 4.04 Ice Cream Sandwich, and is rooted out of the box. I feel like I need to say that again—it is rooted out of the box! You can open a terminal and explore your device with root permissions in a matter of seconds. There's no need to run an exploit to get root access, no need to flash another ROM, no voided warranties. Second, there is 16GB of

internal Flash storage for apps and media, and a microSD slot that can accommodate an additional 32GB of storage. So, there's ample storage for your content, stored locally, on your device—no need to rely on “the cloud” or streaming media services.

Opening the Box

The ZaTab arrived packed well inside a corrugated shipping box, with bubble wrap surrounding the retail box. The retail box is generic OEM fare. I would



The Included Accessories: Micro-USB-to-USB Cable, Micro-USB-to-Female USB Adapter, A/C Adapter and Generic User's Guide

love to see ZaReason produce a branded box for the tablet, but that is another expense and surely would raise the price. Snuggled in the box under the tablet, you will find a micro-USB-to-USB cable, a micro-USB-to-female-USB adapter for connecting USB-based accessories, an A/C adapter and a small generic Android manual. The tablet feels sturdy but not too heavy—definitely not cheap. Unlike most Android tablets out there, the casing is not plastic. The back/sides consist of a solid piece of matte aluminum, with a large ZaReason logo

silk-screened on the back of the unit. The port and button labels also are silk-screened on the back just below their respective buttons/ports. Along the top edge, you will find the power button, 5v DC power input, micro-HDMI out, two micro-USB ports, an 1/8" headphone jack and three small vents to keep things cool. Along the right edge, you will find a back button, which takes you to the previous screen in the Android interface, and the volume up/down buttons. On the back of the ZaTab is a paper-clip-style recessed physical reset button, the main camera



The Clean Aluminum Casing

and a small grill protecting the speaker. The front of the ZaTab is mostly screen with a 1/2" black bezel, with a front-facing camera in the top-right corner.

Android Setup

If you have used any Android device in the past, the initial setup on first boot will be familiar. Input your Google credentials, connect to a network, and you're rolling. You have the option to download and install previous Android apps you have used on any other synced devices you may have, and your

bookmarks as well if you are a Google Chrome user. The device ships with a minimal set of Android apps: Apollo Media Player, Android Web Browser, Calculator, Calendar, Email, Camera, Gallery, Clock, DSPManager, Movie Studio, People (Address Book), ROM Manager, Superuser, Terminal Emulator and the Android Settings app. You won't find any preloaded crapware, no nearly useless game demos and no irremovable commercial apps. The Google Play Store installs upon syncing your Google account, so you have access to the



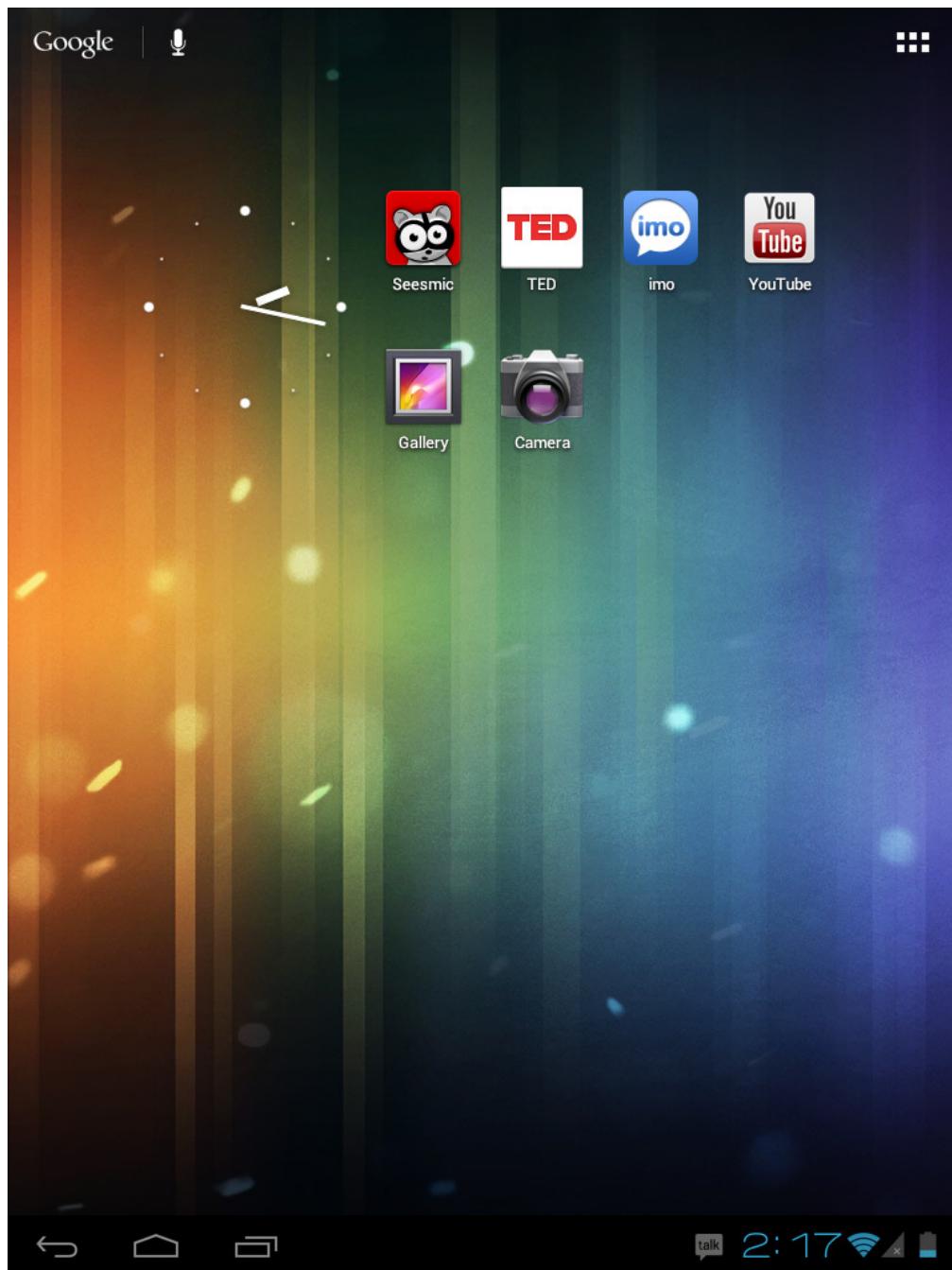
The ZaTab—Ready to Come Out and Play

largest selection of Android apps from the start.

The Apps

Most of the apps I installed work flawlessly. I've had a blast streaming TED

Talks on the ZaTab with TED's official Android app. The video is high-quality and plays flawlessly on the ZaTab. Google's Gmail app is perfect for the tablet with split views for folders and messages. The generic e-mail app works in much the



The Android ICS Home Screen on the ZaTab

same way with support for Exchange, IMAP and POP. The Plume Twitter client is a pleasure to use on the big screen. Amazon's Kindle app looks great as well with easily configurable font sizes and text colors from which to choose. *Linux*

ZaTabs as you read this.

There were a few apps that just would not play nice with the ZaTab. Netflix, for example: the app's interface worked fine, but the app would stall when trying to stream video. Twitter's homegrown client

Journal's own app looks good on the ZaTab, with text-mode rendering sharp text. I was able to connect to my employer's Cisco VPN using Cisco's Anyconnect for rooted Android devices. Earl, from ZaReason, was kind enough to provide a tun.ko tun module for the ZaTab when asked in the #zareason IRC chat room on Freenode. This was necessary for the Anyconnect client as it uses a tun kernel module to facilitate the VPN connection. Earl tells me that this module will be preloaded on the ZaTab upon official release, and it may be shipping on

was not available in the store. It must look for certain “approved” device profiles and the ZaTab may not be one of them.

The Hardware in Use

The 9.7" 1024x768 in-plane-switching capacitive touchscreen is bright with brilliant color and has an insane viewing angle. You can tilt this thing nearly 90 degrees in any direction and maintain view-ability. I find the screen size ideal for a personal touchscreen device. Text is sharp, of reasonable size, and movies are a joy to watch.

Both cameras are unimpressive. Photos taken with the main, rear-facing camera are grainy and quite dark indoors because there is no flash. The front-facing camera is adequate for low-resolution video chats, but it is also quite grainy.

Battery life on the other hand is fantastic. You can use the ZaTab heavily all day long without worrying about power. For example, the day after the ZaTab arrived, after a full charge, I spent lots of time downloading and installing apps, watching TED videos, listening to streaming music via Google Music, reading via the Kindle app and exploring the unit via the terminal emulator. After 15 hours of mostly continuous use, I had 40% charge remaining.

The Interface

The ZaTab comfortably runs Android

ICS. The animated UI transitions and elements are smooth, and there is plenty of processing power for most apps despite the tablet being a single-core unit. Switching apps using the Recent Applications menu makes multitasking simple. Notifications are unobtrusive, and apps that are notifying can be opened directly from the notification widget. Plenty of informative widgets are available if you like your home screen to be more dashboard than application launcher.

Conclusion

The ZaTab is the most open tablet out there, and it should be on your shopping list if you’re looking for a tablet designed with end-user freedom in mind. This is the ideal device for Android developers or Linux developers looking to shoehorn a traditional Linux distribution onto a tablet. There is a good chance you will see a full Linux distro running on the ZaTab in the future. ZaTabs are in the hands of KDE and Edubuntu developers, and surely on the wish lists of many other free software developers out there. It runs most Android apps flawlessly—oh, and did I say it was rooted out of the box? You don’t have to be a hacker to enjoy this tablet either, with plenty of storage and access to Google’s Play Store and Amazon’s Kindle books, it makes a great media device. As I’m wrapping

up this review in early July, the ZaTab has yet to see official release. Earl at ZaReason tells me there is still one minor software bug to squash before the ZaTab is officially launched—debugging the HDMI output driver to be specific. For the most up-to-date information on when the ZaTab will be shipping, to pre-order one, or to order a developer unit sans OS, visit the ZaReason Shop: <http://zareason.com/shop/zatab.html>. ■

Kevin Bush is a Linux systems admin, dad and book-lover who spends far too much time tinkering with gadgetry. ■

ZaTab

PROS:

- **Brilliant screen.**
- **Rooted out of the box!**
- **No crapware!**
- **Ice Cream Sandwich.**
- **Great battery life.**
- **Solid build quality.**
- **Ample and expandable storage.**
- **Totally hackable.**

CONS:

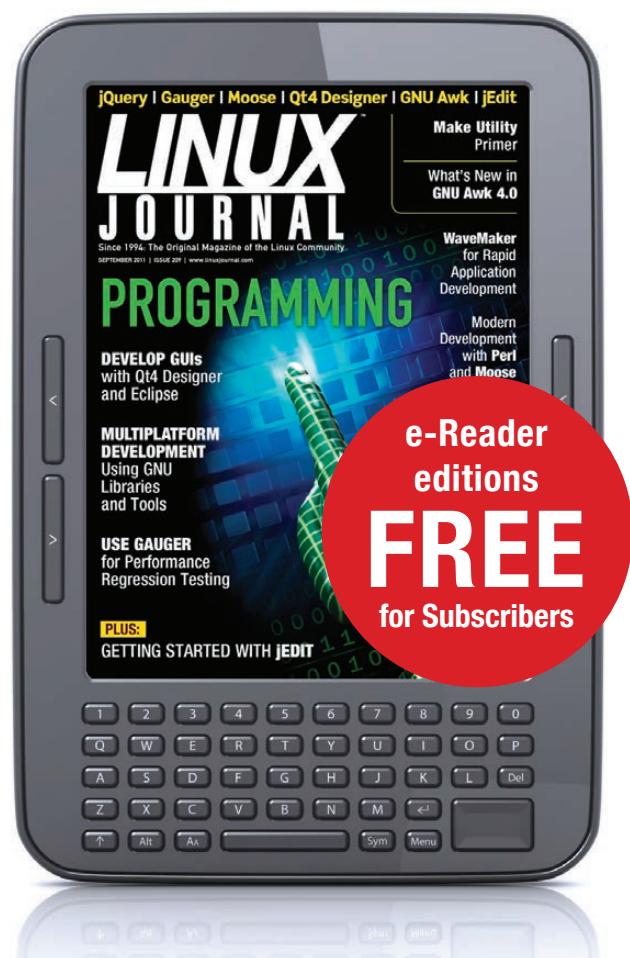
- **A few apps don't play nice.**
- **Single-core processor.**
- **Grainy cameras.**

LINUX JOURNAL

on your
e-Reader

Customized
Kindle and **Nook**
editions
now available

LEARN MORE



LOGITECH SQUEEZEBOX PLATFORM:

Livin' in the Land of (Open-Source) Hi-Fi



Ever wished you could have a house full of music, but afraid to commit to a proprietary platform? The Logitech Squeezebox is an open platform for streaming music all throughout your house and beyond. And, it runs Linux too. CRAIG MALONEY

There's no shortage of options for playing music under Linux. Whether it's local media players, cloud-based music services or streaming music, Linux users are spoiled for choice. But the number of choices diminishes rapidly when you add features like multiroom playback, or add multiplatform support for Windows, Macintosh and smartphones like Android and iPhone.

Platforms like Apple's iTunes support multiple devices via Air Play, but Linux machines can't support Air Play without major effort and hacked-up solutions. Logitech's Squeezebox not only comprises a reasonably priced hardware platform, it also supports several operating systems for both playback and server-side usage. The newer Squeezebox devices (such as the Squeezebox Radio and Squeezebox



Figure 1. Unboxing the Squeezebox Touch

Touch) run on embedded Linux platforms. The Squeezebox server and devices support most major, non-DRM-encumbered formats (including MP3, FLAC and OGG) and support many on-line streaming services, such as Spotify, Pandora, Last.fm and SiriusXM. Logitech also supports an active Squeezebox hacker community and makes the source of both the player and the server freely available.

Logitech's Squeezebox platform is the perfect solution for my listening needs. I use the Squeezebox every chance I can, streaming music from my home machine to my workstation at work via an SSH tunnel. Add a long history of quality hardware devices and an open platform, and you have a compelling reason for every Linux user to consider using the Squeezebox platform. By the end of this article, you'll wonder why you haven't set up a Squeezebox platform of your own, and once you have, you'll wonder why you didn't do it sooner. That's okay, though; there's never been a better time to start.

Brief History

The Logitech Squeezebox player has a long history of development. It started back in 2000 with the formation of Slim Devices. Slim Devices released its first music player, the SLiMP3, in 2001. It was

a wired-only device, capable of playing MP3-only streams. It relied heavily on the Slim Server to perform transcoding duties for other formats. Later, Slim Devices released wireless versions of the SLiMP3 with enhanced displays and support for other formats, such as OGG, WAV, AAC and WMA. The first of these models, the Squeezebox1 (or SB1) supports only WEP over 802.11b, so it's more useful on today's networks via a wired connection. The Squeezebox2 (SB2) adds both WPA encryption, 802.11g wireless communication and native FLAC support. If I were looking for an older Squeezebox unit, the SB2 has the base feature set I would want for any Squeezebox unit. The SB3 (initially named the Squeezebox, later renamed the Squeezebox Classic) is essentially the SB2 with a different case design. Every one of these devices still is supported by the Slim Server (now Logitech Media Server), so if you find one in the wild, be sure to pick it up!

In 2006, Logitech acquired Slim Devices and continues supporting the Squeezebox line with frequent software releases, and with newer hardware to refresh the Squeezebox line. Logitech released the Squeezebox Boom, the first all-in-one Squeezebox device that included stereo speakers, and used the same interface as the Squeezebox Classic. Logitech also released the

Squeezebox Duet, which bundled the Squeezebox Receiver (a headless device capable of playing Squeezebox streams) together with the Squeezebox Controller (a remote-control device capable of controlling any Squeezebox or Transporter device). Both the Boom and Duet have since been discontinued, but the Squeezebox Controller is notable as it was the first device to ship with the new Linux-based SqueezeOS operating system and the Lua-based SqueezePlay interface. Logitech released other Linux-based hardware devices: the Squeezebox Radio and the Squeezebox Touch (which I discuss later in this article). Logitech also sells the Transporter, which is geared toward the audiophile market. The Transporter uses two fluorescent displays (similar to those used in the Squeezebox Classic), and includes upgraded, audiophile-quality hardware (see the Logitech Transporter sidebar). At the time of this writing, the Transporter, Squeezebox Radio and Squeezebox Touch are the only hardware players sold by Logitech.

Okay, enough history. Let's get something set up so you can start playing music!

Getting Started

You'll need both a Logitech Media Server and one or more Squeezebox clients to

make use of the Squeezebox platform. Fortunately, you won't need to make a trip to the store, as both the server and clients are freely available on-line. The server software is available from <http://www.mysqueezebox.com/download> in prepackaged RPM or .deb formats, or as a tarball of Perl source code. The server software does not require a mysqueezebox.com account, but I recommend signing up for one, as some proprietary music services will not work without a mysqueezebox.com account. mysqueezebox.com also will be able to act as a Squeezebox server should you be unable (or unwilling) to connect to a local server.

Follow the installation instructions for your platform to install the Logitech Squeezebox server. Once you have it installed, navigate to your server's address, port 9000 (for this article, I'll use `http://localhost:9000` as the server URL). Enter your mysqueezebox.com credentials (if you have them), and click Next to continue (or Skip). Next, select where your music is located on the server and where to store playlists. Once those are selected, your server is active.

Server

Let me talk a bit about the Media Server. The Media Server is the brains of the Squeezebox platform. The server acts as

FEATURE Logitech Squeezebox Platform

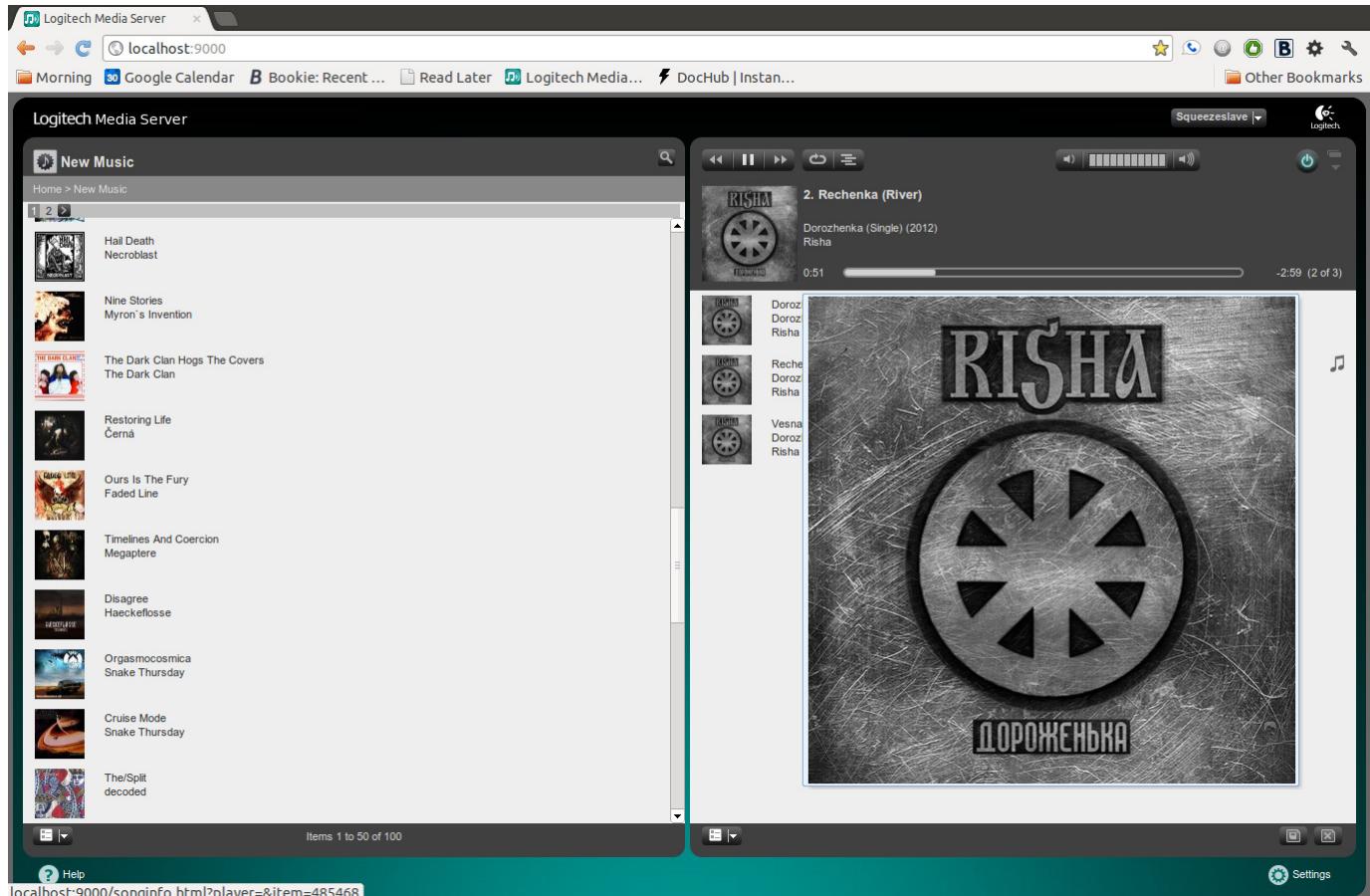


Figure 2. Playing Music via the Squeezebox Server's Web Interface

the repository for files and playlists, as well as a Web-based controller for all of the connected Squeezebox devices. Most functions and options available on the Squeezebox players can be performed using the Squeezebox server. The server has a bunch of settings for setting various functions of the connected Squeezebox devices. One of those that I use determines the quality of output to the Squeezebox devices. I have a work Squeezebox client that I don't want to

deliver full-sized FLAC files to, so I tell the Media Server to transcode those files to 160Kbps MP3 files before sending them to my client. There are way too many settings to cover in this article, but suffice it to say that there are a bunch of ways to configure the server and clients. The Media Server also is extensible using a variety of plugins and applications, which can customize the Logitech Media Server to fit your music listening needs.



Figure 3. Harmoniously Synchronized (Left–Right: Squeezebox Classic, Squeezebox Touch, Squeezebox Radio, Squeeze Commander/Squeeze Player on Android, Squeezeslave on ASUS EEE 701, SoftSqueeze)

Synchronicity

Because the Squeezebox keeps a running tab on each of the players currently playing, it makes it very easy to see what a particular player is playing. But, what's even cooler is syncing several players together. Navigate to the player drop-down list to select the player you want to start with. Next, select the player drop-down list, and select **Synchronize....** The server will show a list of current players. Select the player you want to sync with, and get ready to hear music in both rooms at the same time. (If you notice they're not perfectly in sync, don't worry; there are plenty of options on the server to take care of most synchronizing issues.) Let the Squeezebox pipe music throughout your house at your next party!

Current Hardware Players

Logitech currently sells three hardware Squeezebox clients: the Squeezebox Radio, the Squeezebox Touch and the Transporter. I've personally used the Squeezebox Radio and Squeezebox Touch, as well as an older Squeezebox Classic, so I discuss those units in more detail here. Both the Squeezebox Touch and Squeezebox Radio are ARM-based Linux devices using a real-time customized Linux distribution called SqueezeOS. User interactions are handled by SqueezePlay, a Lua-based front end for interfacing with Squeezebox devices. The Squeezebox Radio features a single bi-amplified speaker, an 1/8" input jack and an 1/8" headphone jack. The Radio comes in three different colors: black, red and white (white is available exclusively from the Logitech on-line store). It's perfect as a standalone radio (I use my Squeezebox Radio in the bedroom as a clock-radio and occasionally in the kitchen when doing the dishes).

There are six hardware preset buttons on the Squeezebox Radio, which can be used for marking Internet Radio stations or other favorite songs. The interface centers around a jog-dial for selecting menu items and is pretty intuitive to use. The Squeezebox Touch is also a SqueezeOS-based ARM device, but it

forges the speaker in favor of digital and analog connections. It features both coaxial and optical S/PDIF digital ports, as well as RCA stereo and 1/8" headphone analog outputs. The Squeezebox Touch also adds a slot to accept SD card media and a USB port for external drives or other media. These both allow the Squeezebox Touch to play audio files without using a server, and also allow the Squeezebox Touch to act as a small Media Server. (Logitech recommends the Squeezebox Touch serve only a small number of clients and fewer than 5,000 files, as it doesn't have the CPU to handle larger amounts.) The Touch interface is extremely touch-friendly, using swipes to scroll selections of songs, and featuring an on-screen keyboard for textual input. This gives it a slight improvement over the jog-dial interface of the Squeezebox Radio. The Squeezebox Touch also ships with a full-featured remote control with a directional pad, a numeric pad that handles alphanumeric entry, and other useful remote functions like volume, favorites and more. The Squeezebox Radio offers both the remote and an internal battery as options, which are available via the Squeezebox Radio Accessory Pack. The remote for the Squeezebox Radio is a reduced-functionality remote

compared with the remote that ships with the Squeezebox Touch (it forgoes the numeric input) but the Squeezebox remotes are interchangeable.

All of the Squeezebox devices I've tried have great sound. The Squeezebox Touch has the advantage of more standard connections (especially if you currently have digital inputs on your receiver) and a very user-friendly interface. I find the Squeezebox Radio is great for portable applications (especially with the optional battery), while the Squeezebox Touch fits perfectly into our home audio system.

Software Players

One of the benefits of the Logitech Squeezebox platform's openness is the number of software player options available. One of those is Logitech's own SqueezePlay software. The SqueezePlay software is the same interface as the Squeezebox Touch or Radio. It is available for Windows, Macintosh and Linux. Unfortunately, I had little success getting it to work under my Ubuntu machines, but it worked well under the Windows system I tested. What works well on my Linux machine is a terminal-based program called Squeezeslave. Squeezeslave is a C-based program that emulates the interface of the Squeezebox Classic devices faithfully (it even requires you to use the numeric

Logitech Transporter

I haven't covered the Squeezebox Transporter, but suffice it to say that if you have the budget for a higher-end digital player, this would be the one to look at. It boasts audiophile components, but uses the older-style Squeezebox client software (similar to the Squeezebox Classic). It has two individually addressable fluorescent displays, and it adds two XLR connectors, infrared in/out jacks, as well as the normal digital and analog connectors. If you measure your listening environment with an oscilloscope or are looking for a Squeezebox for a custom installation, definitely check out the Squeezebox Transporter.

The Spirit of Radio

The Logitech Media Server ships with a bunch of Internet streaming and radio services ready for configuration. I was able to set up most of the local radio stations by just entering my zip code. The Logitech Media Server uses a service called TuneIn (formerly RadioTime) to determine what local radio stations are available. Not only was I able to select most area stations, I also was able to find the streaming NOAA Weather Radio station, as well as police and fire stations. Additionally, I found low-power radio stations that have on-line streaming, so I can listen to college radio stations that would have required a special antenna to receive. And, that's without scratching the surface of the rest of the world's Internet-available radio stations. It's like having the best of radio right at home.

keys to enter search text, as if you were using a remote). Squeezeslave boasts excellent sound quality, and it can be run as a daemon so you won't have to dedicate a terminal to use it.

Another excellent player is the Java-based Soft Squeeze. It is a more graphically faithful version of the Squeezebox Classic devices, with a variety of skins (including some that look like the Squeezebox Transporter, Boom and Classic). Both of these applications fit nicely with my listening habits, because I can use the same server for both home and work. I've set up an SSH tunnel at work to ports 9000 and 3483 (the stream and control ports, respectively) and have access to both my large library of songs, my list of radio streams via the Squeezebox server and one of the aforementioned clients (Squeezeslave, primarily). So, if the mood strikes me to listen to Kendra Springer at work and all I have on my phone is Death Metal, I can hook up to my Squeezebox server at home and listen to all the Kendra Springer I want. (Hey, sometimes it happens!)

In addition, several applications are available for Android, iPhone and Nokia devices. Logitech's own Squeezebox Controller (available for both Android and iPhone) acts as a controller for any Squeezebox device hooked to a server.

The menus and interface are similar to the Squeezebox Touch interface, and it works as you might expect. The third-party applications are really where the power of the Squeezebox platform is realized. iPeng (available on the Apple iPhone) acts as a Squeezebox controller application, but for a few dollars more, you can unlock a receiver application that turns your iPhone into a portable Squeezebox receiver. This allows you to control and listen to your Squeezebox music as far as your network will let you. Android users have the option of purchasing a separate application to act as a Squeezebox receiver: Squeeze Player (not to be confused with the SqueezePlay interface). Squeeze Player acts only as a Squeezebox Receiver, but when paired with the Logitech Squeezebox controller application, it becomes a very capable remote player. Android users have the option of using a very cool third-party controller application: Squeeze Commander. Squeeze Commander has all of the functionality of the excellent Logitech Squeezebox application, but also includes a bunch of features. Most notably is the ability to download music files available to the Squeezebox directly onto the Android Device. I find this extremely handy, as I can download my music without needing to have the phone hooked directly to the computer. Nokia users have the

option of using Squeezester controller, though I'm not aware of any receiver applications for those devices. There are also similar Squeezebox receiver and controller projects for XMBC in various stages of development. Check on-line to see if there's Squeezebox support for your

platform. You may be pleasantly surprised. (And if there isn't, you'll have all of the tools and documentation to create one.)

Applications and Plugins

Linux users are usually left to their own hackish solutions whenever it comes

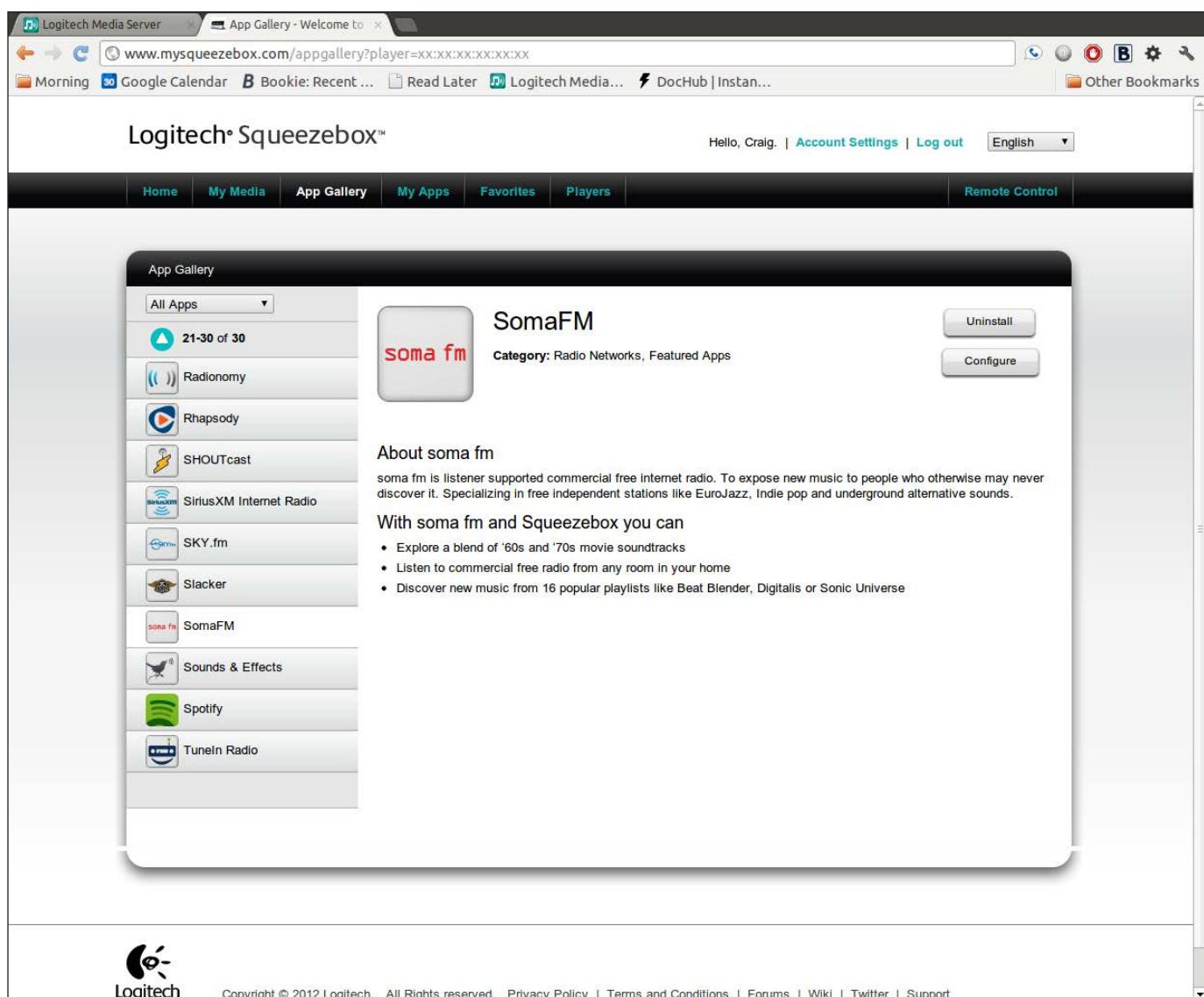


Figure 4. You can install applications for the Squeezebox server via mysqueezebox.com.

to streaming services like Pandora and Spotify, or for streaming Sirius XM satellite radio. With the Logitech Squeezebox player, support for these services (as well as many other Internet-radio services like Spotify and Soma.fm) are installable via applications or plugins. Application installs are handled from your My Squeezebox account. Select the application you want to install, click install, and your Media Server will install the application.

The Squeezebox supports many different music services, as well as Facebook and Flickr. Yes, you read that right: Flickr. Because the Squeezebox Radio and Squeezebox Touch sport color LCD screens, you can have them perform a slideshow of Flickr images on the device. There also are hundreds of plugins available for the Squeezebox, many of which were written by third-party authors. These cover myriad uses, like adding UPnP/DLNA capabilities to the server, switching playlists and positions from one player to another, and many more. There is an active plugin community available, and extensive documentation for creating plugins inside the Help menu on the server (Help→Technical Information→Logitech Media Server Plugins).

Peeking Under the Covers

Logitech and Slim Devices went the

extra mile to make the Squeezebox platform controllable and extensible. The Squeezebox server ships with a telnet-addressable command-line interface. Telnet to port 9090 of your server, and you can control every aspect of your Squeezebox server and connected devices. You can learn more about the commands via the help documentation (Help→Technical Information→Command Line Interface).

Better still, all of the SqueezeOS-based players (Squeezebox Radio, Touch and Duet) have an SSH server built in (which is turned off by default, but it can be enabled simply by navigating to Home→Settings→Advanced→Remote Login→Enable SSH). Once enabled, ssh to it with the default credentials (root/1234). Once inside, the best Message Of The Day (MOTD) I ever have seen is displayed:

This network device is for authorized use only. Unauthorized or improper use of this system may result in you hearing very bad music. If you do not consent to these terms, LOG OFF IMMEDIATELY.

Ha, only joking. Now that you have logged in feel free to change your root password using the

'passwd' command. You can safely modify any of the files on this system. A factory reset (press and hold add on power on) will remove all your modifications and revert to the installed firmware.

Once you've logged in, you have access to all of the internals of the Squeezebox device. This is one of the most forward-thinking moves I've seen from any hardware manufacturer. It came in handy for diagnosing some trouble I had with alarms on the Squeezebox Radio. (When was the last time you ran `tail` on your alarm clock?) Logitech even releases the source code for SqueezeOS and includes instructions on how to build and flash the firmware (handy for those of you who never run stock-anything on your hardware).

If you really want to dig in to the capabilities of the Logitech platform, the server ships with some of the most comprehensive documentation I have ever seen for a hardware product. Logitech ships the Media Server with gorgeous and thorough documentation. Hidden under the normally useless moniker of "Help", Logitech provides a comprehensive pile of documentation about the internals of the Squeezebox Protocol, how to create skins for the

server, how to create plugins for the Media Server, the display API and so much more. There is also documentation for xPL, which is an automation protocol I learned about while skimming the documentation. The server can support xPL calls via enabling a plugin. With some hardware hacking, it's entirely possible to set up near-field communications with a device that notifies your Media Server to play "The Imperial March" from *Star Wars* on all of your Squeezebox devices whenever you come home. (Note: the author takes no responsibility for the other residents' reactions if you do this.)

I hope this taste of the Squeezebox Platform inspires you at least to download the server and one of the software clients. I know it has opened up a world of possibilities for my music listening. Few of the solutions I've tried boast the interconnectivity of the Squeezebox platform. None of them come close to the openness and control of the Squeezebox. Using your phone or Web browser to control every music player in your house is a liberating experience. Being able to listen to the same collection of music and radio streams remotely from my home machine is like a dream come true for me. Having permission to play, tinker and expand the platform (with excellent documentation

and a full open-source stack) is unheard of in the higher-end audio space. May your exploration of the Logitech Squeezebox Platform make your listening experience more enjoyable.

Acknowledgements

Special thanks to Logitech Corporation for its assistance with this article. ■

Craig Maloney is the host of Open Metalcast (a Creative Commons Metal Podcast at <http://openmetalcast.com>) and the co-host of Lococast.net. He's also the contact for the Ubuntu Michigan Loco, and board member for the Michigan UNIX Users Group. When he's not listening to music on his Squeezebox, he's enjoying time with his lovely wife JoDee, developing stuff only a compiler could love, and playing various tabletop and computer games. He can be reached at his site: <http://decafbad.net>.

Resources

Logitech Squeezebox: <http://mysqueezebox.com>

Squeezebox Radio: http://wiki.slimdevices.com/index.php/Squeezebox_Radio

Squeezebox Touch: http://wiki.slimdevices.com/index.php/Squeezebox_Touch

iPhone Applications:

- Logitech Squeezebox Controller:
<http://itunes.apple.com/us/app/logitech-squeezebox-controller/id431302899?mt=8>
- iPeng: <http://penguinlovesmusic.de>

Android Applications:

- Logitech Squeezebox Controller:
<https://play.google.com/store/apps/details?id=com.logitech.squeezeboxremote>
- Squeeze Commander:
<https://play.google.com/store/apps/details?id=de.cedata.android.squeezecommander>
- Squeeze Player:
<https://play.google.com/store/apps/details?id=de.bluegaspode.squeezeplayer>

Nokia Squeezester: <http://talk.maemo.org/showthread.php?t=78966>

SqueezeOS Documentation:

- Squeezebox Protocol: http://wiki.slimdevices.com/index.php/SlimProto_TCP_protocol
- SqueezeOS Documentation: <http://wiki.slimdevices.com/index.php/SqueezeOS>
- SqueezeOS Architecture Diagram: http://wiki.slimdevices.com/index.php/SqueezeOS_Architecture

Join Us For
Ohio Linux Fest
2012



September 28-29
Greater Columbus Convention Center
400 North High Street
Columbus, Ohio

ARDUINO TEACHES OLD CODER NEW TRICKS

**Using Linux open-source hardware design tools
to create an Arduino-inspired hardware project.**

EDWARD COMER

I became aware of the Arduino Project from occasional media reports and a presentation at Atlanta LinuxFest 2009. I was impressed with what the Arduino community was doing, but at that time, I saw no personal use for it. It took a grandson who is heavily involved in a high-school competitive robotics program to change things for me. During a 2011 Thanksgiving family

gathering, he asked me some questions about robotics-related electronics, and I told him to google Arduino. He did. Arduino ended up on his Christmas list, and Santa delivered.

I would be more helpful in assisting the grandson's Arduino efforts if I understood more about it myself, so I ordered a couple Arduino Nanos and some peripherals, such as rotors, servos,

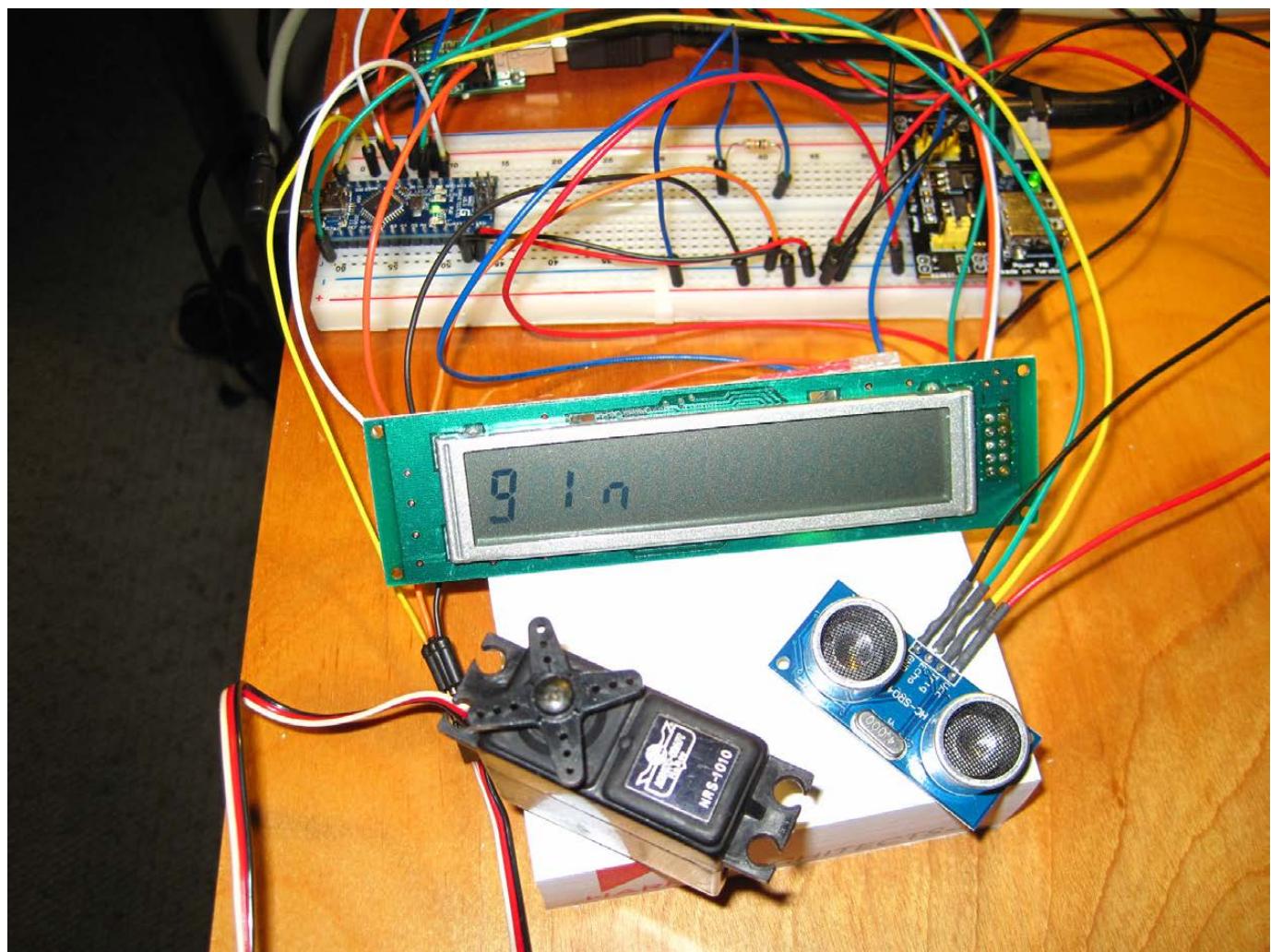


Figure 1. Arduino Pro Mini in Breadboard Tests

ultrasonic sensors and LCD displays, and dug in. I now had a purpose for using the Arduino and a reason to dust off my soldering iron. I used a breadboard for testing, as shown in Figure 1.

It didn't take very long to remove the mental cobwebs and get into the elegant simplicity of the Arduino Project. Years ago, when I built microprocessor projects, the underlying system code always was the problem. Before I actually could write my application, I had to develop or adapt systems-level code to interface the application-level code with the underlying hardware. It was always a major pain and, quite frankly, drudgery. The Arduino Project does away with worrying about most of the low-level systems code, leaving you with the now much-simplified task of creating your application. Using the Arduino IDE and included or contributed libraries enables you to interface to a plethora of hardware easily. Anyone who has developed in the C and C++ languages will find the Arduino platform easy to master quickly. Although Arduino is actually based upon the Wiring Project, compatibility with C, C++ and Linux are very high.

After implementing and testing code for the various peripherals that I had accumulated and generally mastering the

Arduino platform, I said to myself, "now what?" So, I abandoned the nice Arduino IDE and switched over to developing code using Linux tools, such as Make. I also wanted to get closer to the hardware, so I abandoned the Arduino boards and did my implementations on the underlying ICs used by all Arduino boards, the Atmel 8-bit series of microcontrollers. Using the Arduino libraries with the Atmel microcontrollers is a joy to behold. I am so thrilled that the drudgery of systems code can be mostly ignored as it is mainly handled by the hardware abstraction features of Arduino's built-in libraries. It is important to note that the Atmel ICs are microcontrollers, not microprocessors. In other words, they are almost complete computers equipped with RAM, EPROM and Flash memory, multidirectional I/O, serial ports (in some cases) and interface circuitry (such as pull-up resistors and so on). Just adding a power source will yield a computer in a chip.

The hardware's interfaces of the Atmel microcontroller are abstracted by Arduino in a uniform way—at least, uniform for those Atmel microcontrollers implemented by the Arduino group. Arduino libraries use internal code, generically called the "core", to define the available I/O pins on a given Atmel microcontroller, assigned by pin number. For example,

the Atmel ATmega168 physical pin 4 is defined as Arduino I/O pin 2, yet with the Atmel ATmega32u4 microcontroller, the same Arduino pin 2 is matched to physical pin 19. Thus, the Arduino syntax of “pinMode(2, OUTPUT)” defines, in software, an abstracted hardware pin as a digital output. Because the Arduino module pins are labeled with Arduino pin numbers, the abstraction becomes physical, at least on the module level. Nonetheless, it is this abstraction as well as robust libraries that enable the Arduino to be so easy to work with. One caveat alluded to above is that Atmel microcontrollers not implemented in Arduino modules don’t have uniform core definitions—for example, the Atmel Attiny series. You still can use the Arduino libraries and tools, but the cores must be obtained elsewhere. For the Atmel ATtiny85 and ATtiny84 microcontrollers, I use the core from the code.google project named arduino-tiny. However, there are other, competing cores around for these chips, and they are *not* necessarily compatible.

Burning your program into an Arduino module is extremely easy to accomplish. The USB connection not only can power the module as well as serve as the serial communications interface, but the Arduino IDE also uses it to install your

program into the Flash memory. It is more complex with the Atmel ATTiny series, because they have no USB port or even a hardware serial port, for that matter. For the ATTiny series, you must use an external programmer. Many people use an Arduino board as the programmer once they have loaded the ArduinoISP software, or sketch, as programs are named in the Arduino world. In my case, I chose to use a dedicated programmer called a USBasp. It is readily available on eBay, or you even can make your own with plans from its creator, Thomas Fischl. I purchased mine on eBay because it was cheaper than the parts cost to make my own. The USBasp uses the open-source AVRDUDE software.

The Project

Now that I had invested a lot of time into learning the Arduino system and the Atmel microcontrollers, I wanted to take the next logical step: move a design from the breadboard to a printed circuit board. Some interesting projects exist in this area, such as Fritzing, which is designed to facilitate doing exactly that. It’s a clever project and you should check it out, but I took a different path—using the gEDA open-source Linux software suite for printed circuit development.

I looked at my inventory of parts and started thinking of what I could create that wasn't already readily available. I settled upon the LCD display. The displays being used in Arduino projects were interfaced with a lot of I/O pins and code space, neither of which is in great supply on the Atmel chips. I felt that if I could create a same-size daughter board that I could attach onto the back of the display and put the smarts into the board that would communicate with the LCD display via an ASCII serial interface, I would have something useful that didn't exist in the marketplace in an affordable form. This is commonly called a serial LCD.

Being somewhat of an old-timer, I spent a lot of time using and coding for the DEC-VT100 display terminals upon which the ECMA-48/ANSI X3.64 standards are based. I felt that if I coded the daughter board to turn an LCD display into a tiny, affordable DEC-VT100, I would have something reasonably unique and useful. Serial-driven LCD displays do exist, but they typically have proprietary protocols, and some are rather expensive. As far as I have been able to determine, there exists no open-source (software and hardware) serial LCD display with VT100 protocol. I found my project!

Gathering the Parts

I selected parts for the VT100-LCD project, such that the parts would be as affordable as possible. In fact, I purchased all the parts from two sources, eBay and Digi-Key, based on cost. Table 1 shows the required materials to build one vt100lcd. Costs are shown on a per-item basis; however, I purchased most of these items in quantities of five or more.

Schematic Design

To design the circuitry for the VT100-LCD, I chose gschem of the gEDA Project at <http://geda-project.org>. This suite includes not only the schematic design program but also a PCB layout program, as well as various helper programs. A number of schematic/PCB design software programs exist, but I'm focusing on the open-source software of the gEDA Project by geda-project.org here. Other open-source projects that run on Linux, include KiCad, as well as several commercial products, the most popular of which is Eagle PCB by CadSoft, which runs pretty well under WINE.

gschem is fairly straightforward, and many functions are intuitive, but a few, useful but arcane commands necessitate printing out a cheat sheet (hey, I'm getting older and I can't memorize all of those keystrokes). Yes, although gschem

Table 1. vt100lcd Parts List

PART	QTY	SOURCE	COST
1602 HD44780 LCD	1	eBay seller (China)	\$2.95
Atmel ATtiny84	1	Digi-Key ATTINY84-20PU-ND	\$3.01
Switch, tactile FSM4JH	1	Digi-Key 450-1650-ND	\$0.80
Socket, IC, 14-pin	1	eBay seller (USA)	\$0.15
Header, 1X20, Female, 2.54mm	1	eBay seller (China)	\$0.39
Header, 1X40, Male, 2.54mm	1	eBay seller (China)	\$0.20
Resistor, 330 ohm, 1/4W	1	eBay seller (Thailand)	\$0.02
Resistor, 10k ohm, 1/4W	1	eBay seller (Hong Kong)	\$0.02
Pot, trim, 5k, RM-065	1	eBay seller (USA)	\$0.30
Capacitor, .1uf, ceramic disc, 50V	1	eBay seller (Hong Kong)	\$0.05
Transistor, 2N3906	1	eBay seller (Thailand)	\$0.01
Diode, 1N4148	1	eBay seller (Thailand)	\$0.01
			Total: \$7.91
Optional Parts			
Commercial PCB	1	Panel Aggregator	\$7.43
Capacitor, 22pf, ceramic disc, 50V	2	eBay seller (USA)	\$0.40
Crystal, 20MHz, ATS200-E	1	Digi-Key CTX1105-ND	\$0.64

is a GUI program, useful keyboard shortcuts appear nowhere in the GUI's menus. This is especially true of the PCB layout program that I discuss later.

The process consists of inserting electronic component symbols into the schematic drawing either from the built-in library or from your private library and then connecting the pins by drawing traces. I highly recommend reviewing the gEDA Project's on-line documentation before starting your own schematic. There are a few tutorials on the Web about using the gEDA suite, and Stuart Brorson wrote

a tutorial article in the November 2005 issue of *Linux Journal* (see Resources).

I created two versions of my VT100-LCD project: one using the eight-pin ATtiny85 microcontroller and another using the 14-pin ATtiny84 microcontroller. The schematic for the ATtiny84 microcontroller version is shown in Figure 2.

Because some of the components I was using do not exist in the built-in library, I scoured the Internet for contributed symbols, and in a few cases, I had to design my own symbols. A good source for contributed symbols and footprints

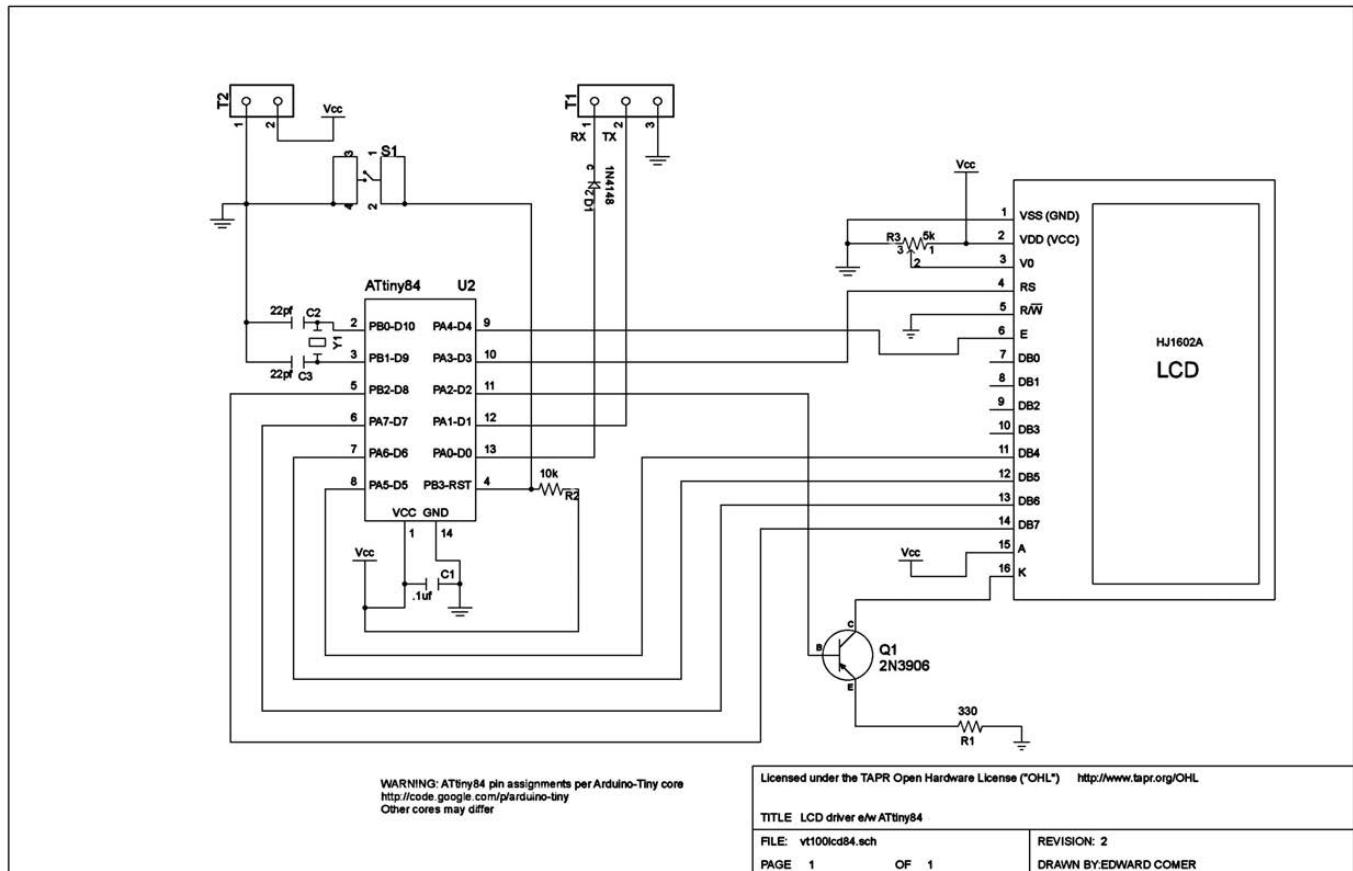


Figure 2. Schematic for VT100-LCD e/w ATtiny84

is <http://gedasymbols.org>. For creating your own symbols, see David Weber's Online Symbol Creation Tool at <http://EmbeddedToolBox.com>. Symbols actually are text files. Figure 3 illustrates a symbol along with a portion of the text file used to draw it. Symbol files are not just an image. They also hold important pin definitions and the name of the footprint file that the gEDA PCB program ultimately will use to represent the component on the circuit board.

A gEDA schematic is a text file

interpreted for GUI presentation by gschem, but it also serves as the source for gEDA's PCB program. An intermediary helper program named gsch2pcb is used to prepare the schematic file for use as input to the PCB program. While xgsch2pcb is a GUI version of gsch2pcb for gsch2pcb, I use the gsch2pcb command-line version. For example, given the schematic file vt100lcd84.sch as an input, gsch2pcb creates vt100lcd84pcb, vt100lcd84.net and vt100lcd84.cmd, all necessary files for PCB creation. gsch2pcb

also displays important instructions as part of its command-line text output. To make the process a little easier, I use a file named “project” in the project folder for the current design. Figure 4 shows my project folder for the vt100lcd84 project, the “project” file and the command line with the gsch2pcb command just before execution.

It is worth noting that the gEDA suite includes circuit simulation capability (SPICE), enabling virtual design testing.

```
v 20120206 1
P 2100 1700 1800 1700 1 0 0
{
T 1900 1750 5 8 1 1 0 0 1
pinnumber=5
T 1900 1650 5 8 0 1 0 2 1
pinseq=5
T 1750 1700 9 8 1 1 0 6 1
pinlabel=PB0
T 1750 1700 5 8 0 1 0 8 1
pintype=io
}
P 2100 1300 1800 1300 1 0 0
{
T 1900 1350 5 8 1 1 0 0 1
pinnumber=6
T 1900 1250 5 8 0 1 0 2 1
pinseq=6
T 1750 1300 9 8 1 1 0 6 1
pinlabel=PB1
T 1750 1300 5 8 0 1 0 8 1
pintype=io
}
...
device=Attiny85
T 400 2600 5 10 0 0 0 0 1
footprint=DIP8
T 400 2800 5 10 0 0 0 0 1
author=Edward Comer
T 400 3000 5 10 0 0 0 0 1
documentation=http://www.msarnoff.org/chipdb/ATtiny85
T 400 3200 5 10 0 0 0 0 1
description=8-bit AVR microcontroller
T 400 3400 5 10 0 0 0 0 1
rfdes=U?
```

Figure 3. Symbol Example 1

I did not use SPICE with my VT100-LCD project, but see the Resources for this article if you are interested.

Software Design

Now that I had the circuitry designed for the project, it was time for the software. I wrote the software as a simple state machine that parses each character received on a character-by-character basis, meaning that there is no buffer. Characters are handled differently



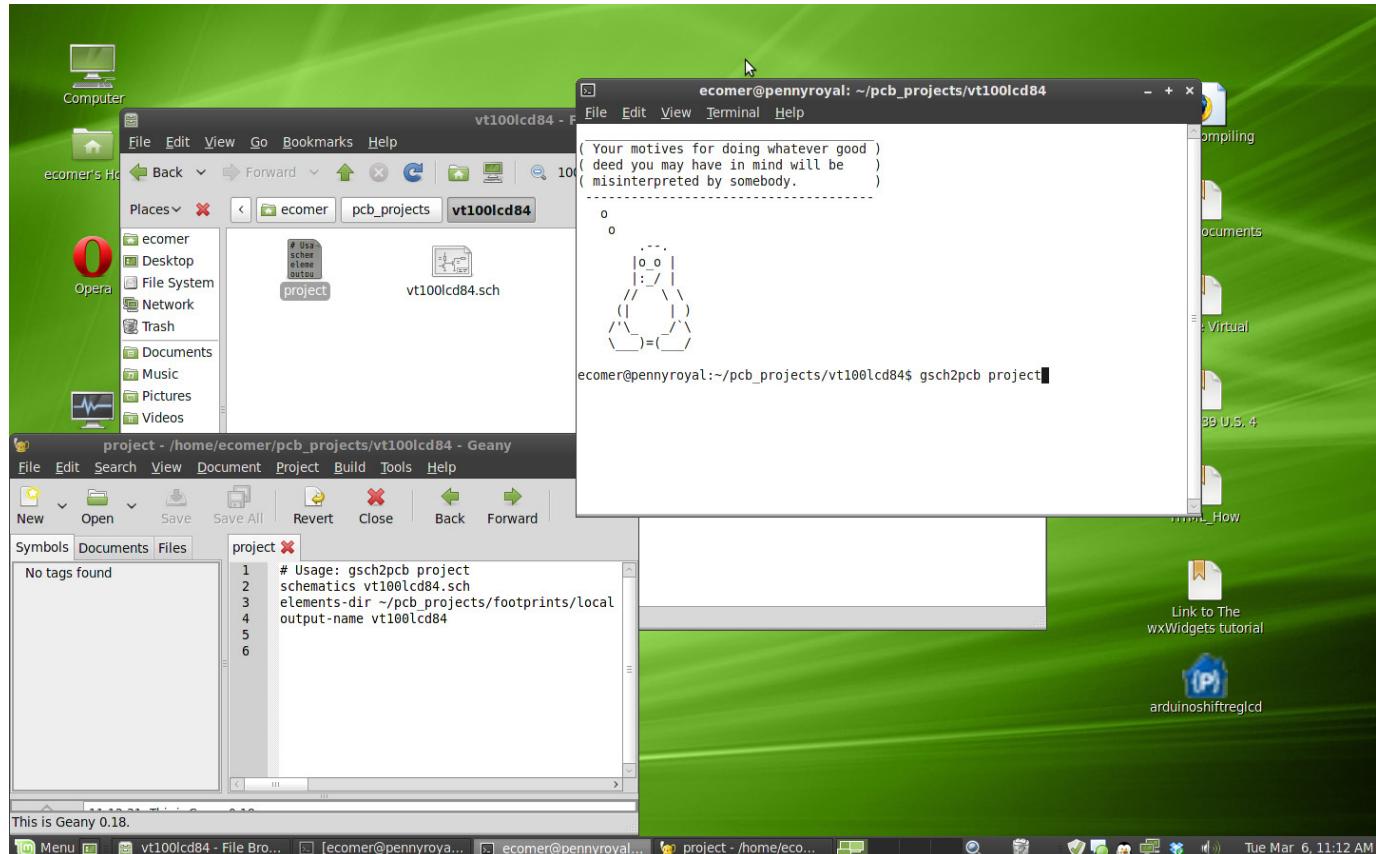


Figure 4. Example of gsch2pcb Project File

based upon the current state of the machine. If the state is NOTSPECIAL, the character simply is passed to the LCD screen for display. However, if the state is GOT_ESCAPE, GOTBRACKET or INNUM, the character is processed further. For example, if the state is GOTBRACKET, both an escape and left-bracket character have been received previously, and the current character must be parsed in that context. For illustration, the VT100 sequence for Screen-Clear is \033[2J, and if the current character being parsed was

the 2, the state would be GOTBRACKET, and the next state would be INNUM (number collection).

This method of parsing has the advantage of simplicity, which is suitable for the limited-capacity microcontrollers but with the disadvantage of not being able to scroll the screen due to the absence of a buffer holding a copy of what is on the screen. See Resources for a copy of the software source. I used Arduino libraries to build the code. Although the source can be compiled using the

Supported VT100 Commands

Return	Cursor to leftmost of current line
Linefeed	Cursor down
Esc c	Resets LCD
Esc D	Cursor down
Esc M	Cursor up
Esc E	Move cursor to start of next line
Esc [A	Cursor up one line (arrow key)
Esc [B	Cursor down one line (arrow key)
Esc [C	Cursor right one column (arrow key)
Esc [D	Cursor left one column (arrow key)
Esc [H	Cursor to HOME 1;1
Esc [s	Save cursor position
Esc [u	Restore to saved cursor position
Esc [m	All attributes off
Esc [Pn A	Cursor up Pn lines
Esc [Pn F	Cursor up to column 1 of Pn lines
Esc [Pn B	Cursor down Pn lines
Esc [Pn E	Cursor down column 1 of Pn lines
Esc [Pn C	Cursor right Pn characters
Esc [Pn D	Cursor left Pn characters
Esc [Pn G	Cursor to column Pn of current line
Esc [2 J	Erase Screen and home cursor
Esc [P1;PcH	Direct cursor addressing, where P1 is line#
Esc [P1;Pcf	Same as above
Esc [= Pn h	Set (LCD lines) - Pn 2 = 16X2 - 4 = 16X4
Esc [0 m	All attributes off (underscore cursor off)
Esc [4 m	Underscore on
Esc [0 c	Report terminal type
Esc [5 n	Reports max qty lcd lines (1s based)
Esc [6 n	Reports cursor position (1s based)
Esc [0 q	Turn LCD's LED 1 off
Esc [1 q	Turn LCD's LED 1 on

Arduino IDE, I used Linux make. Using the Arduino libraries makes the project extremely easy to build. Most of the

drudgery of low-level code and the bootloader is hidden away within the Arduino libraries, which freed me to focus solely on my project. Even main() is hidden away such that Arduino code contains two required routines: setup() and loop(). Main actually does exist deep in the Arduino directory structure in ~/arduino/arduino-1.0/hardware/arduino/cores/arduino/main.cpp and is automatically linked in at compile time.

PCB Layout

When the .pcb file is opened with gEDA's PCB program and the commands invoked are listed in the gsch2pcb command-line text output, you are presented with a jumble of components. I first dispersed the components manually with an approximate

placement and then activated the "rats-nest" display. The "rats-nest" is the connections that must be converted

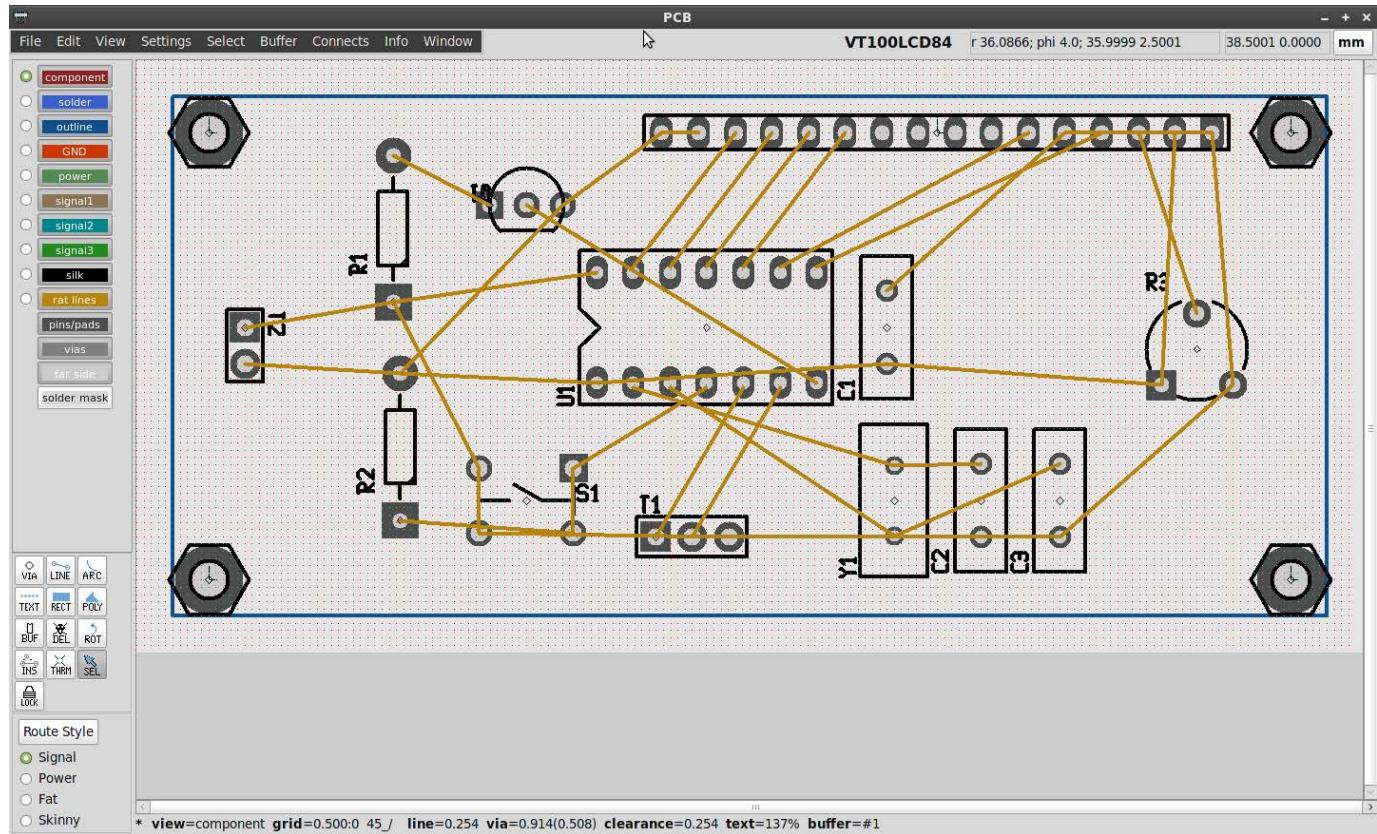


Figure 5. Components with Associated Rats-Nest

to copper traces. After shifting around components to visually bring the “rats-nest” connections to their shortest routes, I was presented with what you see in Figure 5.

PCB possesses the ability to auto-route the traces, namely convert the “rats-nest” into copper trace representations. This tends to do some odd things but produces a workable PCB design once a cleanup is done. I chose a semi-manual layout so I could control the placement and appearance. Basically, I used the

auto-route for the power traces, did some manual cleanup, then used auto-route for the signal traces, followed by more cleanup. The result was similar to Figure 6, which is my second and final version of the layout for the ATtiny84 version of my VT100-LCD project. PCB Manufacture Printed Circuit Board layout consists of applying upon a copper laminated board an acid-resistant pattern that represents the areas that are to retain copper after etching in an acid solution (etchant). Areas of

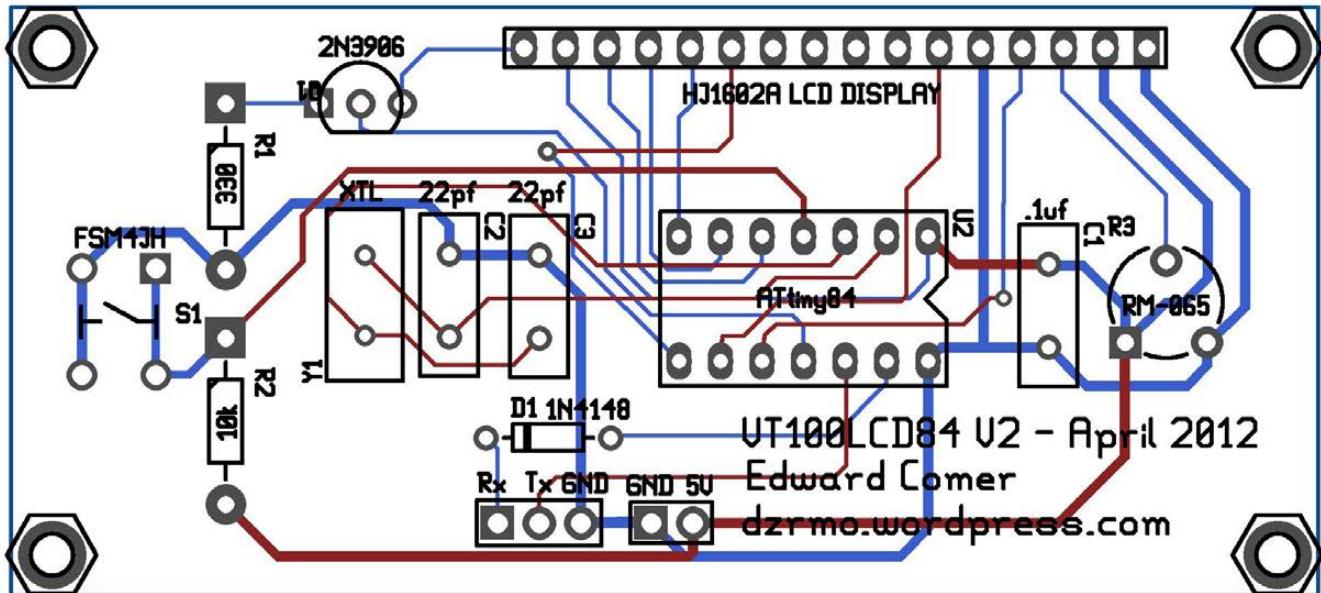


Figure 6. Final Layout of vt1001cd84 Project

the copper laminated board that are exposed to the etchant will be dissolved away, leaving the areas under the acid-resistant pattern intact.

Years ago, I occasionally used to make Printed Circuit Boards using a photographic method that is less common in the DIY community today. The acid-resistant pattern was laid out by hand onto translucent or clear drafting paper using fine black tape for circuit paths and dry transfer patterns for components. This pattern is typically a positive, similar to Figure 7, so a negative must be made photographically for the process to work. The end result is that the negative's acetate sheet is clear where copper should remain after

etching. This photographic work formerly was done in a darkroom, but today, creating the negative can be done using a computer printing to a transparency sheet. I don't discuss the process here, but an example is shown in Figure 8.

Next, the prepared negative pattern would be affixed on a copper laminated board that has a light-sensitive diazo-type emulsion as a top layer. Exposing the prepared PCB to ultra-violet light would alter the properties of the exposed (clear) areas that received the ultra-violet light. Washing the exposed board in a chemical developer dissolved the exposed portions of the emulsion, leaving intact the emulsion that was under the black portions of pattern on prepared negative

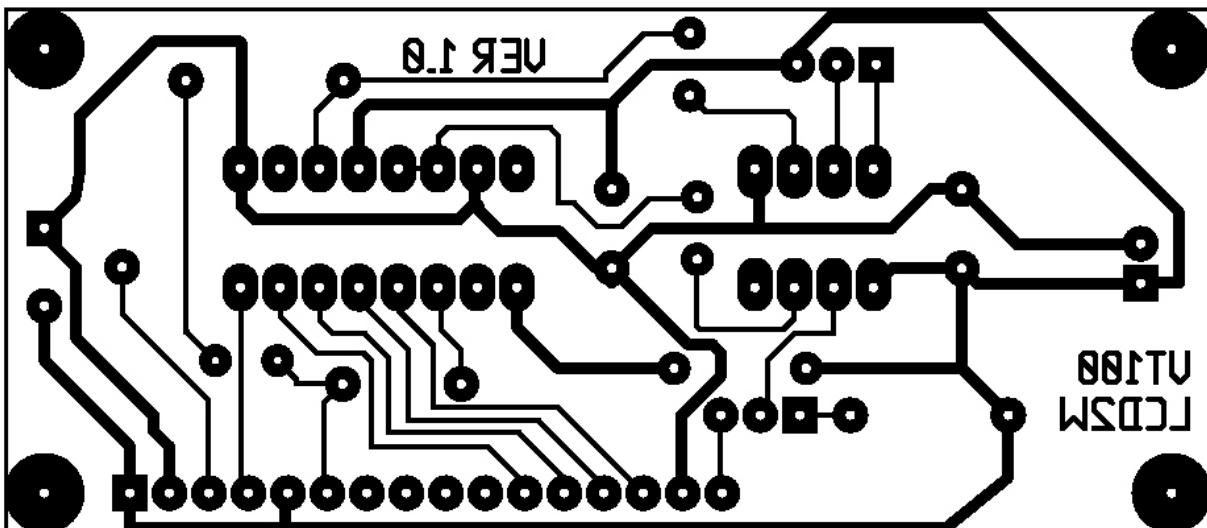


Figure 7. Layout Positive

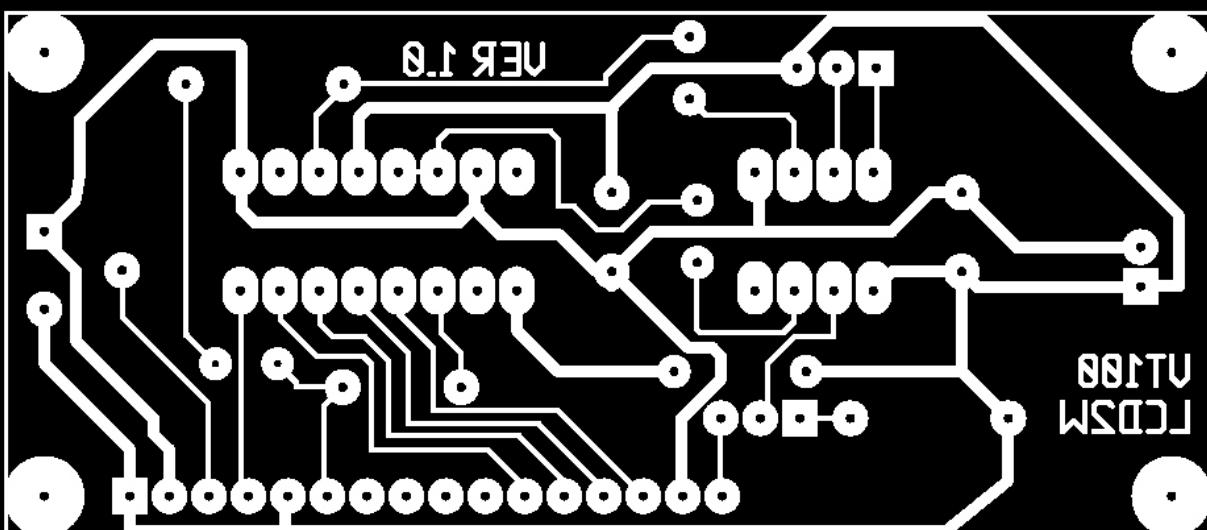


Figure 8. Layout Negative

sheet. Many commercial systems still do a modernized variation of this process as do some serious DIYers.

The casual DIY community has,

thankfully, adopted a new and much easier method of PCB layout for medium-density layouts. High-density layouts still should use the photographic process. The

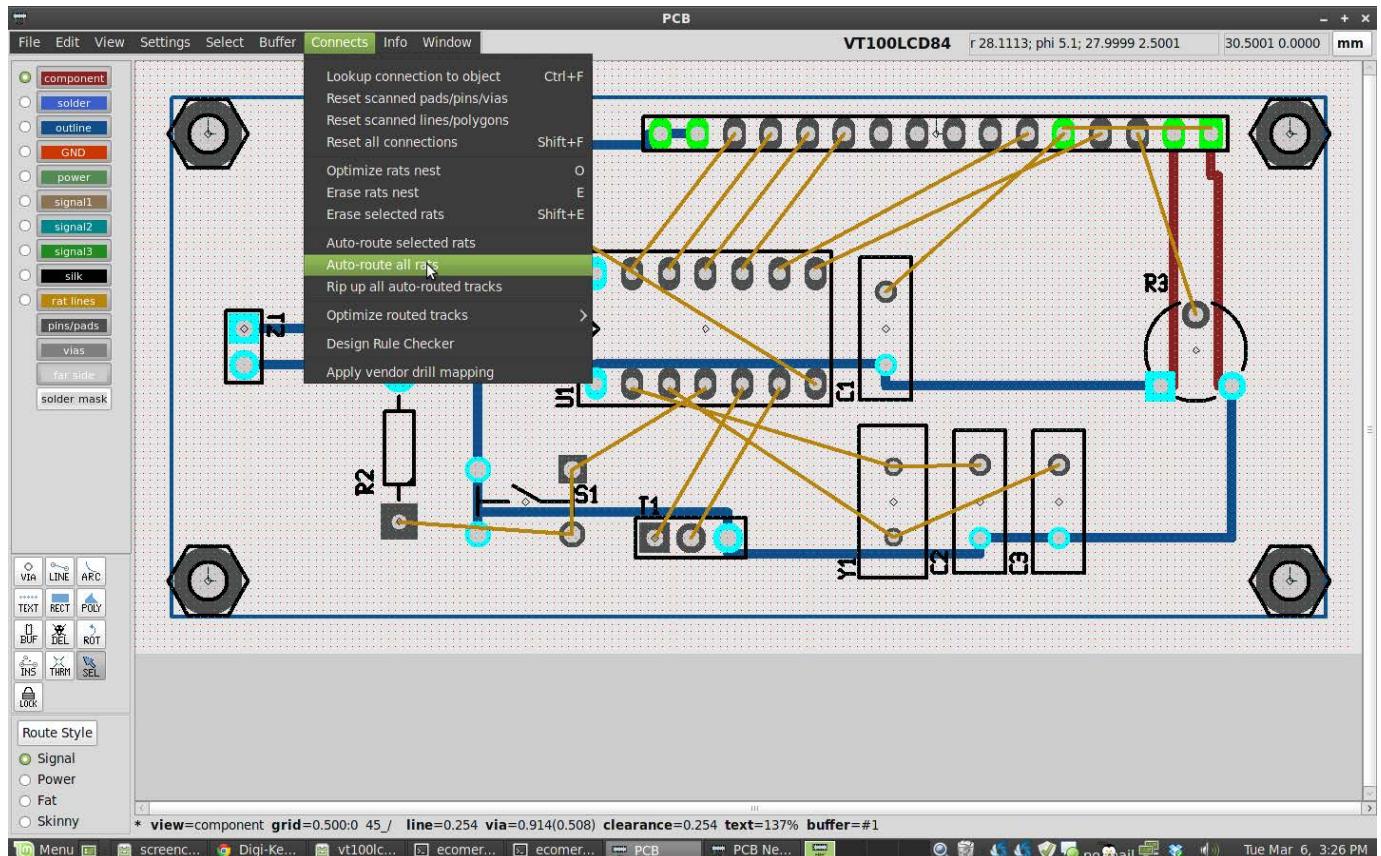


Figure 9. gEDA PCB Layout in Progress

new DIY PCB layout process is commonly called the toner transfer method, because a laser printer is involved. Thankfully, the old paste-up tape and dry-transfer component patterns are a thing of the past. Computer software is now available for the DIY community that takes software-designed schematics as input and produces a representative PCB layout (Figure 7 was produced by such software).

As I mentioned earlier, a number of PCB design software programs exist,

but I'm focusing on the open-source PCB program of the gEDA project by geda-project.org here. An example of an in-progress gEDA PCB layout is shown in Figure 9, and its final output is a positive similar to Figure 7.

The positive of Figure 7 needs to be printed in reverse onto a paper that easily will release the toner when heated. Laser printer toner is a finely ground polymer plastic that is fused to the paper by heat. The trick of the "toner method" is to get the toner to transfer from the paper to

the copper-laminated board once it is re-heated. A big part of the secret here is the type of paper you use.

Several paper solutions for the “toner method” exist, and some are better than others. Regardless of the type of paper used, the process is to place the reverse image positive laser print with the toner touching the metal surface of a clean copper-laminated board and then apply heat and pressure to loosen the toner from the paper, permitting it to transfer to and adhere to the copper-laminated board. Most DIYers use a common clothes iron as the heat source, although a laminating machine designed for identification cards is successfully used with one commercial product that I’ll talk about later.

The cheapest and simplest method is simply to use ordinary copier paper. Once heated under pressure, the toner ends up adhering to both the paper and the copper-laminated board. The paper/copper-laminated board is then soaked under water, and the waterlogged paper is rubbed off with your fingers. This method leaves a lot of paper residue embedded in the toner’s surface, this is undesirable for reasons explained later.

Many other paper types are used by various DIYers. One of the most popular

is to use a high-quality magazine page that has a smooth, glossy appearance. The gloss is caused by a white clay (kaolin) coating. Because the kaolin fills in many of the pores of the paper, the toner is less firmly bonded to the paper. Second, the kaolin dissolves in water, thus freeing the toner more readily than plain paper. This method is superior to using plain paper, but it still leaves too much paper residue embedded in the toner’s surface.

Another popular method is to print onto the glossy side of photo paper or backing paper for labels. This method is superior to either plain paper or high-quality magazine paper, because there is no paper residue embedded in the toner’s surface. However, a significant problem with this method is actually getting the toner to stick to the paper evenly. Quite often, PCB traces simply fall off the slick surface while the paper works its way through the printer. Obviously, there is a lot of variability among laser printers and glossy paper types. I don’t like variability; I like dependable repeatability.

This leads me to the paper and method I use that has dependable, predictable results. The paper is colloquially known as dextrin-coated paper. Some DIYers actually make their own by making dextrin and coat paper with it. Dextrin is simply cooked cornstarch, and the

process is easy albeit a bit labor- and time-intensive. Also, getting an even coat is a challenge. If you are interested, numerous articles and videos exist on the subject—simply google “make dextrin paper”. I, however, feel that purchasing commercial dextrin paper is worth the cost. My preferred product is made by PulsarProFX (<http://www.pcbfx.com>). The company primarily sells a kit called Fab-In-A-Box, but the entire kit isn’t really necessary. Instead, buy the refill package of Transfer Paper. Also buy the Green Toner Foil. Digi-Key sells refill kits of both. Pulsar really pushes use of a laminator but cautions that its laminator isn’t hot enough to melt the toner used in Brother laser printers. My printer is a Brother HL-2140, so I simply use a clothes iron. A word of caution here: use genuine Brother toner. After-market toner cartridges may contain fuser oil that prevents the toner from adhering to copper. After several failed boards, I figured out that the problem was my new Rosewill-brand toner cartridge. When I put in a genuine Brother cartridge, my boards were successful again.

You need the Green Toner Foil because the toner adhering to the copper-laminated board is porous, and even though you cannot see it with the naked eye, there are sufficient holes for the etchant to

penetrate the toner traces and remove metal that you do not want removed. The Green Toner Foil is ironed onto the copper-laminated board resident toner, creating a smooth, impervious surface on the top of the toner traces, resulting in superior board etches. Now, remember I said that the aforementioned transfer methods were deficient due to paper residue embedded in the toner’s surface? This is because the paper residue prevents the Green Toner Foil from making a good bond to the toner.

New on LinuxJournal.com, the White Paper Library



www.linuxjournal.com/whitepapers

How do I make my own single-sided PCBs? It's fairly simple:

1. Print a reverse image positive of the PCB pattern onto the shiny side of Pulsar dextrin transfer paper.
2. Place the transfer paper's toner side against a copper-laminated board that has been cleaned with steel wool.
3. Place a sheet of ordinary paper above the transfer paper to help prevent slippage.
4. For two minutes, apply, with a few of pounds of pressure, a common clothes iron set to the highest "cotton" setting.
5. Immerse into water, the ironed-together paper/copper-laminated board. After a couple minutes, the paper probably will float off. If it doesn't, lift it off.
6. Dry the board, and with the toner side up, lay the dull side of the Green Toner Foil against the toner and another piece of ordinary paper above that.
7. Using the same clothes iron set slightly cooler (to "wool"), iron for one minute with a few pounds of pressure.
8. Peel off the Green Toner Foil.
9. Etch the board as described below.

I make only single-sided boards. If you'd like to make a double-sided board, watch the video at <http://youtu.be/XX7IekbCNIY>. This DIYer uses HP's glossy brochure paper and seems to get pretty good results.

Etching the PCB

Having read much of what is readily available on the Web concerning DIY PCB etching, when the need arose, I decided to etch a single-sided board two different ways: first with the vinegar and salt method and second with the sponge and ferric chloride method. Some DIYers are using muriatic acid, but I have not tried that.

The vinegar and salt method works, albeit slowly. Etching my small board took two hours. The formula I used was equal parts vinegar and hydrogen peroxide and a few tablespoons of table salt. Keep adding salt until the "fizzing" continues all by itself. The liquid starts out clear but then turns an attractive shade of blue (Figure 10).

The sponge and ferric chloride method works extremely well, etching the same board in a couple

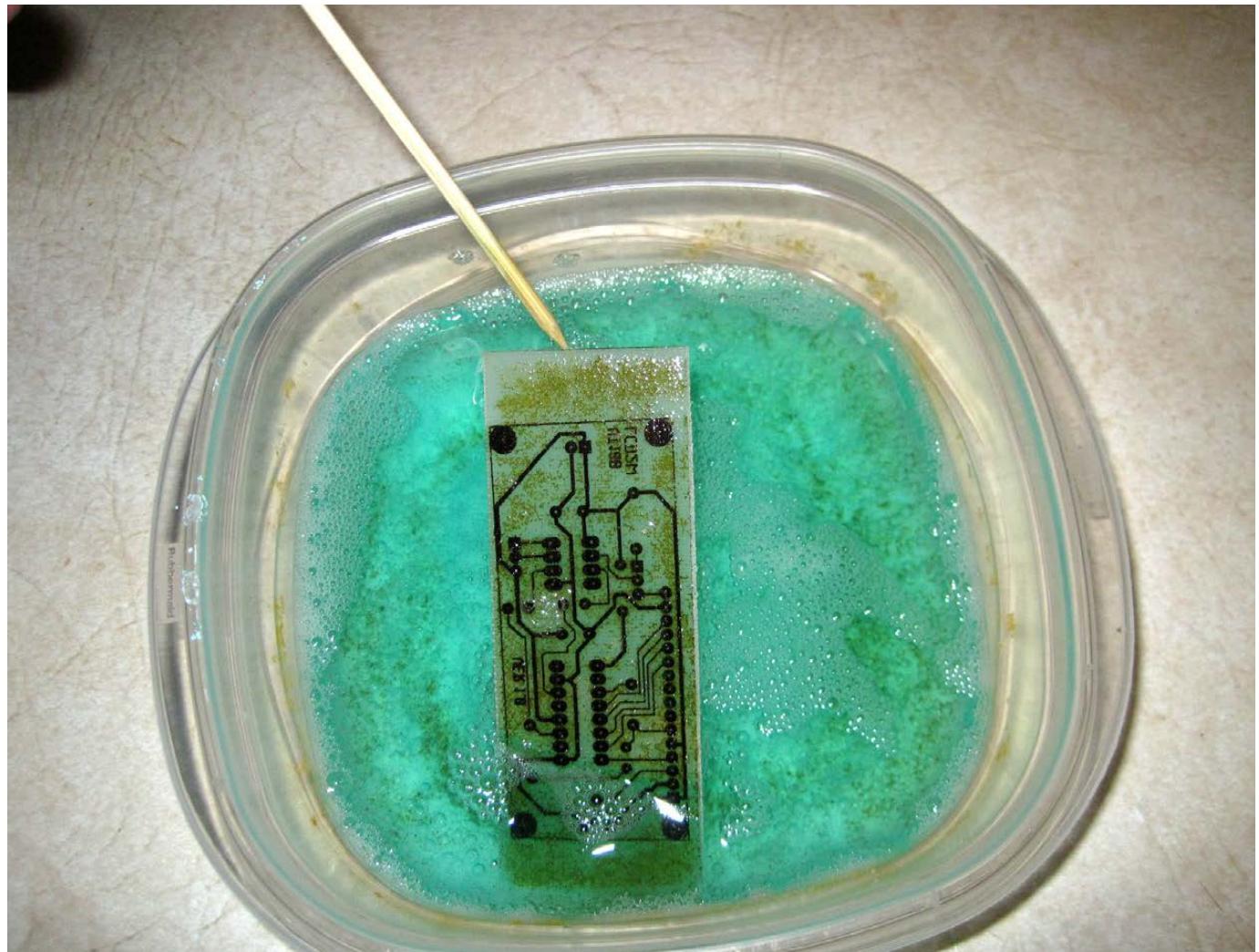
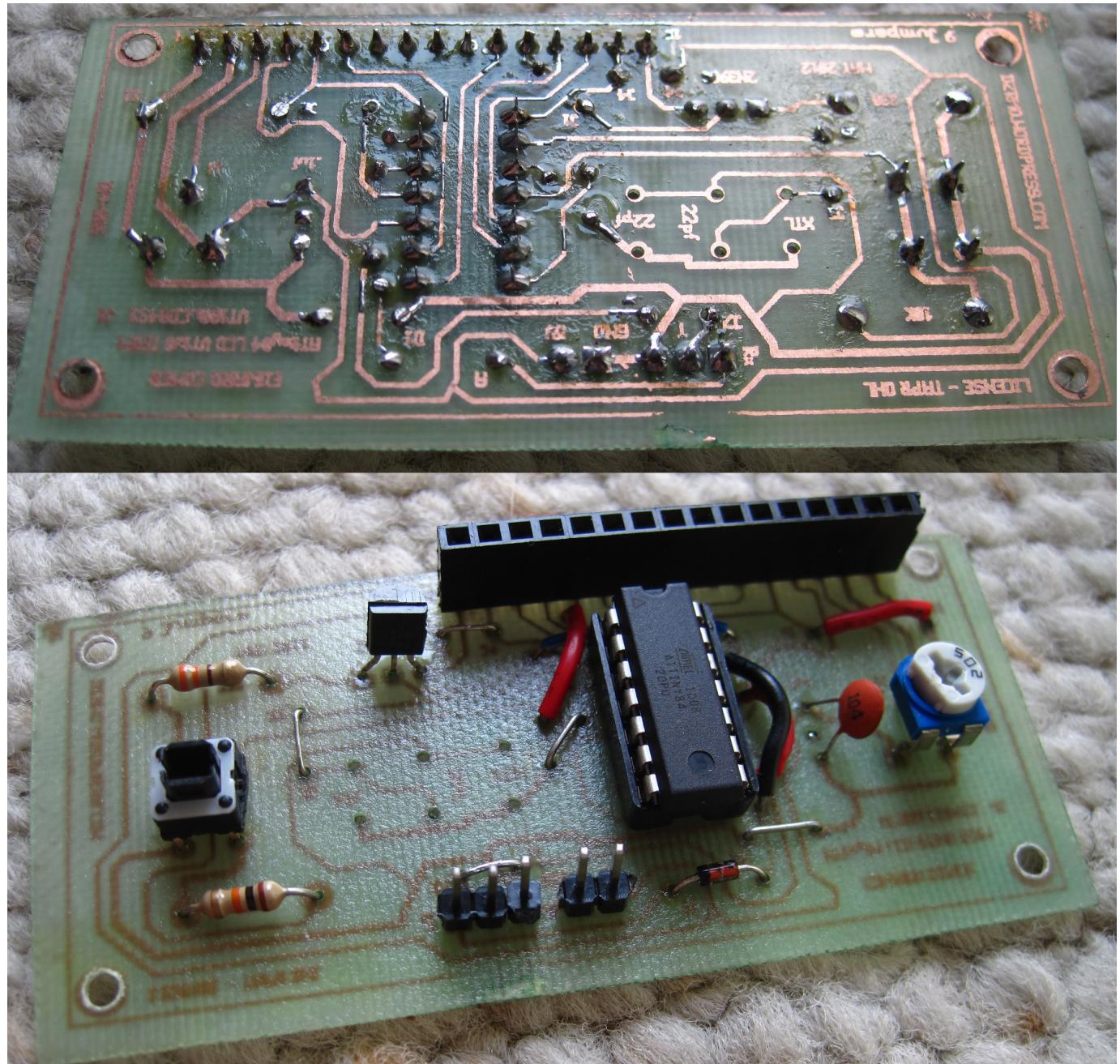


Figure 10. Vinegar and Salt Etchant

of minutes. In the past, I used ferric chloride to etch boards by placing them into a bath of ferric chloride. Even with agitation, etching a board could take ten minutes or so. The sponge and ferric chloride method accelerates the etching by continuously rubbing the surface with a sponge soaked in ferric chloride.

The rubbing removes the oxide layer that continuously builds up, permitting the ferric chloride to get to the raw metal and thus accelerate etching. Instead of a tub of etchant, a couple tablespoons is all you need, which will make a bottle of ferric chloride last for a very, very long time. The technique is simple. Don plastic

**Figure 11. Final Etched ATTiny 84 Board**

gloves, pour a couple tablespoons of ferric chloride into a small container, soak a small piece of soft sponge in the ferric chloride, then continuously

and lightly rub the saturated sponge on the PCB. In a couple minutes, the board will be finished with little mess and little ferric chloride to dispose of.

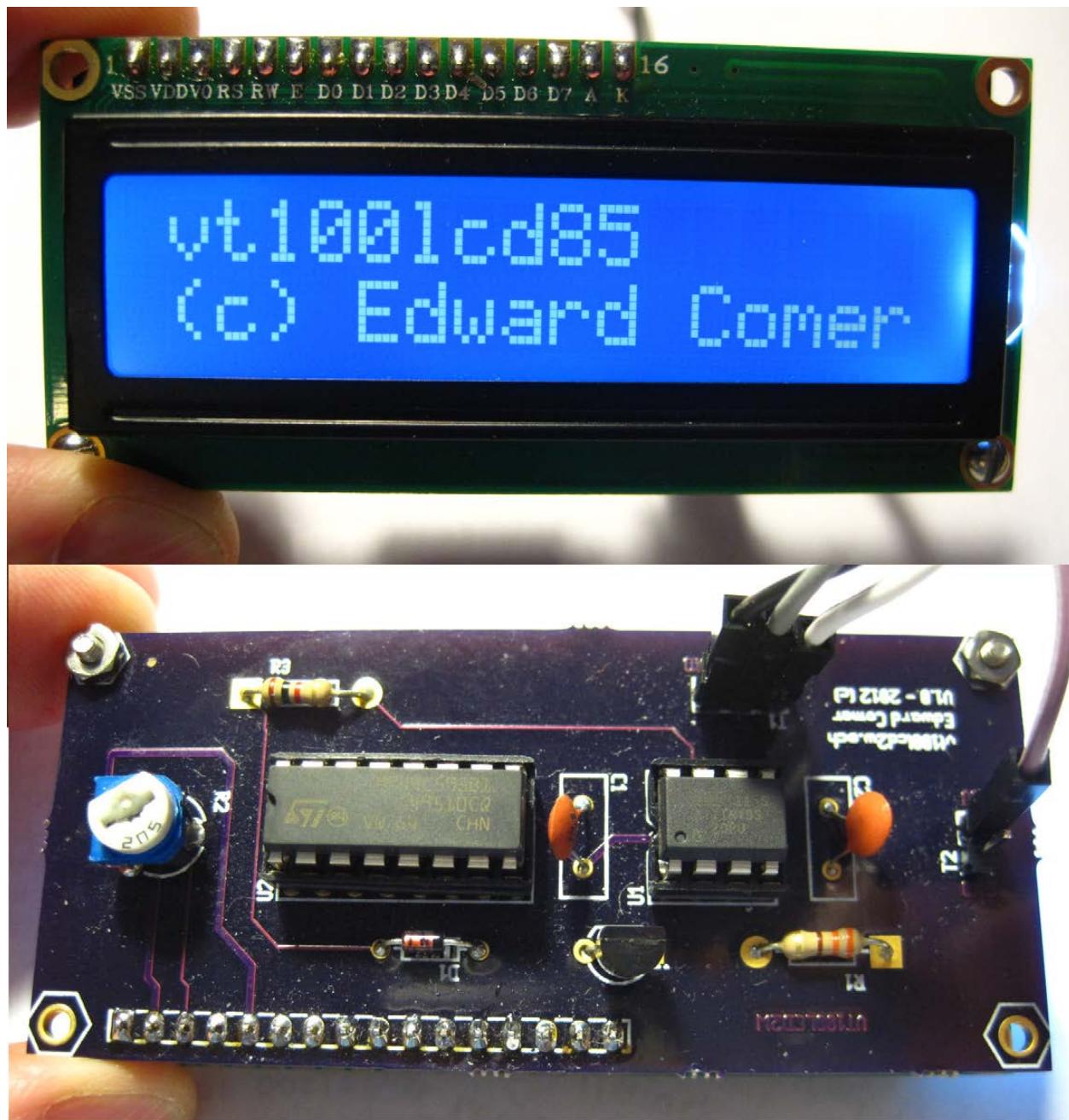


Figure 12. Commercially Made ATTiny85 Board

My final product (after three versions), a single-sided ATTiny84 version of the project, is shown in Figure 11. Given that the board

was single-sided, nine jumpers were required, which are the wires you can see on the component side of the board.



Figure 13. Commercially Made ATTiny84 Board

Commercially Made PCBs

In addition to making my own PCBs, I also had commercial boards made by a panel aggregator. A

panel aggregator is a service that aggregates boards from many sources, filling up a cost-efficient-size printed circuit board panel and then

breaking up the completed panel for delivery. Several such companies support the hobbyist community.

Figure 12 shows my Attiny85 design mounted to a 16x2 LCD.

Figure 13 shows my Attiny84 design mounted to a 16x4 LCD. ■

Edward Comer is retired from the telecommunications industry, having worked for the real AT&T, BellSouth and Numerex Corp during a 30-year career. Ed got his first UNIX login in 1975, while working for AT&T, on an assembly language version of UNIX running on a DEC PDP-11/34. From that point forward, Ed remained immersed in UNIX and later, Linux, while navigating through stints in software development, data-center management, telephone system management and product development R&D.

Resources

Source Code and Hardware Files for the vt100lcd (interested readers can pull down the files and create their own micro-terminal): <http://code.google.com>

The Arduino Project: <http://arduino.cc>

The Wiring Project: <http://wiring.org.co>

The code.google arduino-tiny Project: <http://code.google.com/p/arduino-tiny>

Thomas Fischl's USBasp Web Site: <http://www.fischl.de/usbasp>

AVRDUDE Device Programming Software: <http://www.nongnu.org/avrdude>

The Fritzing Project: <http://fritzing.org>

The gEDA PCB Development Project: <http://www.geda-project.org>

Symbol Creation: <http://embeddedtoolbox.com/mksym>

Footprint Creation by Stefan Salewski: <http://www.ssalewski.de/SFG.html.en>

“Circuit Design on Your Linux Box Using gEDA” by Stuart Brorson, *Linux Journal* November 2005: <http://www.linuxjournal.com/article/8438>

Using gEDA, by Iznogood: <http://www.linuxfocus.org/English/December2004/article355.shtml>

Getting Started with PCB: <http://www.delorie.com/pcb/docs/gs/gs.html>

gsch2pcb Tutorial: http://geda.seul.org/wiki/geda:gsch2pcb_tutorial

gschem → gsch2pcb → PCB: <http://tinyurl.com/gsched2pcb>

Circuit Simulation using gEDA and SPICE—HOWTO by Stuart Brorson: <http://www.brorson.com/gEDA/SPICE/intro.html>

The Radical Future of **NVM**

So, storing data persistently takes about a million times longer than writing to main memory. What happens when main memory is inherently persistent?

Something wonderful.

RICHARD CAMPBELL

Although 20 years of open-source software has revolutionized the way we use computers, the hardware itself has had practically no deep architectural changes. Servers, desktops and blades are still CPU, RAM, hard disks and NICs, and monitors and keyboards. Not that that's a problem—we've all enjoyed more than a thousand times better performance and capacity, lower energy use and lower prices in that time frame.

But the next orders of magnitude better performance will be arriving not in the next 20 years, but in the next couple years. And the hardware changes will not merely make our machinery faster but will usher in radically different approaches to our programming paradigms, our device interaction and the operating system itself.

After 50-some years, what has been considered a fundamental architectural barrier, RAM volatility, is about to disappear.

Right now, in every PC, mainframe, tablet and smartphone, the handful of gigabytes of SRAM and DRAM and DDRAM/2/3 are all electrically dynamic. Turn off the power, and all the program text segments, all the computed data structures, all

the transient user content, all the operating system state is lost. Any data to be referenced in the future requires persistence to "hard disk".

But the difference between RAM and disks is vast. Writes to RAM take nanoseconds; writes to disk take milliseconds—a million times slower. Disk access patterns are serial, and disk writes are most efficient only with sizable buffers, rather than RAM's random-access patterns. And, disks are notoriously less reliable.

The difference is so vast that we architect our software systems around it. We put filesystems on disks and transient data in memory. Our programming paradigm is such that running applications and system state is transient, and there's no language designed to deal with persistence intrinsically.

So, we have architected several ways to get around this: caches, buffers, flushing I/O patterns, APIs and so on, all manifestly alerting us that persistence is costly and dangerous. For reliability, writes to disk must have a variety of exception handlers. We write extra code to compress writes; we don't write complex data structures to disk; rather, we write transaction logs and so on. In traditional UNIX/Linux, we don't even write general data to disk when we want

to—it gets written back every 30 seconds or so when the OS feels like it. We write only file metadata to disk synchronously.

In terms of architectural fundamentals, think about the difference between `open()/write()/close()` and all the inherent buffering and explicit exception handling, and a statement like `int x = 10;`. We base our language and API designs on this dichotomy—RAM

their own acronyms based on their particular technology. Some use the acronym NVRAM, but I don't need to belabor the random-access aspect, do I? Presumably, generations to come will just call it "memory"—after all, memory by definition is something that is persistent.

Now, global companies like IBM, Toshiba, HP and Samsung, and startups like Everspin, Crocus and Hynix, are all

Now, global companies like IBM, Toshiba, HP and Samsung, and startups like Everspin, Crocus and Hynix, are all building and shipping NVM products, primarily used by embedded systems markets.

is fast, reliable but transient; disk writes are slow, error-prone, but persistent.

The Next Revolution

For the past decade, several technologies have been in development that provide the read and write latency and reliability of RAM with the transaction-level persistence of disks.

Here, I'll call this technology NVM for non-volatile memory. RAM, of course, means random access memory, which doesn't even mention its lack of persistence. Newer products use

building and shipping NVM products, primarily used by embedded systems markets. Industries, such as automotive, aerospace and others require very reliable persistence in small form factors. And cameras, phones, RAID controllers, network routers and other tech manufacturers are using NVM under the covers.

NVM technologies include:

- Magneto-Resistive RAM: stores bits as magnetic moments in ferromagnetic areas built in to transistor arrays. Rather

than a hard drive, this is a random-access read/write magnetic media.

- Spin-Transfer Torque RAM: stores bits as spin-aligned electrons rather than magnetically aligned atoms. Right now, this technology leads in the shortest write latencies.
- Phase-Change RAM: stores bits by changing the crystalline structure of a germanium/antimony/tellurium alloy. Fortunately, that alloy crystal can be embedded in a small transistor and can change phase in 10s of nanoseconds.
- Programmable Metallization RAM: stores bits by switching the ionization of atoms in an electrolyte between two nano-electrodes.
- Resistive RAM (memristors): stores bits by changing the conductive properties of a dielectric cell.

These new chips are all nano-fast for writes and reads without any buffers and can store data without any active power supply. But how does that persistence compare to disk drive reliability? In fact, storage to NVM chips is almost as reliable as disk storage now, and it will become more reliable in ways that disks

are unlikely to achieve. Current disks offer an endurance of about 10^{15} read/write cycles; current SRAM/DRAM can handle 10^{16} cycles—without persistence. Right now, NVM technologies offer about 10^{14} – 10^{15} , but should be able to hit 10^{16} – 10^{17} cycles and more. That's several decades of storage stability, matching and exceeding hard drives.

And NVMs should just get better over time. If you've ever seen the "Shouting in the Datacenter" video, you know just how error-prone multiplatter spinning-disks with tiny armatures frantically swinging back and forth are. Storing data in magnetic-moments or electron spins inside solid-state cells will be dramatically more reliable.

Flash in the Pan

But isn't Flash SSD a pretty good answer, available now? Yes, but this is a temporary technology, bringing some benefits now, but slated for quick obsolescence. Flash is better than a disk drive, but production Flash NAND gates are slow. Fast writes are possible only with a large RAM write cache stuck into the Flash memory cards. Worse, the gates are good for only 10^5 – 10^6 cycles—each electronic write damages the cell, so large Flash devices need additional circuitry for leveling writes across

less-used cell areas.

Because of these properties and odd buffering policies, many application use cases simply don't get any performance benefit from writing to Flash. Flash manufacturers know this—in fact, Flash is marketed as an SSD (a solid-state “drive”) telling you that it is not competing with RAM.

But this is not a bash-Flash article; I'd be just as happy with Flash-based gigabytes of persistence, as long as it is RAM-fast and reliable. Whether the

future is Flash or MR-RAM, STT-RAM, PC-RAM, PM-RAM, R-RAM or any of the other possibilities—Nanotube-RAM, SONOS, Racetrack memory and so on—the key feature is memory that is uniformly ultra-low-latency reads and writes, random-access, high-bandwidth, long-term persistent and capable of large-scale, cheap production.

(As an aside, battery-backed RAM could be just as good as NVM, if batteries didn't corrode, fail, require recharging or catch fire. Battery technology is so poor

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

today that it is simply another point of failure, not a truly safe alternative for long-term storage.)

Although current shipping quantities of NVM are merely (!) millions of units with capacities of just megabytes, all of these manufacturers are committed to continuing scale-up to GB. Some of these larger units will start shipping in 2014/2015.

This is a big industry with many creative and agile startups pushing the envelope and the leading IT manufacturers shipping their own products. The production

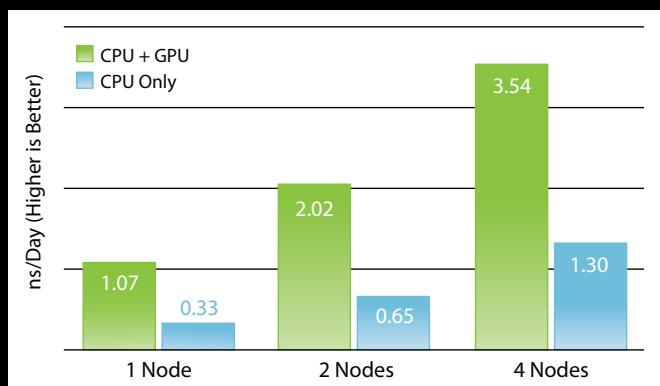
technologies are variants of existing fabrication processes. But let's move past arguing the pros and cons of the different implementation technologies. I've no idea which technology or which company will win out. But the breadth and commitment of the industry is such that the NVM future will arrive sooner than later. My motivation here is to talk about some of the implications of this hardware future and to help the software community think about what we'd like to do with it.

As mentioned, current uses of this

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



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



are mainly in the industrial sector, but it will soon be more and more visible in existing appliances, such as network routers and other communications equipment. And, we'll quickly see better hybrid-drives and RAID devices. Hybrid-drives and RAIDs, which currently use Flash or battery-backed RAM to provide a persistent cache for a traditional hard drive, will be available with the much lower latencies and higher reliabilities

8GB (and more) main-memory NVM coupled with a modern CPU?

Next Steps

The interesting changes will come with GB-sized NVM add-in cards. You can imagine a new BIOS recognizing that card and making it available as a distinct storage device to the Linux kernel. With that, it should be pretty trivial to port our standard filesystems, like ext2/3/4, xfs or

Happily, all of this means the practical elimination of filesystem maintenance, such as defragging or optimizing file placement or lengthy boot-time repair of metadata corruption.

of NVM. The benefit here is that our software architectures will be using these devices with little or no changes. That will help drive the market for more such devices, which in turn will make the manufacturing processes scale up more and more. But this is just more of the faster and better world that we expect from technology.

Our challenge is to think about Linux systems in general and a near-future world of NVM computer architecture where persistent data writes are RAM-fast. What will we do when we can have

btrfs. And they'll work fine and provide dramatic performance advantages.

But NVM is different from disks. With real random access, the algorithms for rotating disk-drive data placement are outdated. With RAM-like write latencies, we can eliminate most of the buffering and waits for error reports. And we can optimize file access made available via memory mapping to virtually eliminate any API or OS buffering.

This new "NVMfs" should likely still have a transaction journal for metadata, not to queue up writes that haven't

been written to disk, but to queue up CPU transactions and cache writes that may get interrupted. There's still a need to take care of this, but the RAM-speed of write latencies means that the kernel can do much more "filesystem" activity directly and not have to wait for interrupt callbacks. Happily, all of this means the practical elimination of filesystem maintenance, such as defragging or optimizing file placement or lengthy boot-time repair of metadata corruption.

With this, many I/O-intensive applications and databases like MySQL will see magnitudes of performance improvement. For messaging systems like ActiveMQ, no longer will there be a trade-off between unreliable and guaranteed messaging. An application using NVM-optimized SQLite will be awesome. And distributed memory caches like Memcached won't have to skimp on persistence features.

Meanwhile, encryption and compression still can be an important feature of an NVMfs. Unfortunately, for hard drives, compress/encryption comes with little cost because of the mismatch between fast CPU/RAM speed and slow disk write latencies. With an NVMfs, the performance difference between file data stored plain versus encrypted will be obvious, though still faster than those

stored on hard disks.

In the end, NVM-based filesystems probably will mean that all notebooks, desktop PCs and commodity servers will be totally solid-state systems by the end of the decade. We then can optimize our OSes to write-back memory pages to networked disk storage, not for persistence, but for distributed access and disaster recovery. Hard drives, which still will have an orders of magnitude advantage in total storage capacity for years to come, will be relegated to the data center and cloud where they can be cared for properly.

Radical Changes

But if we can store filesystem data in NVM, we can store application data there too. One simple model could be for applications to ask to map in NVM memory, as is done now with mmap'd files. Of course, NVM memory regions need no backing disk store—they are inherently resilient. Many performant applications ignore persistence functionality, not even using a transaction log. NVM means that all sorts of applications can have persistent semantics, being able to use complex data structures in their programmatic idioms, without even a `foo.save()` required.

How should these NVM areas be named? How should they be secured?

Whereas current programming paradigms assume that variables are transient ipso facto, future languages or extensions may allow different semantics for different data. Rather than naming data regions and using some sort of `brk()` or `mmap()` call, certain language keywords or data-naming rules would enable automatic re-mapping to persisted NVM data. (Ironically, the “volatile” keyword in C/C++ may be required for transient data!)

For decades, virtual machine-based languages, such as Smalltalk and various Lisps, have had to have cumbersome “save world” commands to write out all the in-memory data structures and class or function definitions to disk. In an NVM world, we don’t need a separate command for this—all VM use of NVM is persistent. A virtual machine world will be both dynamic, fast and long-lived. Modern VM languages like Java and also dynamic scripting languages like Ruby/Python/Tcl could enable an application or system to store all of its active data structures in NVM with no need for laborious serialization on and off slow disks.

Perhaps functional languages, such as Erlang and Haskell, with their immutable value design, could take the best advantage of NVM. Their clean,

mathematical philosophy has never much liked the “side effect” of storage. Now they may be able to support persistence as a virtually free feature.

NVM on ACID

Of course, with automatic and pervasive persistence, comes the problem of transactional support. Although the data may be persistent, there is no guarantee that what data you’re reading was all written out as a single, correct transaction. Take setting some array, map or string to some value. If some number of (persistent) stores or cache writes are interrupted, the array/map/string may be only half set correctly. Naïvely getting reconnected to that data segment won’t tell you what portion of the data is correct. To help with this, applications could ask the kernel to set restore points for their data. A more-sophisticated solution would be for VMs to provide software transaction semantics or for NVM hardware transactional memory to ensure atomicity and consistency.

As far as ACID (atomicity, consistency, isolation and durability) goes, NVM could eliminate worrying about “D”. But will this NVM data be truly resilient? Won’t bullet-proof resilience as needed for financial transactions still exact enormous penalties? Probably not. If NVM itself isn’t good enough, redundant boards

Software Freedom Day

Saturday 15th September 2012
www.softwarefreedomday.org



Let's spread the word of Free / Open Source Software and celebrate SFD together!

What is Software Freedom Day?

Software Freedom Day (SFD) is a worldwide celebration of Free and Open Source Software (FOSS). Our goal in this celebration is to educate the worldwide public about of the benefits of using high quality FOSS in education, in government, at home, and in business -- in short, everywhere! The non-profit organization Software Freedom International coordinates SFD at a global level, providing support, giveaways and a point of collaboration, but volunteer teams around the world organize the local SFD events to impact their own communities.

Sponsors and Partners



FreeBSD



should provide the needed availability. Better yet, HA systems with primary and secondary NVM still will work thousands of times faster than current HA disk-based systems.

Linux

Now that you're getting used to the idea of "RAM" being "NV", let's go all the way down the stack to the operating system itself. What advantage could the Linux kernel take of NVM? It's not just that a Linux NVM system could boot in fractions of a second, but that having some (or most?) kernel state persisted at practically no extra cost in time opens up many interesting possibilities. The bootloader still can execute hardware power-on self-tests, but there's very little extra work required to get the kernel running when much of the kernel state and instruction space is magically still available.

During a transition period, when DRAM and NVM coexist in a system, the kernel process table could be modified to note which processes are running wholly in NVM. On reboot, the kernel process table (also in NVM) could ignore DRAM-based processes, while letting NVM processes get going as soon as system devices are initialized. And, as mentioned, the kernel could help with application data restore points.

An NVM kernel also may help with managing devices. The ability of

RAM-fast data persistence could enable the kernel to remember the state of attached hardware to a far greater ability (or, to be fair, the devices may have their own NVM to help out).

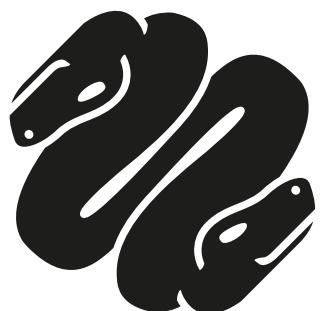
The Future

NVM is coming. Without much work, it will provide an enormous benefit to applications and use cases where storage performance is a limiting factor. I've tried to outline some of the more revolutionary ways that we can take advantage of the technology. As RAM volatility has been a fundamental assumption of our computing architecture, it is hard to figure out what an NVM future could look like. What may have been the design principles, kernel semantics and language design in an alternate computing world where NVM was invented in, say, the 1960s? More radical notions could work in theory, but there may be no easy migration path from where we are today. It will be up to the global community to figure out answers with open source and Linux driving the way. ■

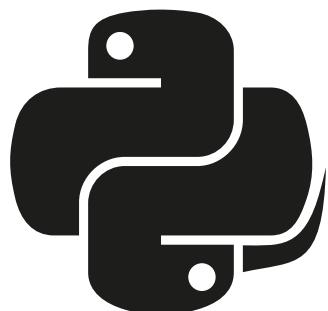
Richard Campbell is a trading systems architect living in New Jersey and the author of *Managing AFS: The Andrew File System*. His first computer had a 12KHz Z-80 CPU with 256 bytes of ROM, 10KB of RAM, and used 1100 baud cassette tapes for storage. Send comments to nvm@netrc.com. See <http://www.netrc.com/nvm> for links and more information.

nz.pycon.org

Registrations
and
Talk Submissions
now open!



KIWI PYCON
2012
DUNEDIN
1st-2nd Sept.
New Zealand





DOC SEARLS

Making the Case to Muggles

Explaining Linux isn't enough.

Most people experience Linux the way they experience a light switch or a water faucet. When they use it, they expect it to work and give them what they want. And if it doesn't work, they expect an expert to come and make it work. In their experience, Linux is the business end of infrastructure: the road, not the rubber that meets it.

But the difference between Linux and water, electricity or a road is that most people know what those other things are—and they don't know what Linux is, even when they've heard of it. That's why we need metaphors like the above if we're going to explain Linux to them.

But do we really need to explain Linux to people who don't know or care much about it? And if so, why?

For most of Linux's history, those of us close to the topic believed Linux mattered enough to deserve understanding by

others, especially since we were certain that Linux would some day achieve what we liked to call World Domination. Linux has crossed that threshold, but not by the crowning victory we had hoped for from the start: running on many millions of personal computing and communication devices and getting full credit for that, by name. Today, the only form of Linux doing that is Android, which is "Linux-based", rather than Linux itself.

Today, as I write this, news comes that \$100 million has been invested in GitHub, the "social coding" site that currently hosts millions of code repositories for millions of people, all using the Git distributed revision control system created by Linus Torvalds. There are excited stories about GitHub in the *Wall Street Journal*, *Forbes*, *Reuters*, *TechCrunch*, the *San Francisco Chronicle*, the *Washington Post*, *Red Herring*, *GigaOM* and dozens of other mainstream

pubs that don't mention Linus at all. I was about to stop looking when I finally found one: Rafe Needleman, writing in CNET ([http://news.cnet.com/8301-1023_3-57468899-93/github-raises-\\$100-million-from-andreessen-horowitz](http://news.cnet.com/8301-1023_3-57468899-93/github-raises-$100-million-from-andreessen-horowitz)), credits Linus right up front. Still, nobody mentions Junio Hamano, who has been maintaining Git since Linus handed that duty to him July 2005. At the time of this writing, Junio's entry in Wikipedia is a three-line stub.

How many \$billions have been made because of Linus' founding work? How many more will be made thanks to Linus' and Junio's work on Git?

A better question: would Linux and Git have succeeded so spectacularly if Linus had tried to own either of them? No. As Harry Truman said, "It is amazing what you can accomplish if you do not care who gets the credit."

Linux, Git and countless other code bases are working as infrastructural building materials today because their creators made them free in the first place. That's what matters—not who gets the credit. The real problem we have today is that freedoms embodied in code are barely understood or credited at all. The same goes for free hardware. And, for lack of that understanding, we are losing those freedoms today.

Eben Moglen made this fact clear

in a speech titled "Innovation under Austerity", which he gave at the Freedom to Connect conference in Silver Spring, Maryland, in May of this year. I joined Eben on stage for a conversation after that speech, and opened by saying it was not only one of the best speeches I'd ever heard, but one of the most important. See the video (<http://boingboing.net/2012/05/27/innovation-under-austerity-eb.html>) in an excellent posting in BoingBong by Cory Doctorow. The Software Freedom Law Center has a full transcript as well (http://softwarefreedom.org/events/2012/freedom-to-connect_moglen-keynote-2012.html). Here's a compressed excerpt:

For the policy makers, in other words, an overwhelming problem is now at hand: how do we have innovation and economic growth under austerity? They do not know the answer to this question, and it is becoming so urgent that it is beginning to deteriorate their political control.

Nobody will ever try to create a commercial encyclopedia again.

Disintermediation, the movement of power out of the middle of

the Net is a crucial fact about 21st century political economy. It proves itself all the time. Somebody's going to win a Nobel Prize in Economics for describing, in formal terms, the nature of disintermediation.

The greatest technological innovation of the late 20th century is the thing we now call the World Wide Web, an invention less than 8,000 days old. That invention is already transforming human society more rapidly than anything since the adoption of writing.

What do we know about how to achieve innovation under austerity? We created the Cloud. We created the idea that we could share operating systems and all the rest of the commoditizable stack on top of them. We did this using the curiosity of young people, not venture capital. Venture capital came towards us not because innovation needed to happen, but because innovation had already happened.

That curiosity of young people could be harnessed because all of

the computing devices in ordinary day-to-day use were hackable, and so young people could actually hack on what everybody used. That made it possible for innovation to occur where it can occur without friction, which is at the bottom of the pyramid of capital. Hundreds of thousands of young people around the world hacking on laptops, hacking on servers, hacking on general-purpose hardware available to allow them to scratch their individual itches—technical, career, and just plain ludic itches (“I wanna do this; it would be neat”)—which is the primary source of the innovation which drove all of the world’s great economic expansion in the past ten years. The way innovation really happens is that you provide young people with opportunities to create on an infrastructure which allows them to hack the real world and share the results.

That’s the upside. The downside is this:

All of that innovation comes from the simple process of letting the kids play and getting out of the way. Which, as you are aware,

we are working as hard as we can to prevent, now, completely. Increasingly, around the world, the actual computing artifacts of daily life for individual human beings are being locked so you can't hack them. The individual computing laboratory in every 12-year-old's pocket is being locked down. If you prevent people from hacking on what they own themselves, you will destroy the engine of innovation from which everybody is profiting. The goal of the network operators is to attach every young human being to a proprietary network platform with closed terminal equipment that she can't learn from, can't study, can't understand, can't whet her teeth on, can't do anything with except send text messages that cost a million times more than they ought to.

This paragraph replaces a long digressive harangue I spent two days writing. I visited patents and copyrights, ACTA and SOPA (both of which *Linux Journal* readers by now know a great deal), and a new issue: the Trans-Pacific Partnership Agreement (TPP, <https://www.eff.org/issues/tpp>), by

which the US is quietly working to muscle New Zealand (<http://internetnz.net.nz/our-work/Openness/Trans-Pacific-Partnership-TPP-agreement>) and other countries into matching the US's Hollywood-driven and freedom-hostile intellectual property laws. So I urge you to pay attention to that one, while here we look instead at the freedom found in general-purpose computing. This is perhaps the most important issue, and also the hardest one to explain.

General-purpose computing was born out of IBM's original PC, which arrived in 1982. That machine itself was not free and open, but its BIOS could be reverse-engineered, which Phoenix Technologies did in 1993, making possible the manufacture of "IBM-compatible" PCs, better known at the time as clones, by anybody. Succeeding generations of PCs mostly ran Microsoft's operating systems. But they didn't need to. That was what made Linux and countless other operating systems possible. *General-purpose computers don't depend on any one company's controlling technology.*

General-purpose communications are the same. We aren't locked in to anybody or anything. This is the miracle of the Internet. We don't need a phone company to make the connection for us. We don't need a license to use it. We have a choice of many services and

paths. We have open protocols for file transfer, for e-mail and for much else. For specialized communications, such as that provided by Skype, there are many choices, and opportunities for many more. But none excludes any other.

The latest threat to general-purpose computing is UEFI, the Unified Extensible Firmware Interface. Intended as a security measure, it adds a layer of complication to running an operating system other than preinstalled Windows on otherwise generic PC hardware. To make installs easy, Fedora has elected to pay what Cory Doctorow calls “blood money” to make booting a non-hassle (<http://boingboing.net/2012/07/06/zareason-a-computer-company-w.html#more-169692>). (Go to Implementing UEFI Secure Boot in Fedora for the details at <http://mjg59.dreamwidth.org/12368.html>. It won’t give you warm fuzzies.) The direction this development points is toward less general purposefulness. And this isn’t good.

One of the best characterizations of the Internet I’ve ever heard was “a way, not a place”, which was the title and key point of a speech Phil Windley gave at a conference earlier this year. (He makes the same point in this post: http://www.windley.com/archives/2012/03/ways_not_places.shtml.)

A protocol is a way. And thus, so is the Internet. We may talk about spaces, domains, locations, sites and addresses, all of which frame the Net as real estate. But TCP/IP is a way, not a place. All it does is make a best effort to connect any two end points by any means possible. Its purpose could not be more general.

Back in 1997, a hacker (presumably with the (very pre-Twitter) handle @Man put up a page titled “Attention, Fat Corporate Bastards!” It lasted until 2010, but can still be found in the Internet Archive (<http://web.archive.org/web/19970607134127/http://www.ecst.csuchico.edu/~atman/attention-fat-bastards.html>). After yelling about freedom for most of the page, he treats the reader to a passage that rings as true today as it did 14 years ago:

You almost certainly think of the Internet as an audience of some type—perhaps somewhat captive. If you actually had even the faintest glimmering of what reality on the net is like, you’d realize that the real unit of currency isn’t dollars, data, or digicash. It’s reputation and respect. Think about how that impacts your corporate strategy. Think about how you’d feel if a guy sat down at your lunch table

Advertiser Index

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

ADVERTISER	URL	PAGE #
1&1	http://www.1and1.com	31
EMAC, INC.	http://www.emacinc.com	37
EMPEROR LINUX	http://www.emperorlinux.com	57
HPC ON WALL STREET	http://www.flaggmgmt.com/hpc/	49
iXSYSTEMS	http://www.ixsystems.com	7
KIWI PYCON	http://nz.pycon.org	117
LULLABOT	http://www.lullabot.com	47
MICROWAY	http://www.microway.com	110, 111
OHIO LF	http://ohiolinux.org/	83
SILICON MECHANICS	http://www.siliconmechanics.com	3
SOFTWARE FREEDOM DAY	http://www.softwarefreedomday.org	115
USENIX LISA	http://www.usenix.org/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>.

one afternoon when you were interviewing an applicant for a vice-president's position and tried to sell the two of you a car, and wouldn't go away. Believe it or not, what you want to do with the Internet is very similar. Just as you have a reasonable expectation of privacy and respect when you're at a table for two in a public place, so too do the users of the Internet have a reasonable expectation of privacy and respect. When you think of the Internet, don't think of Mack trucks full of widgets destined for distributorships, whizzing by countless billboards. Think of a table for two.

What could be more general-purpose than a table? Or easier to explain?

Computers are complicated when you look inside them. So are communications. But their purposes are general, which makes them simple. Nobody needs a license to build or operate a table. In explaining freedom, maybe it's best to start there. For everybody's sake. ■

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.