Pry | TCP | PirateBox | 3-D Printing | OpenLDAP | Webmin

LINUX JOURNAL™

Since 1994: The Original Magazine of the Linux Community

A LOOK AT SOFTWARE FOR 3-D PRINTERS

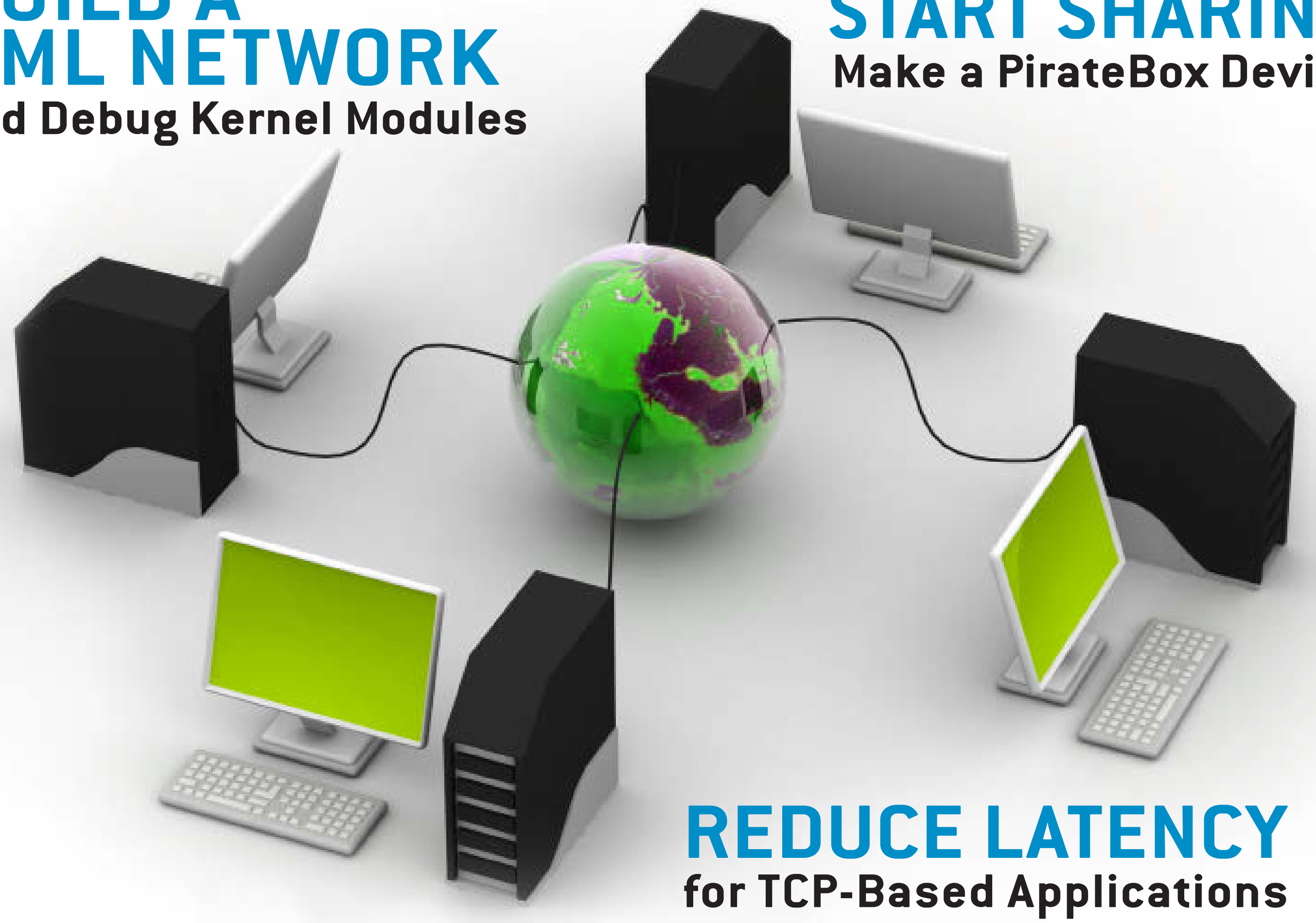
WHAT'S YOUR DATA WORTH?

JULY 2012 | ISSUE 219 | www.linuxjournal.com

NETWORKING

BUILD A UML NETWORK
and Debug Kernel Modules

START SHARING
Make a PirateBox Device



REDUCE LATENCY
for TCP-Based Applications

PRY: A MODERN REPLACEMENT FOR RUBY'S IRB

ENGINEER AN OpenLDAP DIRECTORY

USE WEBMIN TO MANAGE YOUR LINUX SERVER

CONTENTS

JULY 2012
ISSUE 219

NETWORKING

FEATURES

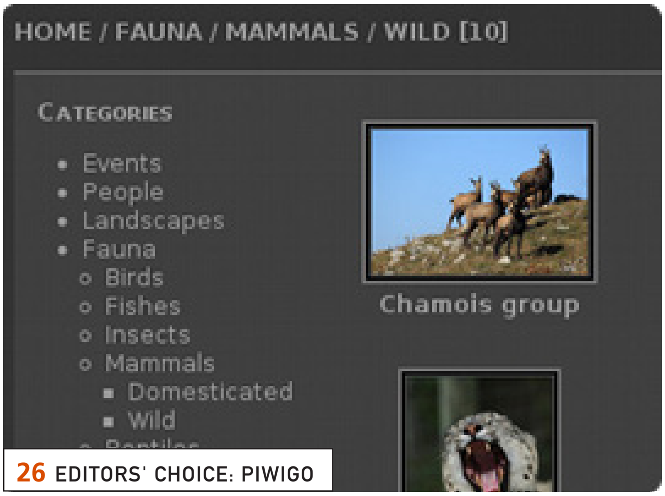
ON THE COVER

Cover Image: © Can Stock Photo Inc. / rbhavana

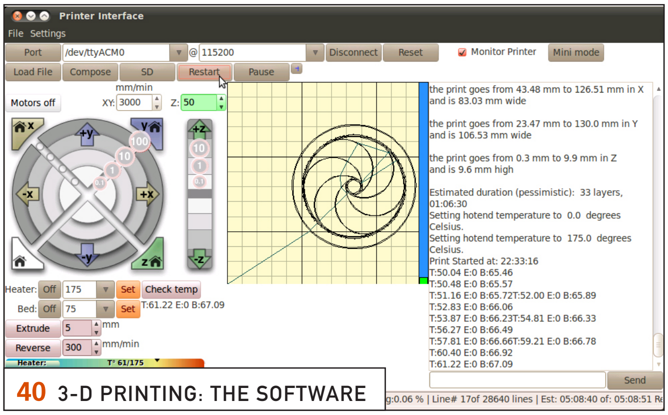
COLUMNS

INDEPTH

IN EVERY ISSUE



26 EDITORS' CHOICE: PIWIGO

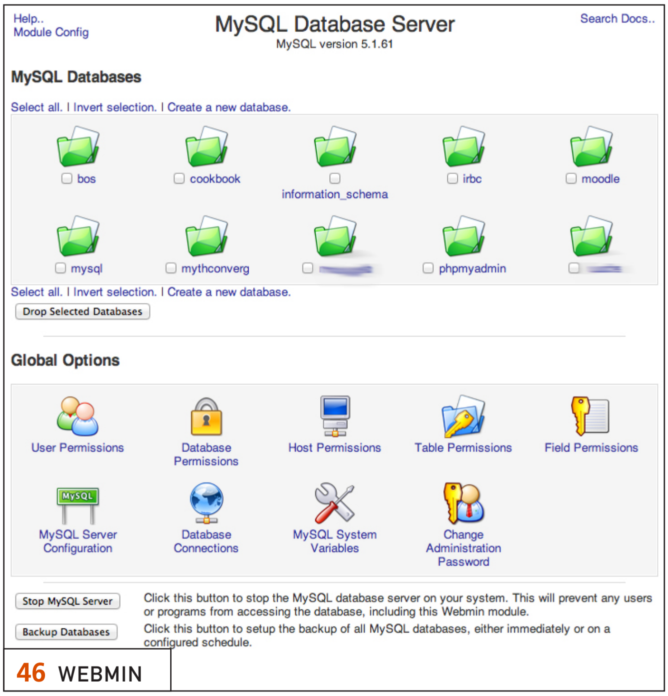


```
T:61.22 E:0 B:67.09
```

```
the print goes from 43.48 mm to 126.51 mm in X
and is 83.03 mm wide

the print goes from 23.47 mm to 130.0 mm in Y
and is 106.53 mm wide

the print goes from 0.3 mm to 9.9 mm in Z
and is 9.6 mm high

Estimated duration (pessimistic):  33 layers,
01:06:30
Setting hotend temperature to  0.0  degrees
Celsius.
Setting hotend temperature to  175.0  degrees
Celsius.
Print Started at: 22:33:16
T:50.04 E:0 B:65.46
T:50.48 E:0 B:65.57
T:51.16 E:0 B:65.72T:52.00 E:0 B:65.89
T:52.83 E:0 B:66.06
T:53.87 E:0 B:66.23T:54.81 E:0 B:66.33
T:56.27 E:0 B:66.49
T:57.81 E:0 B:66.66T:59.21 E:0 B:66.78
T:60.40 E:0 B:66.92
T:61.22 E:0 B:67.09
```

g:0.06 % | Line# 17of 28640 lines | Est: 05:08:40 of: 05:08:51 Re

40 3-D PRINTING: THE SOFTWARE



Click this button to stop the MySQL database server on your system. This will prevent any users or programs from accessing the database, including this Webmin module.

Click this button to setup the backup of all MySQL databases, either immediately or on a configured schedule.

46 WEBMIN

LINUX JOURNAL (ISSN 1075-3583) is published monthly by Belltown Media, Inc., 2121 Sage Road, Ste. 310, Houston, TX 77056 USA. Subscription rate is $29.50/year. Subscriptions start with the next issue.

LINUX JOURNAL

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

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

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

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

iXsystems Servers + Intel® Xeon® Processor E5-2600 Family =

Unparalleled performance density

E5-2600

HIGH & Throughput INCREDIBLE Performance Density

iXsystems is pleased to present a range of new, **blazingly fast servers** based on the Intel® Xeon® Processor E5-2600 family and the Intel® C600 series chipset.

The Intel® Xeon® Processor E5-2600 Family employs a new microarchitecture to boost performance by up to 80% over previous-generation processors. The performance boost is the result of a combination of technologies, including Intel® Integrated I/O, Intel® Data Direct I/O Technology, and Intel® Turbo Boost Technology.

The iXR-1204+10G features Dual Intel® Xeon® E5-2600 Family Processors, and packs up to 16 processing cores, 768GB of RAM, and dual onboard 10GigE NICs in a single unit of rack space. The robust feature set of the iXR-1204+10G makes it suitable for clustering, high-traffic webservers, virtualization, and cloud computing applications.

IXR-1204+10G-10GbE On-Board

For computation and throughput-intensive applications, iXsystems now offers the iXR-22x4IB. The iXR-22x4IB features four nodes in 2U of rack space, each with dual Intel® Xeon® E5-2600 Family Processors, up to 256GB of RAM, and a Mellanox® ConnectX QDR 40Gbp/s Infiniband w/QSFP Connector. The iXR-22x4IB is perfect for high-powered computing, virtualization, or business intelligence applications that require the computing power of the Intel® Xeon® Processor E5-2600 family and the high throughput of Infiniband.

IXR-22x4IB

iXR-1204+10G

- Dual Intel® Xeon® E5-2600 Family Processors
- Intel® X540 Dual-Port 10 Gigabit Ethernet Controllers
- Up to 16 Cores and 32 process threads
- Up to 768GB Main Memory
- 700W Redundant high-efficiency power supply

iXR-22x4IB

- Dual Intel® Xeon® E5-2600 Family Processors per node
- Mellanox® ConnectX QDR 40Gbp/s Infiniband w/QSFP Connector per node
- Four server nodes in 2U of rack space
- Up to 256GB Main Memory per server node
- Shared 1620W Redundant high-efficiency Platinum level (91%+) power supply

iXsystems®

intel® inside™ Xeon®

Intel, the Intel logo, and Xeon Inside are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

Call iXsystems toll free or visit our website today! **1-855-GREP-4-IX | www.iXsystems.com**

SHAWN POWERS

Cast the Nets!

I thought we'd gone native this month and were going to show how to work nets and fish like the penguins do. I had a double-fisted, sheep-shanked, overhand cinch loop to teach you, along with the proper way to work your net in a snow storm. As it turns out though, it's actually the "networking" issue. That's still pretty cool, but instead of the half hitch, you get a crossover cable, and instead of my constrictor knot, you get load balancing.

Reuven M. Lerner starts out the issue with an article on Pry. If you're a Python programmer using iPython, you'll want to check out its Ruby counterpart, Pry. Although it's not required for coding with Ruby, it makes life a lot easier, and Reuven explains why. With a similar goal of improving your programming skills, Dave Taylor shows how to use subshells in your scripting. This doesn't mean you can't continue to write fun scripts like Dave's been demonstrating the past few months, it just means Dave is showing you how to be more efficient scripters. His tutorial is a must-read.

I got into the networking theme myself this month with a column on Webmin. Some people consider managing a server with Webmin to be a crutch, but I see it as a wonderful way to learn system administration. It also can save you some serious time by abstracting the underlying nuances of your various server types. Besides, managing your entire server via a Web browser is pretty cool. Speaking of "pretty cool", Kyle Rankin finishes his series on 3-D printing this issue. The printer itself is only half the story, and Kyle explains all the software choices for running it.

If Webmin seems a little light for your networking desires, perhaps Ratheesh Kannoth's article on the reconnaissance of the Linux network stack is more up your alley. Ratheesh peels back the mystery behind what makes Linux such a powerful and secure kernel, and does it using UML. If that sounds confusing, don't worry; he walks you through the entire process.

If you're actually creating or tweaking a network application, Andreas Petlund's article on TCP thin-stream modifications will prove invaluable. Anyone who ever has been fragged by an 11-year-old due to network latency knows a few milliseconds can be critical. Certainly there are other applications that rely on low network

Anyone who ever has been fragged by an 11-year-old due to network latency knows a few milliseconds can be critical.

latency, but few are as pride-damaging as that. Andreas shows how to tweak some settings in the kernel that might make the difference between fragging or getting fragged. Unfortunately, no amount of tweaking can compare with the fast reflexes of an 11-year-old—for that you're on your own.

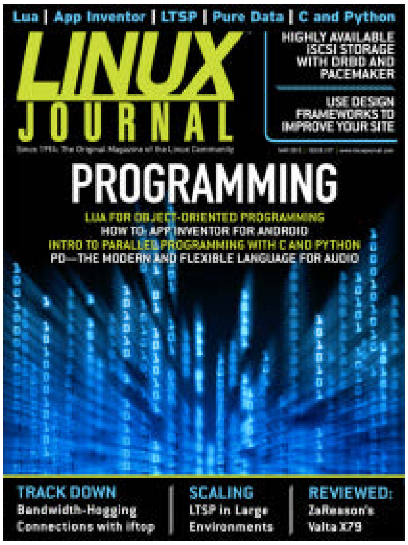
Stewart Walters picks up his OpenLDAP series from the April issue, and he demonstrates how to manage replication in a heterogeneous authentication environment. OpenLDAP is extremely versatile, but it still runs on hardware. If that hardware fails, a replicated server can make a nightmare into a minor inconvenience. You won't want to skip this article.

If my initial talk of fishing nets, knots and the high seas got you excited, fear not. Although this issue isn't dedicated to fish-net-working, my friend Adrian Hannah introduces the PirateBox. If the Internet is too commonplace for you, and you're more interested in dead drops, secret Wi-Fi and hidden treasure, Adrian's article is for you. The PirateBox doesn't track users, won't spy on your family and won't steal your dog. What it will do is share its digital contents to anyone in range. If your interest is piqued, check out Adrian's article and build your own. Yar!

This issue focuses on networking, but like every month, we try hard to include a variety of topics. Whether you're interested in Doc Searls' article on personal data or want to read new product and book announcements, we've got it. If you want to compare your home network setup with other *Linux Journal* readers, check out our networking poll. Perhaps you're in the market for a cool new application for your Linux desktop. Be sure to check out our Editors' Choice award for the app we especially like this month. Cast out your nets and reel in another issue of *Linux Journal*. We hope you enjoy reading it as much as we enjoyed putting it together. ■

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


Lua | App Inventor | LTSP | Pure Data | C and Python
LINUX JOURNAL
HIGHLY AVAILABLE ISCSI STORAGE WITH DRBD AND PACEMAKER
USE DESIGN FRAMEWORKS TO IMPROVE YOUR SITE
PROGRAMMING
LUA FOR OBJECT-ORIENTED PROGRAMMING
HOW TO: APP INVENTOR FOR ANDROID
INTRO TO PARALLEL PROGRAMMING WITH C AND PYTHON
PD—THE MODERN AND FLEXIBLE LANGUAGE FOR AUDIO
TRACK DOWN Bandwidth-Hogging Connections with iftop
SCALING LTSP in Large Environments
REVIEWED: ZaReason's Valta X79

Clarifications

In Florian Haas' article "Replicate Everything! Highly Available iSCSI Storage with DRBD and Pacemaker" (in the May 2012 issue of *LJ*), we noticed some information that has loose factual bearing upon the conclusions that are stated and wanted to offer our assistance as the developers of the software.

When reading the article, we felt it misrepresented information in a way that could be easily misinterpreted. We have listed a few sentences from the article with an explanation and suggested corrections below.

1) Statement: "That situation has caused interesting disparities regarding the state of vendor support for DRBD."

Clarification: we would like to mention that DRBD is proudly supported by Red Hat and SUSE Linux via relationships with DRBD developer LINBIT.

Correction: DRBD is widely supported by enterprise software vendors and also free open-source operating system developers. It comes prepackaged in Debian, Ubuntu, CentOS, Gentoo and is available for download directly from LINBIT. Red Hat and SUSE officially accept DRBD as an enterprise solution, and its customers benefit from having a direct path for support.

2) Statement: "Since then, the 'official' DRBD codebase and the Linux kernel have again diverged, with the most recent DRBD releases remaining unmerged into the mainline kernel. A re-integration of the two code branches is currently, somewhat conspicuously, absent from Linux kernel mailing-list discussions."

Clarification: this is simply FUD and not true. DRBD 8.3.11 is included in the mainline kernel. DRBD 8.4 (which has pending feature enhancements) is not included in the mainline kernel until testing is complete and features are brought to stable. This does not mean code is diverged or unsupported; it simply means "alpha" and "beta" features aren't going to find their way into the Linux mainline. This is standard operating practice for kernel modules like DRBD.

Correction: Since then, DRBD has

been consistently pulled into the mainline kernel.
—Kavan Smith

Florian Haas replies: *1) In context, the paragraph that followed explained that the "vendors" referred to were clearly distribution vendors. Between those, there clearly is some disparity in DRBD support, specifically in terms of how closely they are tracking upstream. It is also entirely normal for third parties to support their own products on a variety of distributions.* LJ *readers certainly need no reminder of this, and the article made no assertion to the contrary.*

2) From Linux 3.0 (in June 2011) to the time the article was published, the mainline kernel's drivers/block/drbd directory had seen ten commits and no significant merges. The drbd subdirectory of the DRBD 8.3 repository, where the out-of-tree kernel module is maintained, had 77 in the same time frame, including a substantial number of bug fixes. To speak of anything other than divergence seems odd, given the fact that the in-tree DRBD at a time lagged two point releases behind the out-of-tree code, and did not see substantial updates for four kernel releases straight—which, as many LJ *readers will agree, is also not exactly "standard operating procedure" for kernel modules. After the article ran, however, the DRBD developers submitted an update of the DRBD 8.3 codebase for the Linux 3.5 merge window, and it appears that DRBD 8.3 and the in-tree DRBD are now lining up again.*

The Digital Divide

I'm yet another reader who has mixed feelings about the new digital version of *LJ*, but I'm getting used to it. Unfortunately though, the transition to paperless just exacerbates the digital divide. Where I live in western Massachusetts, residents in most communities do not have access to better than dial-up or pretty-slow satellite service. I happen to be among the lucky few in my community to have DSL. But even over DSL, it takes several minutes to download the magazine. In general, I think I prefer the digital form of the publication. For one thing, it makes keeping back issues far more compact, and I guess being able to search for subjects should be useful. But, please do keep in mind that many of your readers probably live on the other side of the digital divide, being served by seriously slow connections. Keeping the file size more moderate will help those of us who are download-challenged. (By the way, in the community I live in, Leverett, Massachusetts, we are taking

steps to provide ourselves with modern connection speeds.)
—George Drake

I feel your pain, George. Here in northern Michigan, roughly half of our community members can't get broadband service. In an unexpected turn of events, it's starting to look like the cell-phone companies will be the first to provide broadband to the rural folks in my area. They've done a nice job installing more and more towers, and they have been marketing MiFi-like devices for home users. It's not the cheapest way to get broadband, but at least it's an option. Regarding the size of the digital issues, I've personally been impressed with Garrick Antikajian (our Art Director), as he keeps the file size remarkably low for the amount of graphics in the magazine. Hopefully that helps at least a little with downloading.—Ed.

Sharing *LJ*?

I'm a long-term subscriber of *LJ*. I was happy with the old printed version, and I'm happy with the new one. I don't want to go into the flaming world of printed vs. electronic, and I'm a bit tired of all those letters in *every* issue of *LJ*. But, I have a question. In the past, I used to pass my already-read issues to a couple of (young) friends, a sort of gift, as part of my "personal education in open source": helping others, especially young people, in developing an "open-source conscience" is a winning strategy for FOSS IMHO, together with access to the technical material. But now, what about with electronic *LJ*? Am I allowed to give away the *LJ* .pdf or .epub or .mobi after reading it? If not, this could lead to a big fail in FOSS! Hope you will have an answer to this. Keep rockin'!
—Ivan

Ivan, Linux Journal *is DRM-free, and the Texterity app offers some fairly simple ways to share content. We've always been anti-DRM for the very reasons you cite. Along with great power comes great responsibility though, so we hope you keep in mind that we also all still need to pay rent and feed our kids. Thanks for inquiring about it!—Ed.*

Digital on Portable Devices

I just subscribed to *LJ* for the first time in my life. I really love the digital formats. Things shipped to Bulgaria don't travel fast and often get "lost", although things probably have been a little bit better recently. Anyway, this way I can get the magazine hot off the press, pages burning my fingers. I still consider my Kindle 3 the best buy of the year, even though I bought it almost two years ago. It makes it easy to carry lots of bulky books with me. I already avoid buying paper books and tend to go digital if I can choose. Calibre

is my best friend, by the way. I have two recommendations to make. 1) Yesterday, I tried to download some .epubs on my Android phone. I logged in to my account and so on, but neither Dolphin nor the boat browser started the download. It would be great if you could check on and fix this problem, or provide the option in your Android app. 2) Send .mobi to the Kindle. This probably is not so easy to do, and I use Calibre to do it, but I still have to go through all the cable hassle.
—Stoyan Deckoff

I'm not sure why your Android phone gave you problems with the .epubs. Were you using the Linux Journal *app or downloading from e-mail? If the latter, maybe you need to save it and then "open" it from the e-book-reader app. As far as sending it to the Kindle, Amazon is getting quite flexible with its personal documents, and as long as you transfer over Wi-Fi, sending via e-mail often is free. Check out Amazon's personal document stuff and see if it fits your need.—Ed.*

Add CD and DVD ISO Images

It might be a good idea to sell CDs and DVDs as an encryption key (PGP) and send a specific link to a specifically generated downloadable image for each customer. This is a fairly old idea, a bit like what shareware programs used to do to unlock extra functionality. I accept that the pretty printed CD/DVD is nice to hold and for shelf cred. But an ISO is enough for me at least, apart from which we do seem to get offered a lot of them only an issue or two different. A very long-time reader (number 1 onward).
—Stephen

I'll be sure to pass the idea along, or are you just trying to start a war over switching the CD/DVDs to digital!??!!? Only teasing, of course.—Ed.

Electronic *LJ*

I love it. I just subscribed. I was going to use Calibre but forgot that my Firefox had EPUB Reader, and it's great. I turn my Ubuntu laptop display 90° left and have a nice big magazine. Keep up the good work.
—Pierre Kerr

I love the e-book-reader extension for Firefox! I have one for Chromium too, but it's not as nice as the Firefox extension. I'm glad you're enjoying the subscription.—Ed.

Reader Feedback

I think by now we all understand that there are people who do not like the fact that *LJ* is digital only and others who like it and some in between. Now, I can't imagine that these are the only letters you get from readers these days. It gets kind of old when every issue is filled with belly-aching about how bad a move it was to go digital (even if the alternative would've been to go bankrupt) and what not. We get it. I've been using Linux since 1993 and reading *Linux Journal* since the beginning. Let's move on and cut that whining.

—Michael

Michael, I do think we're close to "everything being said that can be said", but I assure you, we don't cherry-pick letters. We try to publish what we get, whether it's flattering or not. As you can see in this issue, we're starting to get more questions and suggestions about the digital issue. I think that's a good thing, and different from simply expressing frustration or praise. Maybe we're over the hump!—Ed.

Disgusting Ripoff

For weeks you've been sending me e-mails titled "Linux Weekly News", which is a well-known highly reputable community news site that has been in existence for almost as long as *Linux Journal*. By stealing its name and appropriating it for your own newsletter, you sink to the lowest of the low. I'm embarrassed I ever subscribed to a magazine that would steal from the Linux community in this way.

—Alan Robertson

Alan, I can assure you there was no ill intent. LWN is a great site, and we'd never intentionally try to steal its thunder. The newsletter actually was titled "Linux Journal Weekly News Notes" and has been around for several years. Over the course of time, it was shortened here and there to fit in subject lines better. We really like and respect the LWN crew and don't want to cause unnecessary confusion, so we're altering the name a bit to "Linux Journal Weekly News".—Ed.

Birthday Cake

I am a *Linux Journal* subscriber and Linux user since 2006. I got rid of Windows completely in 2007, and since then, my wife and I have been proud Ubuntu users and promote Linux to everyone we know.

I have been working in IT since 1981, and I am also the proud owner of a French blog since November 2011 that promotes Linux to French Canadians with our bi-monthly podcast and Linux articles. The blog is still very young and modest, but it's starting to generate some interesting traffic: **http://www.bloguelinux.ca** or **http://www.bloglinux.ca**.

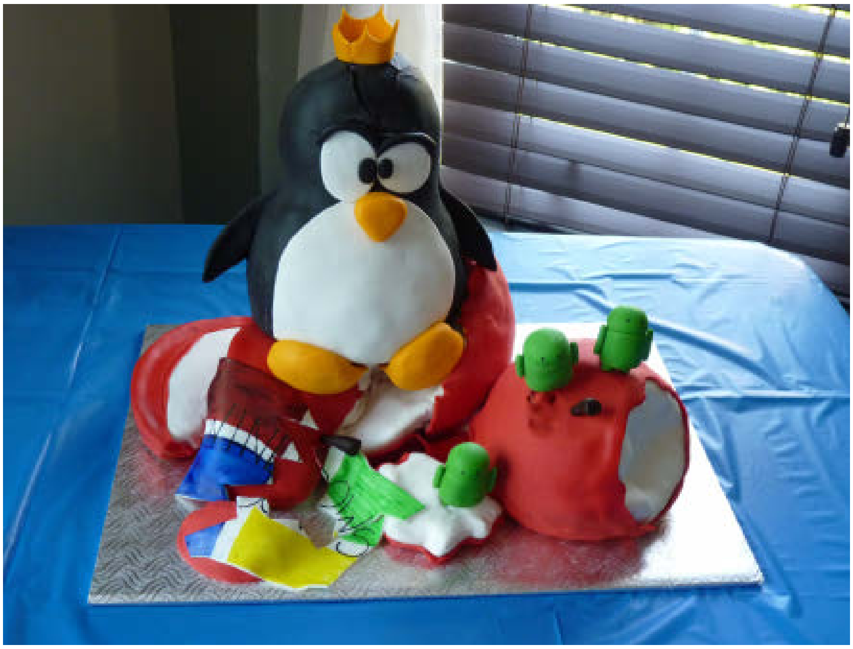
The reason for my writing is that I turned 50 on the 27th of May, and my wife got me a special cake to emphasize my passion for Linux. I wanted to share the pictures with everyone at *Linux Journal*.

The cake is a big Tux crushing an Apple. On its right is a broken Windows, and on the left, small Androids are eating an Apple.

The cake is a creation of La Cakerie in Quebec: **http://www.facebook.com/lacakerie**.

I'm not writing to promote anything, but I would be very proud to see a picture of my cake in one of your issues.
—Patrick Millette

I think the Linux Journal *staff should get to eat some of the cake too, don't you think? You know, for quality control purposes. Seriously though, that's awesome! Thanks for sending it in.—Ed.*



Patrick Millette's Awesome Birthday Cake

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

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://ww.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

An interesting side effect of last year's cyber attack on the kernel.org server was to identify which of the various services offered were most needed by the community. Clearly one of the hottest items was **git** repository hosting. And within the clamor for that one feature, much to **Willy Tarreau**'s surprise, there was a bunch of people who were very serious about regaining access to the **2.4 tree**.

Willy had been intending to bring this tree to its end of life, but suddenly a cache of users who cared about its continued existence was revealed. In light of that discovery, Willy recently announced that he intends to continue to update the 2.4 tree. He won't make any more versioned releases, but he'll keep adding fixes to the tree, as a centralized repository that 2.4 users can find and use easily.

Any attempt to simplify the **kernel licensing** situation is bound to be met with many objections. **Luis R. Rodriguez** discovered this recently when he tried to replace all kernel symbols indicating both the **GPL version 2** and some other license, like the **BSD** or **MPL**, with the simple text **"GPL-Compatible"**.

It sounds pretty reasonable. After all, the kernel really cares only if code is GPL-compatible so it can tell what interfaces to expose to that code, right? But, as was pointed out to Luis, tons of issues are getting in the way. For one thing, someone could interpret "GPL-Compatible" to mean that the code can be re-licensed under the **GPL version 3**, which **Linus Torvalds** is specifically opposed to doing.

For that matter, as also was pointed out, someone could interpret "GPL-Compatible" as indicating that the code in that part of the kernel could be re-licensed at any time to the second of the two licenses—the BSD or whatever—which also is not the case. Kernel code is all licensed under the GPL version 2 only. Any dual license applies to code distributed by the person who submitted it to the kernel in the first place. If you get it from that person, you can re-license under the alternate license.

Also, as **Alan Cox** pointed out, the license-related kernel symbols are likely to be valid evidence in any future court case, as indicating the intention of whomever released the code. So, if Luis or anyone else adjusted those symbols, aside from the person or organization who submitted the code in the first place, it could cause legal problems down the road.

And finally, as **Al Viro** and Linus Torvalds both said, the "GPL-Compatible" text only replaced text that actually contained useful information with something that was more vague.

It looks like an **in-kernel disassembler** soon will be included in the source tree. **Masami Hiramatsu** posted a patch implementing that specifically so **kernel oops** output could be rendered more readable.

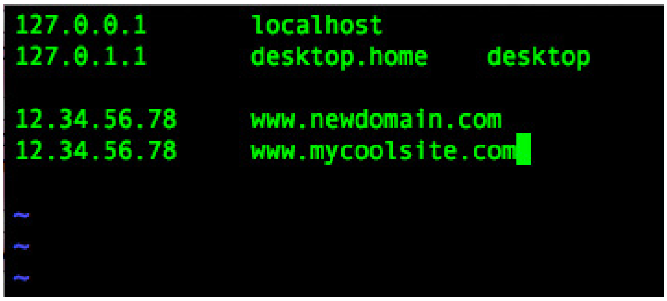
This probably won't affect regular users very much though. **H. Peter Anvin**, although in favor of the feature in general, wants users to have to enable it explicitly on the command line at bootup. His reasoning is that oops output already is plentiful and scrolls right off the screen. Masami's disassembled version would take up more space and cause even more of it to scroll off the screen.

With support from folks like H. Peter and **Ingo Molnar**, it does look as if Masami's patch is likely to go into the kernel, after some more work.—**ZACK BROWN**

Stop Waiting For DNS!

I am an impulse domain buyer. I tend to purchase silly names for simple sites that only serve the purpose of an inside joke. The thing about impulse-buying a domain is that DNS propagation generally takes a day or so, and setting up a Web site with a virtual hostname can be delayed while you wait for your Web site address to go "live".

Thankfully, there's a simple solution: the /etc/hosts file. By manually entering the DNS information, you'll get instant access to your new domain. That doesn't mean it will work for the rest of the Internet before DNS propagation, but it means you can set up and test your Web site immediately. Just remember to delete the entry in /etc/hosts after DNS propagates, or you might end up with a stale entry when your novelty Web site goes viral and you have to change your Web host!



```
127.0.0.1       localhost
127.0.1.1       desktop.home    desktop

12.34.56.78     www.newdomain.com
12.34.56.78     www.mycoolsite.com
~
~
~
```

The format for /etc/hosts is self-explanatory, but you can add comments by preceding with a # character if desired.

—**SHAWN POWERS**

Editors' Choice at LinuxJournal.com

Looking for software recommendations, apps and generally useful stuff? Visit **http://www.linuxjournal.com/editors-choice** to find articles highlighting various technology that merits our Editors' Choice seal of approval. We think you'll find this listing to be a valuable resource for discovering and vetting software, products and apps. We've run these things through the paces and chosen only the best to highlight so you can get right to the good stuff.

Do you know a product, project or vendor that could earn our Editors' Choice distinction? Please let us know at ljeditor@linuxjournal.com.

—KATHERINE DRUCKMAN

They Said It

Building one space station for everyone was and is insane: we should have built a dozen.

—Larry Niven

Civilization advances by extending the number of important operations which we can perform without thinking of them.

—Alfred North Whitehead

Do you realize if it weren't for Edison we'd be watching TV by candlelight?

—Al Boliska

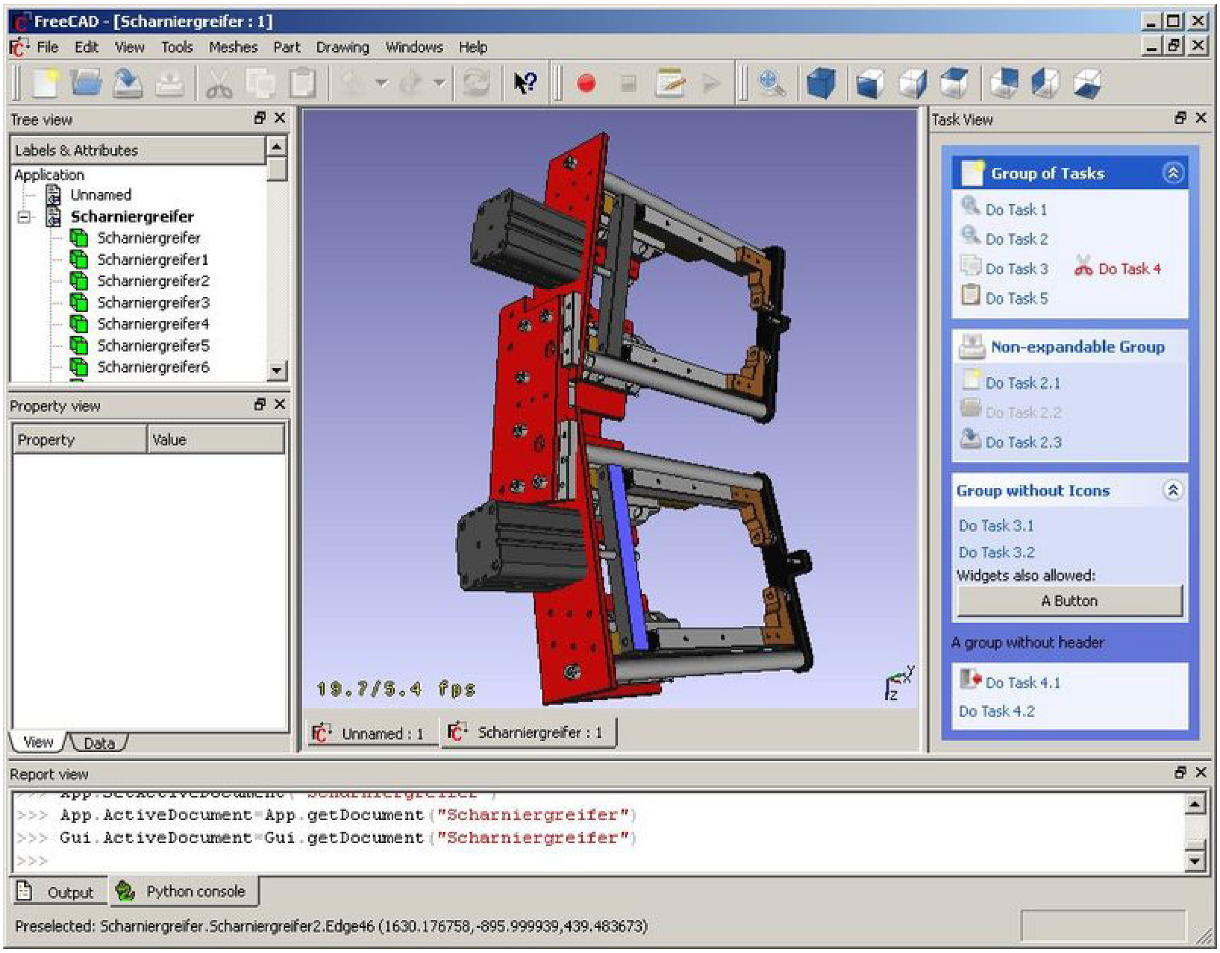
And one more thing...

—Steve Jobs

All right everyone, line up alphabetically according to your height.

—Casey Stengel

Non-Linux FOSS



```
>>> App.setActiveDocument("Scharniergreifer")
>>> App.ActiveDocument=App.getDocument("Scharniergreifer")
>>> Gui.ActiveDocument=Gui.getDocument("Scharniergreifer")
>>>
```

Preselected: Scharniergreifer.Scharniergreifer2.Edge46 (1630.176758,-895.999939,439.483673)

Although AutoCAD is the champion of the computer-aided design world, some alternatives are worth looking into. In fact, even a few open-source options manage to pack some decent features into an infinitely affordable solution.

QCAD from Ribbonsoft is one of those hybrid programs that has a fully functional GPL base (the Community Edition) and a commercial application, which adds functionality for a fee. On Linux, installing QCAD is usually as easy as a quick trip to your distro's package manager. For Windows users, however, Ribbonsoft offers source code, but nothing else. Thankfully, someone over at SourceForge has compiled QCAD for Windows, and it's downloadable from **http://qcadbin-win.sourceforge.net**.

For a completely free option, however, FreeCAD might be a better choice. With binaries available for Windows, OS X and Linux, FreeCAD is a breeze to distribute. In my very limited field testing, our local industrial arts teacher preferred FreeCAD over the other open-source alternatives, but because they're free, you can decide for yourself! Check out FreeCAD at **http://free-cad.sourceforge.net**.

—**SHAWN POWERS**

File Formats Used in Science

My past articles in this space have covered specific software packages, programming libraries and algorithm designs. One subject I haven't discussed yet is data storage, specifically data formats used for scientific information. So in this article, I look at two of the most common file formats: NetCDF (**http://www.unidata.ucar.edu/software/netcdf**) and HDF (**http://www.hdfgroup.org**). Both of these file formats include command-line tools and libraries that allow you to access these file formats from within your own code.

NetCDF (Network Common Data Format) is an open file format designed to be self-describing and machine-independent. The project is hosted by the Unidata program at UCAR (University Corporation for Atmospheric Research). UCAR is working on it actively, and version 4.1 was released in 2010.

NetCDF supports three separate binary data formats. The classic format has been used since the very first version of NetCDF, and it is still the default format. Starting with version 3.6.0, a 64-bit offset format was introduced that allowed for larger variable and file sizes. Then, starting with version 4.0, NetCDF/HDF5 was introduced, which was HDF5 with some restrictions. These files are meant to be self-describing as well. This means they contain a header that describes in some detail all of the data that is stored in the file.

The easiest way to get NetCDF is to check your distribution's package management system. Sometimes, however, the included version may not have the compile time settings that you need. In those cases, you need to grab the tarball and do a manual installation. There are interfaces for C, C++, FORTRAN 77, FORTRAN 90 and Java.

The classic format consists of a file that contains variables, dimensions and attributes. Variables are N-dimensional arrays of data. This is the actual data (that is, numbers) that you use in your calculations. This data can be one of six types (char, byte, short, int, float and double). Dimensions describe the axes of the data arrays. A dimension has a name and a length. Multiple variables can use the same dimension, indicating that they were measured on the same grid. At most, one dimension can be unlimited, meaning that the length can be updated continually as more data is added. Attributes allow you to store metadata about the file or variables. They can be either scalar values or one-dimensional arrays.

A new, enhanced format was introduced with NetCDF 4. To remain backward-compatible, it is constructed from the classic format plus some extra bits. One of the extra bits is the introduction of groups. Groups are hierarchical structures of data, similar to the UNIX filesystem. The second extra part is the ability to define new data types. A NetCDF 4 file contains one top-level unnamed group. Every group can contain one or more named subgroups, user-defined types, variables, dimensions and attributes.

Some standard command-line utilities are available to allow you to work with your NetCDF files. The ncdump utility takes the binary NetCDF file and outputs a text file in a format called CDL. The ncgen utility takes a CDL text file and creates a binary NetCDF file. nccopy copies a NetCDF file and, in the process, allows you to change things like the binary format, chunk sizes and compression. There are also the NetCDF Operators (NCOs). This project consists of a number of small utilities that do some operation on a NetCDF file, such as concatenation, averaging or interpolation.

Here's a simple example of a CDL file:

```
netcdf simple_xy {
dimensions:
   x = 6 ;
   y = 12 ;
variables:
   int data(x, y) ;
data:

 data =
  0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,
  12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23,
  24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,
  36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47,
  48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59,
  60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71 ;
}
```

Once you have this defined, you can create the corresponding NetCDF file with the ncgen utility.

To use the library, you need to include the header file netcdf.h. The library function names start with nc_. To open a file, use `nc_open(filename, access_mode, file_pointer)`. This gives you a file pointer that you can use to read from and write to the file. You then need to get a variable identifier with the function `nc_inq_varid(file_pointer, variable_name, variable_identifier)`. Now you can actually read in the data with the function `nc_get_var_int(file_pointer, variable_identifier, data_buffer)`, which will place the data into the data buffer in your code. When you're done, close the file with `nc_close(file_pointer)`. All of these functions return error codes, and they should be checked after each execution of a library function.

Writing files is a little different. You need to start with `nc_create`, which gives you a file pointer. You then define the dimensions with the `nc_def_dim` function. Once these are all defined, you can go ahead and create the variables with the `nc_def_var` function. You need to close off the header with `nc_enddef`. Finally, you can start to write out the data itself with `nc_put_var_int`. Once all of the data is written out, you can close the file with `nc_close`

The Hierarchical Data Format (HDF) is another very common file format used in scientific data processing. It originally was developed at the National Center for Supercomputing Applications, and it is now maintained by the nonprofit HDF Group. All of the libraries and utilities are released under a BSD-like license. Two options are available: HDF4 and HDF5. HDF4 supports things like multidimensional arrays, raster images and tables. You also can create your own grouping structures called vgroups. The biggest limitation to HDF4 is that file size is limited to 2GB maximum. There also isn't a clear object structure, which limits the kind of data that can be represented. HDF5 simplifies the file format so that there are only two types of objects: datasets, which are homogeneous multidimensional arrays, and groups, which are containers that can hold datasets or other groups. The libraries have interfaces for C, C++, FORTRAN 77, FORTRAN 90 and Java, similar to NetCDF.

The file starts with a header, describing details of the file as a whole. Then, it will contain at least one data descriptor block, describing the details of the data stored in the file. The file then can contain zero or more data elements, which contain the actual data itself. A data descriptor block plus a data element block is represented as a data object. A data descriptor is 12-bytes long, made up of a 16-bit tag, a 16-bit reference number, a 32-bit data offset and a 32-bit data length.

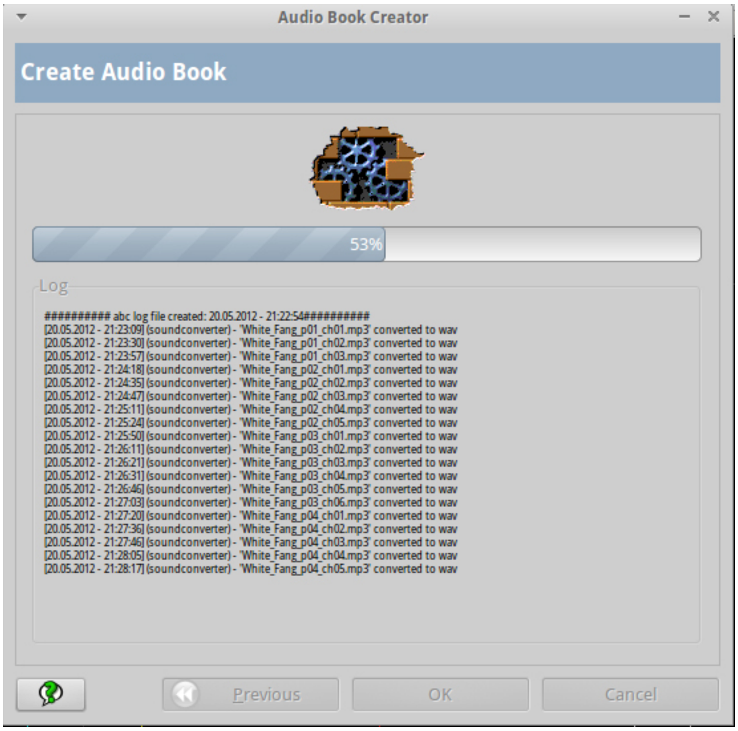
Several command-line utilities are available for HDF files too. The hdp utility is like the ncdump utility. It gives a text dumping of the file and its data values. hdiff gives you a listing of the differences between two HDF files. hdfls shows information on the types of data objects stored in the file. hdfed displays the contents of an HDF file and gives you limited abilities to edit the contents. You can convert back and forth between HDF4 and HDF5 with the h4toh5 and h5toh4 utilities. If you need to compress the data, you can use the hdfpack program. If you need to alter options, like compression or chunking, you can use hrepack.

The library API for HDF is a bit more complex than for NetCDF. There is a low-level interface, which is similar to what you would see with NetCDF. Built on top of this is a whole suite of different interfaces that give you higher-level functions. For example, there is the scientific data sets interface, or SD. This provides functions for reading and writing data arrays. All of the functions begin with SD, such as SDcreate to create a new file. There are many other interfaces, such as for palettes (DFP) or 8-bit raster images (DFR8). There are far too many to cover here, but there is a great deal of information, including tutorials, that can help you get up to speed with HDF.

Hopefully now that you have seen these two file formats, you can start to use them in your own research. The key to expanding scientific understanding is the free exchange of information. And in this age, that means using common file formats that everyone can use. Now you can go out and set your data free too.

—**JOEY BERNARD**

Audiobooks as Easy as ABC



```
########## abc log file created: 20.05.2012 - 21:22:54##########
[20.05.2012 - 21:23:09] (soundconverter) - 'White_Fang_p01_ch01.mp3' converted to wav
[20.05.2012 - 21:23:30] (soundconverter) - 'White_Fang_p01_ch02.mp3' converted to wav
[20.05.2012 - 21:23:57] (soundconverter) - 'White_Fang_p01_ch03.mp3' converted to wav
[20.05.2012 - 21:24:18] (soundconverter) - 'White_Fang_p02_ch01.mp3' converted to wav
[20.05.2012 - 21:24:35] (soundconverter) - 'White_Fang_p02_ch02.mp3' converted to wav
[20.05.2012 - 21:24:47] (soundconverter) - 'White_Fang_p02_ch03.mp3' converted to wav
[20.05.2012 - 21:25:11] (soundconverter) - 'White_Fang_p02_ch04.mp3' converted to wav
[20.05.2012 - 21:25:24] (soundconverter) - 'White_Fang_p02_ch05.mp3' converted to wav
[20.05.2012 - 21:25:50] (soundconverter) - 'White_Fang_p03_ch01.mp3' converted to wav
[20.05.2012 - 21:26:11] (soundconverter) - 'White_Fang_p03_ch02.mp3' converted to wav
[20.05.2012 - 21:26:21] (soundconverter) - 'White_Fang_p03_ch03.mp3' converted to wav
[20.05.2012 - 21:26:31] (soundconverter) - 'White_Fang_p03_ch04.mp3' converted to wav
[20.05.2012 - 21:26:46] (soundconverter) - 'White_Fang_p03_ch05.mp3' converted to wav
[20.05.2012 - 21:27:03] (soundconverter) - 'White_Fang_p03_ch06.mp3' converted to wav
[20.05.2012 - 21:27:20] (soundconverter) - 'White_Fang_p04_ch01.mp3' converted to wav
[20.05.2012 - 21:27:36] (soundconverter) - 'White_Fang_p04_ch02.mp3' converted to wav
[20.05.2012 - 21:27:46] (soundconverter) - 'White_Fang_p04_ch03.mp3' converted to wav
[20.05.2012 - 21:28:05] (soundconverter) - 'White_Fang_p04_ch04.mp3' converted to wav
[20.05.2012 - 21:28:17] (soundconverter) - 'White_Fang_p04_ch05.mp3' converted to wav
```

Whether you love Apple products or think they are abominations, it's hard to beat iPods when it comes to audiobooks. They remember your place, support chapters and even offer speed variations on playback. Thanks to programs like Banshee and Amarok, syncing most iPod devices (especially the older iPod Nanos, which are perfect audiobook players) is simple and works out of the box.

The one downside with listening to audiobooks on iPods is that they accept only m4b files. Most audiobooks either are ripped from CDs into MP3 files or are downloaded as MP3 files directly. There are some fairly simple command-line tools for converting a bunch of MP3 files into iPod-compatible m4b files, but if GUI tools are your thing, Audio Book Creator (ABC) might be right up your alley.

ABC is a very nice GUI application offered by a German programmer. The Web site is **http://www.ausge.de**, and although the site is in German, the program itself is localized and includes installation instructions in English. The program does require a few dependencies to be installed, but the package includes very thorough instructions. If you want to create iPod-compatible audiobooks, ABC is as simple as, well, ABC!

—**SHAWN POWERS**

Networking Poll

We recently asked LinuxJournal.com readers about their networking preferences, and after calculating the results, we have some interesting findings to report. From a quick glance, we can see that our readers like their Internet fast, their computers plentiful and their firewalls simple.

One of the great things about *Linux Journal* readers and staff is that we all have a lot in common, and one of those things is our love of hardware. We like to have a lot of it, and I suspect we get as much use out of it as we can before letting go, and thus accumulate a lot of machines in our houses. When asked how many computers readers have on their home networks, the answer was, not surprisingly, quite a few! The most popular answer was 4–6 computers (44% of readers); 10% of readers have more than 10 computers on their home networks (I'm impressed); 14% of readers have 7–9 running on their networks, and the remaining 32% of readers have 1–3 computers.

We also asked how many of our surveyed readers have a dedicated server on their home networks, and a slight majority, 54%, responded yes. I'm pleased to know none of us are slacking on our home setups in the least!

Understandably, these impressive computing environments need serious speed. And while the most common Internet connection speed among our surveyed readers was a relatively low 1–3mbps (17% of responses), the majority of our readers connect at relatively fast speeds. The very close second- and third-most-common speeds were 6–10mbps and an impressive more than 25mbps, respectively, and each representing 16% of responses. A similarly large number of surveyed readers were in the 10–15mbps and 15–25mbps ranges, so we're glad to know so many of you are getting the most out of your Internet experience.

The vast majority of our readers use cable and DSL Internet services. Cable was the slight leader at 44% vs. 41% for DSL. And 12% of readers have a fiber connection—and to the mountain-dwelling Canadian reader connected via long-range Wi-Fi 8km away, I salute you! Please send us photos of your view.

The favorite wireless access point vendor is clearly Linksys, with 30% of survey readers using some type of Linksys device. NETGEAR and D-Link have a few fans as well, each getting 15% of the delicious response pie. And more than a handful of you pointed out that you do not use any wireless Internet. I admit, I'm intrigued.

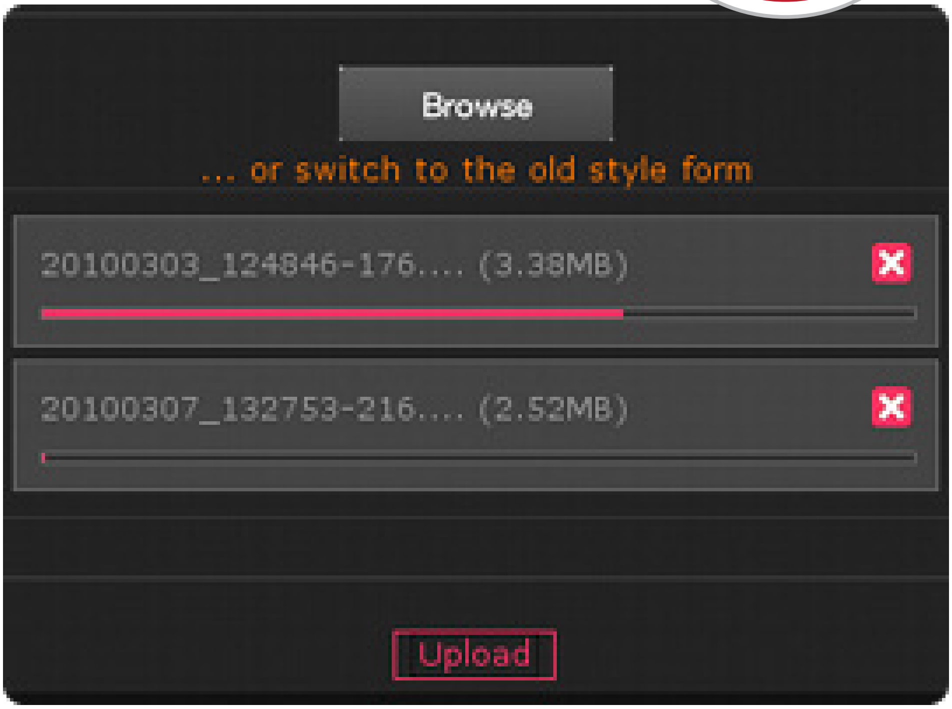
Finally, when asked about your preferred firewall software/appliance, the clear winner was "Stock Router/AP Firmware" with 41% of respondents indicating this as their preferred method. We respect your tendency to keep it simple. In a distant second place, with 15%, was a custom Linux solution, which is not surprising given our readership's penchant for customization in all things.

Thanks to all who participated, and please look to LinuxJournal.com for future polls and surveys.

—KATHERINE DRUCKMAN

CLICK HERE
PLAY

Build Your Own Flickr with Piwigo



20100303_124846-176.... (3.38MB)

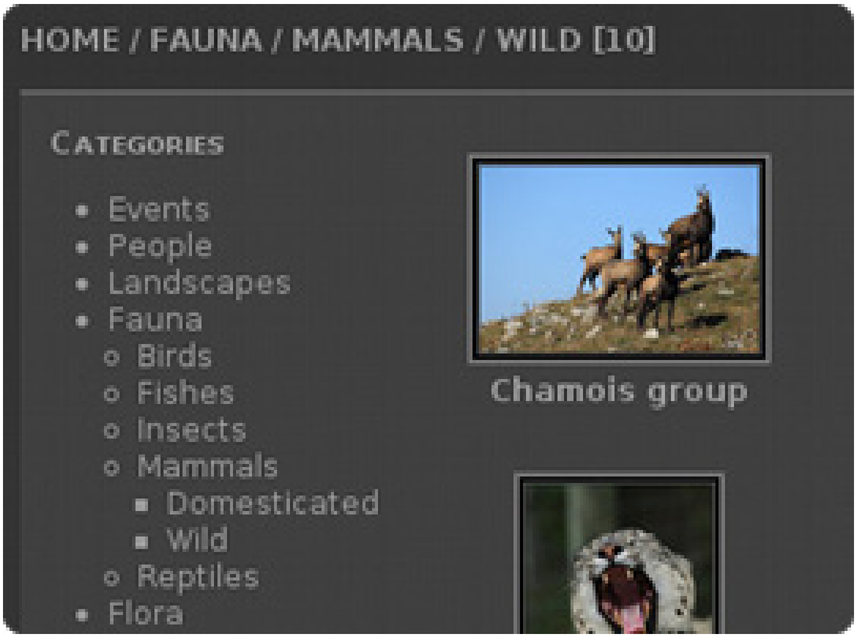
20100307_132753-216.... (2.52MB)

Piwigo supports direct upload of multiple files, but it also supports third-party upload utilities (screenshot courtesy of http://www.piwigo.org).

In 2006, the family computer on which our digital photographs were stored had a hard drive failure. Because I'm obsessed with backups, it shouldn't have been a big deal, except that my backups had been silently failing for months. Although I certainly learned a lesson about verifying my backups, I also realized it would be nice to have an off-site storage location for our photos.

Move forward to 2010, and I realized storing our photos in the "cloud" would mean they were always safe and always accessible. Unfortunately, it also meant my family memories were stored by someone else, and I had to pay for the privilege of on-line access. Thankfully, there's an open-source project designed to fill my family's need, and it's a mature project that just celebrated its 10th anniversary!

Piwigo, formerly called PhpWebGallery, is a Web-based program designed to upload, organize and archive photos. It supports tagging, categories, thumbnails and pretty much every other on-line sorting tool you can imagine. Piwigo has been around long enough that there even are third-party applications that support it out of the box. Want mobile support? The Web site has a mobile theme built in. Want a native app for your phone? iOS and Android apps are available. In fact, with its numerous extensions and third-party applications, Piwigo rivals sites like Flickr and Picasaweb when it comes to flexibility.



HOME / FAUNA / MAMMALS / WILD [10]

CATEGORIES

- Events
- People
- Landscapes
- Fauna
  - Birds
  - Fishes
  - Insects
  - Mammals
    - Domesticated
    - Wild
  - Reptiles
- Flora

Chamois group

Categories, tags, albums and more are available to organize your photos (screenshot courtesy of http://www.piwigo.org).

Plus, because it's open source, you control all your data.

If you haven't considered Piwigo, you owe it to yourself to try. It's simple to install, and if you have a recent version of Linux, your distribution might have it by default in its repositories. Thanks to its flexibility, maturity and downright awesomeness, Piwigo gets this month's Editors' Choice award. Check it out today at **http://www.piwigo.org**.

—**SHAWN POWERS**

REUVEN M. LERNER

Pry

Interact with your Ruby code more easily with Pry, a modern replacement for IRB.

I spend a fair amount of my time teaching courses, training programmers in the use of Ruby and Python, as well as the PostgreSQL database. And as if my graying hair weren't enough of an indication that I'm older than many of these programmers, it's often shocking for them to discover I spend a great deal of time with command-line tools. I'm sure that modern IDEs are useful for many people—indeed, that's what they often tell me—but for me, GNU Emacs and a terminal window are all I need to have a productive day.

In particular, I tell my students, I cannot imagine working without having an interactive copy of the language open in parallel. That is, I will have one or more Emacs buffers open, and use it to edit my code. But I'll also be sure to have a Python or Ruby (or JavaScript) interpreter open in a separate window. That's where I do much of my work—trying new ideas, testing code, debugging code that should have worked in production but didn't, and generally getting a "feel" for the program I'm trying to write.

Indeed, "feeling" the code is a phenomenon I'm sure other programmers understand, and I believe it's crucial when really trying to understand what is going on in a program. It's sort of like learning a new foreign language. At a certain point, you have an instinct for what words and conjugations should work, even if you've never used them before. Sometimes, when things go wrong, if you have enough experience working with the code, you will have an internal sense of what has gone wrong—where to look and how to fix things. This comes from interacting and working with the code on a day-to-day basis.

One of the advantages of a dynamic, interpreted language, such as Python or Ruby, is that you can use a REPL (read-eval-print loop), a program that gives you the chance to interact with the language directly, typing commands and then getting responses. A good REPL will let you do everything from experimenting with one-liners to creating new classes and modules. You're obviously not going to create production code in such an environment, but you might well create

Indeed, if you are a Python programmer and not using iPython in your day-to-day work, you should run to your computer, install it and start to use it.

some classes, objects and methods, and then experiment with them to see how well they work.

I have been using both Python and Ruby for a number of years, and I teach classes in both languages on a regular basis. Part of these classes always involves introducing students to the interactive versions of these languages—the `python` command in the case of Python and `irb` in the case of Ruby.

About a year ago, one of my Python students asked me what I knew about iPython. The fact is that I had heard of it, but hadn't really thought to check much into the project. At home that night, I was pretty much blown away by what it could do, and I scolded myself for not having tried it earlier. Indeed, if you are a Python programmer and not using iPython in your day-to-day work, you should run to your computer, install it and start to use it. It offers a wide and rich variety of functions that provide specific supports for interacting with the language. Of particular interest to me, when teaching my classes, is the ability to log everything I type. At the end of the day, I can send a complete, verbatim log of everything I've written (which is a lot!) to the students.

I have had a similar experience with Ruby during the past few months. When Pry was announced about a year ago, described as a better version of Ruby's interactive IRB program, I didn't really do much with it. But during the past few weeks, I have been using and thoroughly enjoying Pry. I have incorporated it into my courses, and have—as in the case of iPython—wondered how it could be that I ignored such a wonderful tool for as long as I did.

This month, I take a look at Pry, an improved REPL for Ruby. It not only allows you to swap out IRB, the standard interactive shell for Ruby, but it also lets you replace the Rails console. The console is already a powerful tool, but combined with Pry's ability to explore data structures, display documentation, edit code on the fly, and host a large and growing number of plugs, it really sings.

Pry

Pry is a relative newcomer in the Ruby world, but it has become extremely popular, in no small part thanks to Ryan Bates, whose wonderful weekly "Railscasts" screencasts introduced it several months ago. Pry is an attempt to remake IRB, the interactive Ruby

Pry is an attempt to remake IRB, the interactive Ruby interpreter, in a way that makes more sense for modern programmers.

interpreter, in a way that makes more sense for modern programmers.

Installing Pry is rather straightforward. It is a Ruby gem, meaning that it can be installed with:

```
gem install pry pry-doc
```

You actually don't need to install pry-doc, but you really will want to do so, as I'll demonstrate a bit later.

I tend to use the -V (verbose) switch when installing gems to see more output on the screen and identify any problems that occur. You also might notice that I have not used sudo to install the gem. That's because I'm using rvm, the Ruby version manager, which allows me to install and maintain multiple versions of Ruby under my home directory. If you are using the version of Ruby that came with your system, you might need to preface the above command with sudo. Also, I don't believe that Pry works with Ruby 1.8, so if you have not yet switched to Ruby 1.9, I hope Pry will encourage you to do so.

Once you have installed Pry, you should have an executable program called "pry" in your path, in the same place as other gem-installed executables. So you can just type pry, and you will be greeted by the following prompt:

```
[1] pry(main)>
```

You can do just about anything in Pry that you could do in IRB. For example, I can create a class, and then a new instance of that class:

```
[2] pry(main)> class Person
[2] pry(main)*   def initialize(first_name, last_name)
[2] pry(main)*     @first_name = first_name
[2] pry(main)*     @last_name = last_name
[2] pry(main)*   end
[2] pry(main)* end
```

Now, you can't see it here, but as I typed, the words "class", "Person", "def" and "end" were all colorized, similarly to how a modern editor colorizes keywords. The indentation also was adjusted automatically, ensuring that the "end" words line up with the lines that open those blocks.

Once I have defined this class, I can create some new instances. Here are two of them:

```
[3] pry(main)> p1 = Person.new('Reuven', 'Lerner')
=> #<Person:0x007ff832949580 @first_name="Reuven", @last_name="Lerner">
```

```
[4] pry(main)> p2 = Person.new('Shikma', 'Lerner-Friedman')
=> #<Person:0x007ff8332386c8 @first_name="Shikma",
@last_name="Lerner-Friedman">
```

As expected, after creating these two instances, you'll see a printed representation of these objects. Now, let's say you want to inspect one of these objects more carefully. One way to do it is to act on the object from the outside, as you are used to doing. But Pry treats every object as a directory-like, or namespace-like, object, which you can set as the current context for your method calls. You change the context with the cd command:

```
cd p2
```

When doing this, you see that the prompt has changed:

```
[14] pry(#<Person>):1>
```

In other words, I'm now on line 14 of my Pry session. However, I'm currently not at the main level, but rather inside an instance of Person. This means I can look at the object's value for @first_name just by typing that:

```
[15] pry(#<Person>):1> @first_name
=> "Shikma"
```

Remember that in Ruby, instance variables are private. The only way to access them from outside the object itself is via a method. Because I haven't defined any methods, there isn't any way (other than looking at the printed representation using the #inspect method) to see the contents of instance variables. So the fact that you can just write @first_name and get its contents is pretty great.

But wait, you can do better than this; @first_name is a string, so let's go into that:

```
[17] pry(#<Person>):1> cd @first_name
[18] pry("Shikma"):2> reverse
=> "amkihS"
```

As you can see, by cd-ing into @first_name, any method calls now will take place against @first_name (that is, the text string) allowing you to play with it there. You also see how the prompt, just before the > sign at the end, now has a :1 or :2, indicating how deep you have gone into the object stack.

If you want to see how far down you have gone, you can type nesting, which will show you the current context in the code, as well as the above contexts:

```
[19] pry("Shikma"):2> nesting
Nesting status:
--
0. main (Pry top level)
1. #<Person>
2. "Shikma"
```

You can return to the previous nesting level with exit or jump to an arbitrary

Pry supports readline, meaning that I can use my favorite Emacs editing bindings—my favorite being Ctrl-R, for reverse i-search—in the command line.

level with `jump-to N`, where N is a defined nesting level:

```
[25] pry("Shikma"):2> nesting
Nesting status:
--
0. main (Pry top level)
1. #<Person>
2. "Shikma"

[26] pry("Shikma"):2> jump-to 1

[27] pry(#<Person>):1> nesting
Nesting status:
--
0. main (Pry top level)
1. #<Person>

[28] pry(#<Person>):1> exit
=> nil

[29] pry(main)> nesting
Nesting status:
--
0. main (Pry top level)
```

When I first learned about Pry, I worried that `cd` and `ls` were taken for objects and, thus, those commands would be unavailable for directory traversal. Never fear; all shell commands, from `cd` to `ls` to `git`, are available from within Pry, if you preface them with a . character.

Editing Code

Pry supports readline, meaning that I can use my favorite Emacs editing bindings—my favorite being Ctrl-R, for reverse i-search—in the command line. Even so, I sometimes make mistakes and need to correct them. Pry understands this and offers many ways to interact with its shell.

My favorite is !, the exclamation point, which erases the current input buffer. If I'm in the middle of defining a class or a method and want to clear everything, I can just type !, and everything I've written will be forgotten. I have found this to be quite useful.

But, there are more practical items as well. Let's say I want to modify the "initialize" method I wrote before. Well, I can just use the edit-method command:

```
edit-method Person#initialize
```

Because my EDITOR environment variable is set to "emacsclient", this opens up a buffer in Emacs, allowing me to edit that particular method. I change it to take three parameters instead of two, save it and then exit back to Pry, where I find that it already has been loaded into memory:

```
[52] pry(main)> p3 = Person.new('Amotz', 'Lerner-Friedman')
ArgumentError: wrong number of arguments (2 for 3)
from (pry):35:in `initialize'
```

Thanks to installing the pry-doc gem earlier, I even can get the source for any method on my system—even if it is written in C! For example, I can say:

```
show-method String#reverse
```

and I get the C source for how Ruby implements the "reverse" instance method on String. I must admit, I have been working with open source for years and have looked at a lot of source code, but having the source for the entire Ruby standard library at my fingertips has greatly increased the number of times I do this.

Rails Integration

Finally, Pry offers several types of integration with Ruby on Rails. The Rails console is basically a version of IRB that has loaded the Rails environment, allowing developers to work directly with their models, among other things. Pry was designed to work with Rails as well.

The easiest way to use Pry instead of IRB in your Rails console is to fire it up, using the -r option to require a file—in this case, the config/environment.rb file that loads the appropriate items for the Rails environment. So I was able to run:

```
pry -r ./config/environment
```

On my production machine, of course, I had to say:

```
RAILS_ENV=production pry -r ./config/environment
```

Once I had done this, I could navigate through the users on my system—for example:

```
u = User.find_by_email("reuven@lerner.co.il")
```

Sure enough, that put my user information in the variable u. I could have invoked all sorts of stuff on u, but instead, I entered the variable:

```
cd u
```

Then I was able to invoke the "name" method, which displays the full name:

```
[14] pry(#<User>):2> name
=> "Reuven Lerner"
```

But this isn't the best trick of all. If I add Pry into my Gemfile, as follows:

```
gem 'pry', :group => :development
```

Pry will be available during development. This means anywhere in my code, I can stick the line:

```
binding.pry
```

and when execution reaches that line, it will stop, dropping me into a Pry session. This works just fine when using Webrick, but it also can be configured to work with Pow, a popular server system for OS X:

```
def show
  binding.pry
end
```

I made the above modification to one of the controllers on my site, and then pointed my browser to a page on which it would be invoked. It took a little bit of time, but the server eventually gave way to a Pry prompt. The prompt worked exactly as I might have expected, but it showed me the current line of execution within the controller, letting me explore and debug things on a live (development) server. I was able to explore the state of variables at the beginning of this controller action, which was much better and more interactive than my beloved logging statements.

Conclusion

Pry is an amazing replacement for the default IRB, as well as for the Rails console. There still are some annoyances, such as its relative slowness (at least, in my experience) and the fact that readline doesn't always work perfectly with my terminal-window configuration. And as often happens, the existence of a plugin infrastructure has led to a large collection of third-party plugins that handle a wide variety of tasks.

That said, these are small problems compared with the overwhelmingly positive experience I have had with Pry so far. If you're using Ruby on a regular basis, it's very much worth your while to look into Pry. I think you'll be pleasantly surprised by what you find.■

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 Pry is **https://github.com/pry/pry**. You can download the source for Pry from Git, or (as mentioned above) just install the Ruby gem. The Pry home page includes a GitHub Wiki with a wealth of information and FAQs about Pry, its installation, configuration and usage.

A nice blog post introducing Pry is at **http://www.philaquilina.com/2012/05/17/tossing-out-irb-for-pry**.

Finally, a Railscast about using Pry, both with and without Rails, is at **http://railscasts.com/episodes/280-pry-with-rails**.

I also mentioned iPython at the beginning of this column. Pry and iPython are very similar in a number of ways, although iPython is more mature and has a larger following. If you work with Python, you owe it to yourself to try iPython at **http://ipython.org**.

TXLF

Come be a part of the Texas Linux Fest 2012 - The largest Linux and Open Source Software conference in the Lone Star State!

Norris Conference Center in
San Antonio, TX
August 3-4th

register @ http://texaslinuxfest.org

texaslinuxfest

DAVE TAYLOR

Subshells and Command-Line Scripting

No games to hack this time; instead, I go back to basics and talk about how to build sophisticated shell commands directly on the command line, along with various ways to use subshells to increase your scripting efficiency.

I've been so busy the past few months writing scripts, I've rather wandered away from more rudimentary tutorial content. Let me try to address that this month by talking about something I find I do quite frequently: turn command-line invocations into short scripts, without ever actually saving them as separate files.

This methodology is consistent with how I create more complicated shell scripts too. I start by building up the key command interactively, then eventually do something like this:

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

to get what I've built up as the starting point of my shell script.

Renaming Files

Let's start with a simple example. I find that I commonly apply rename patterns to a set of files, often when it's something like a set of images denoted with the .JPEG suffix, but because I prefer lowercase, I'd like them changed to .jpg instead.

This is the perfect situation for a command-line for loop—something like:

```
for filename in *.JPEG
do
   commands
done
```

That'll easily match all the relevant files, and then I can rename them one by one.

Linux doesn't actually have a rename utility, however, so I'll need to use mv

> **Linux doesn’t actually have a rename utility, however, so I’ll need to use mv instead, which can be a bit confusing.**

instead, which can be a bit confusing. The wrinkle is this: how do you take an existing filename and change it as desired? For that, I use a subshell:

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

When I’ve talked in previous columns about how sed can be your friend and how it’s a command well worth exploring, now you can see I wasn’t just filling space. If I just wanted to fill space, I’d turn in a column that read “all work and no play makes Jack a dull boy”.

Now that the old name is “filename” and the new name is “newname”, all that’s left is actually to do the rename. This is easily accomplished:

```
mv $filename $newname
```

There’s a bit of a gotcha if you encounter a filename with a space in its name, however, so here’s the entire script (with one useful line added so you can see what’s going on), as I’d type in directly on the command line:

```
for filename in *.JPEG ; do
  newname="$(echo $filename | sed 's/.JPEG/.jpg/')"
  echo "Renaming $filename to $newname
  mv "$filename" "$newname"
done
```

If you haven’t tried entering a multi-line command directly to the shell, you also might be surprised by how gracefully it handles it, as shown here:

```
$ for filename in *.JPEG
>
```

The > denotes that you’re in the middle of command entry—handy. Just keep typing in lines until you’re done, and as soon as it’s a syntactically correct command block, the shell will execute it immediately, ending with its output and a new top-level prompt.

More Sophisticated Filename Selection

Let’s say you want to do something similar, but instead of changing filenames, you want to change the spelling of someone’s name within a subset of files. It turns out that Priscilla actually goes by “Pris”. Who knew?

There are a couple ways you can accomplish this task, including tapping the powerhouse find command with its

How to get that into the for loop? You could use a temporary output file, but that's a lot of work.

-exec predicate, but because this is a shell scripting column, let's look at how to expand the for loop structure shown above.

The key difference is that in the "for name in pattern" sequence, you need to have pattern somehow reflect the result of a search of the contents of a set of files, not just the filenames. That's done with grep, but this time, you don't want to see the matching lines, you just want the names of the matching files. That's what the -l flag is for, as explained:

```
"-l     Only the names of files containing selected lines
               are written to standard output."
```

Sounds right. Here's how that might look as a command:

```
$ grep -l "Priscilla" *.txt
```

The output would be a list of filenames.

How to get that into the for loop? You could use a temporary output file, but that's a lot of work. Instead, just as I invoked a subshell for the file rename (the "$( )" notation earlier), sometimes you'll also see subshells written with backticks: `cmd`. (Although I prefer $( ) notation myself.)

Putting it together:

```
for filename in $(grep -l "Priscilla" *.txt) ; do
```

Fixing Priscilla's name in the files can be another job for sed, although this time I would tap into a temporary filename and do a quick switch:

```
sed "s/Priscilla/Pris/g" "$filename" > $tempfile
mv "$tempfile" "$filename"
echo "Fixed Priscilla's name in $filename"
```

See how that works?

The classic gotcha in this situation is file permissions. An unexpected consequence of this rewrite is that the file not only has the pattern replaced, it also potentially gains a new owner and new default file permissions. If that's a potential problem, you'll need to grab the owner and current permissions before the mv command, then use chown and chmod to restore the file owner and permission, respectively.

Performance Issues

Theoretically, launching lots of subshells could have a performance hit as the Linux system has to do a lot more than just run individual commands as it invokes additional shells, passes variables and so on. In practice, however, I've found this sort of penalty to be negligible and think it's safe to ignore. If a subshell or two is the right way to proceed, just go for it.

That's not to say it's okay to be sloppy and write highly inefficient code. My mantra is that the more you're going to use the script, the smarter it is to spend the time to make it efficient and bomb-proof. That is, in the earlier scripts, I've ignored any tests for input validity, error conditions and meaningful output if there are no matches and so on.

Those can be added easily, along with a usage section so that a month later you remember exactly how the script works and what command flags you've added over time. For example, I have a 250-line script I've been building during the past year or two that lets me do lots of manipulation with HTML image tags. Type in just its name, and the output is prolific:

```
$ scale
Usage: scale {args} factor [file or files]
  -b      add 1px solid black border around image
  -c      add tags for a caption
  -C xx   use specified caption
  -f      use URL values for DaveOnFilm.com site
  -g      use URL values for GoFatherhood site
  -i      use URL values for intuitive.com/blog site
  -k KW   add keywords KW to the ALT tags
  -r      use 'align=right' instead of <center>
  -s      produces succinct dimensional tags only
  -w xx   warn if any images are more than the specified width
  factor  0.X for X% scaling or max width in pixels.
          A scaling factor of '1' produces 100%
```

Because I often go months without needing the more obscure features, it's extremely helpful and easily added to even the most simple of scripts.

Conclusion

I've spent the last year writing shell scripts that address various games. I hope you've found it useful for me to step back and talk about some basic shell scripting methodology. If so, let me know!■

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

Getting Started with 3-D Printing: the Software

Thinking about getting a 3-D printer? Find out what software you'll need to use it.

This column is the second of a two-part series on 3-D printing. In Part I, I discussed some of the overall concepts behind 3-D printing and gave an overview of some of the hardware choices that exist. In this article, I finish by explaining the different categories of software you use to interface with a 3-D printer, and I discuss some of the current community favorites in each category.

In part due to the open-source leanings of the 3-D printer community, a number of different software choices under Linux are available that you can use with the printer. Like with desktop environments or Web browsers, what software you use is in many cases a matter of personal preference. This is particularly true if your printer is from the RepRap family, because there's no "official" software bundle; instead, everyone in the community uses the software they feel works best for them at a particular time. The software is still, in some cases, in an early phase, so it pays to keep up on the latest and greatest features and newest releases. Instead of getting involved in a holy war over what software is best, I cover some of the more popular software choices and highlight what I currently use, which is based on a general consensus I've gathered from the RepRap community.

In part due to the rapid advancement in this software, and in part due to how new a lot of the software is, in most cases, you won't find any of this software packaged for your distribution. Installation then is a lot like what some of you might remember from the days before package managers like APT. Each program has its own library dependencies listed in its install documentation, and generally the software installs by extracting a tarball (which contains precompiled binaries) into some directory of your choice.

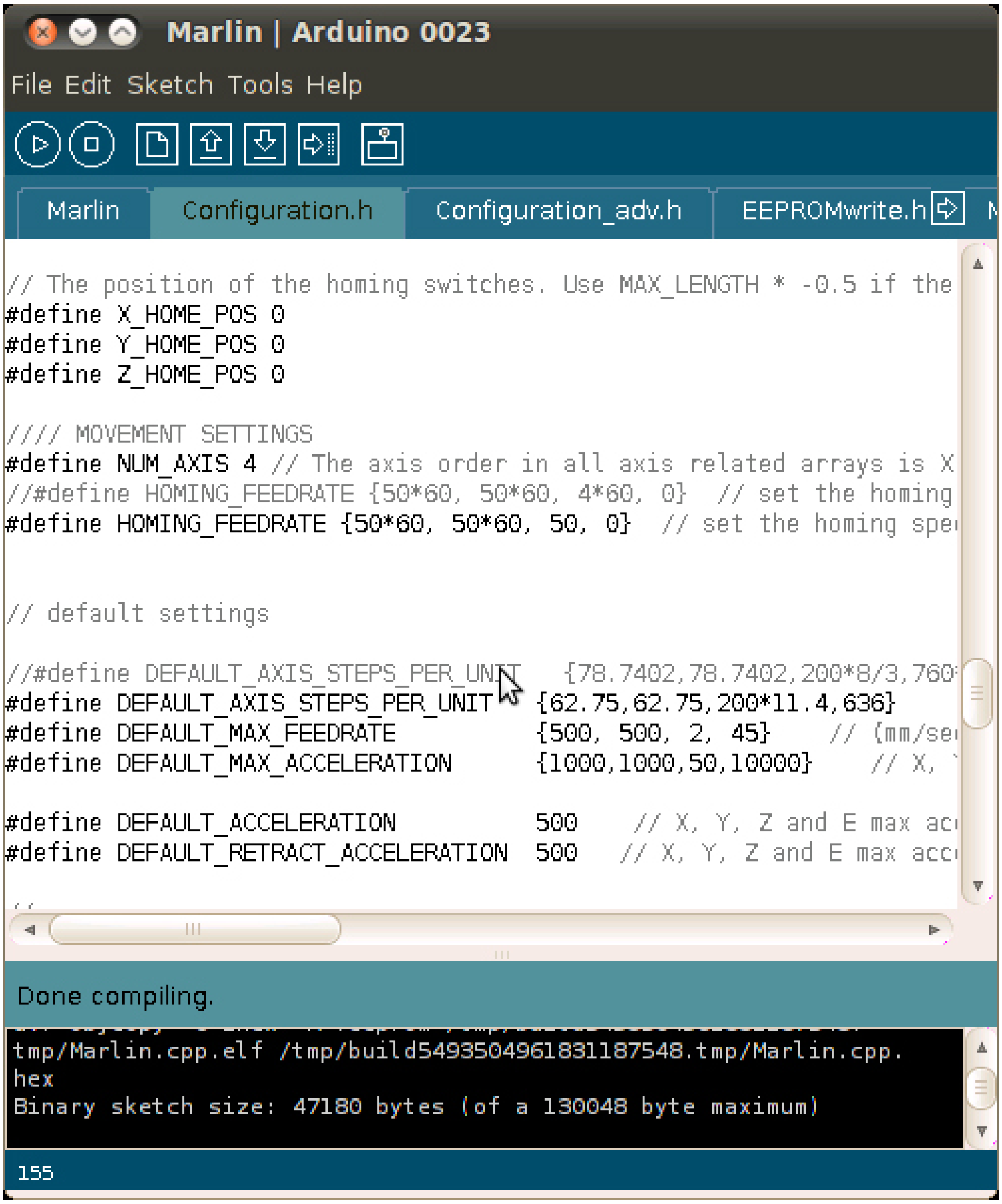
If you are new to 3-D printing, you might assume there's a single piece of software that you download and run, but it turns out that due to how the printers work, you need a few different types of software to manage the printer, including a user interface, a slicer and firmware. Each piece of software performs a specific role, and as you'll see, they all form a sort of logical progression.

Firmware

The firmware is software that runs on electronics directly connected to your printer hardware. This firmware is responsible for controlling the stepper motors and heaters on the printer along with any other electronics, such as any mechanical or optical switches you use as endstops or even fans. The firmware receives instructions over the USB port in the form of G-code—a special language of machine instructions commonly used for CNC machines. The G-code will include instructions to move the printer to specific coordinates, extrude plastic and perform any other hardware functions the printer supports.

Often 3-D printer electronics are Arduino-based, and the firmware as a result is configured with the same software you might use to configure any other Arduino chip. Generally speaking though, you shouldn't have to dig too much into firmware code. There is just a single configuration header file you will need to edit, and only when you need to calibrate your printer. Calibration essentially boils down to telling your printer to do something, such as move 100 millimeters along one axis, measure what the printer actually did, then adjust the numerical settings in the firmware up or down based on the results. Beyond calibration, the firmware will allow you to control stepper motor speeds, acceleration, the size of your print bed and other limits on your printer hardware. Once you have the settings in the firmware calibrated and flash your firmware, you shouldn't need to dig around in the settings much anymore unless you make changes to your hardware.

If you use a MakerBot, your firmware selection is easy, as it has custom firmware. If you use a RepRap, the current most popular firmwares are Sprinter and Marlin. Both are compatible with the most common electronics you'll find on a RepRap, and each has extra features, such as heated build platform and SD card support. I currently use Marlin (Figure 1) as it is the default recommended firmware for my Printrbot's Printrboard. In my case, I needed to patch the default Arduino software so it had Teensylu support, and I needed to install



```
// The position of the homing switches. Use MAX_LENGTH * -0.5 if the
#define X_HOME_POS 0
#define Y_HOME_POS 0
#define Z_HOME_POS 0

//// MOVEMENT SETTINGS
#define NUM_AXIS 4 // The axis order in all axis related arrays is X
//#define HOMING_FEEDRATE {50*60, 50*60, 4*60, 0}  // set the homing
#define HOMING_FEEDRATE {50*60, 50*60, 50, 0}  // set the homing spe


// default settings

//#define DEFAULT_AXIS_STEPS_PER_UNIT   {78.7402,78.7402,200*8/3,760
#define DEFAULT_AXIS_STEPS_PER_UNIT  {62.75,62.75,200*11.4,636}
#define DEFAULT_MAX_FEEDRATE         {500, 500, 2, 45}    // (mm/se
#define DEFAULT_MAX_ACCELERATION     {1000,1000,50,10000}    // X,

#define DEFAULT_ACCELERATION         500    // X, Y, Z and E max ac
#define DEFAULT_RETRACT_ACCELERATION 500   // X, Y, Z and E max acc
```

Done compiling.

```
tmp/Marlin.cpp.elf /tmp/build5493504961831187548.tmp/Marlin.cpp.
hex
Binary sketch size: 47180 bytes (of a 130048 byte maximum)
```

155

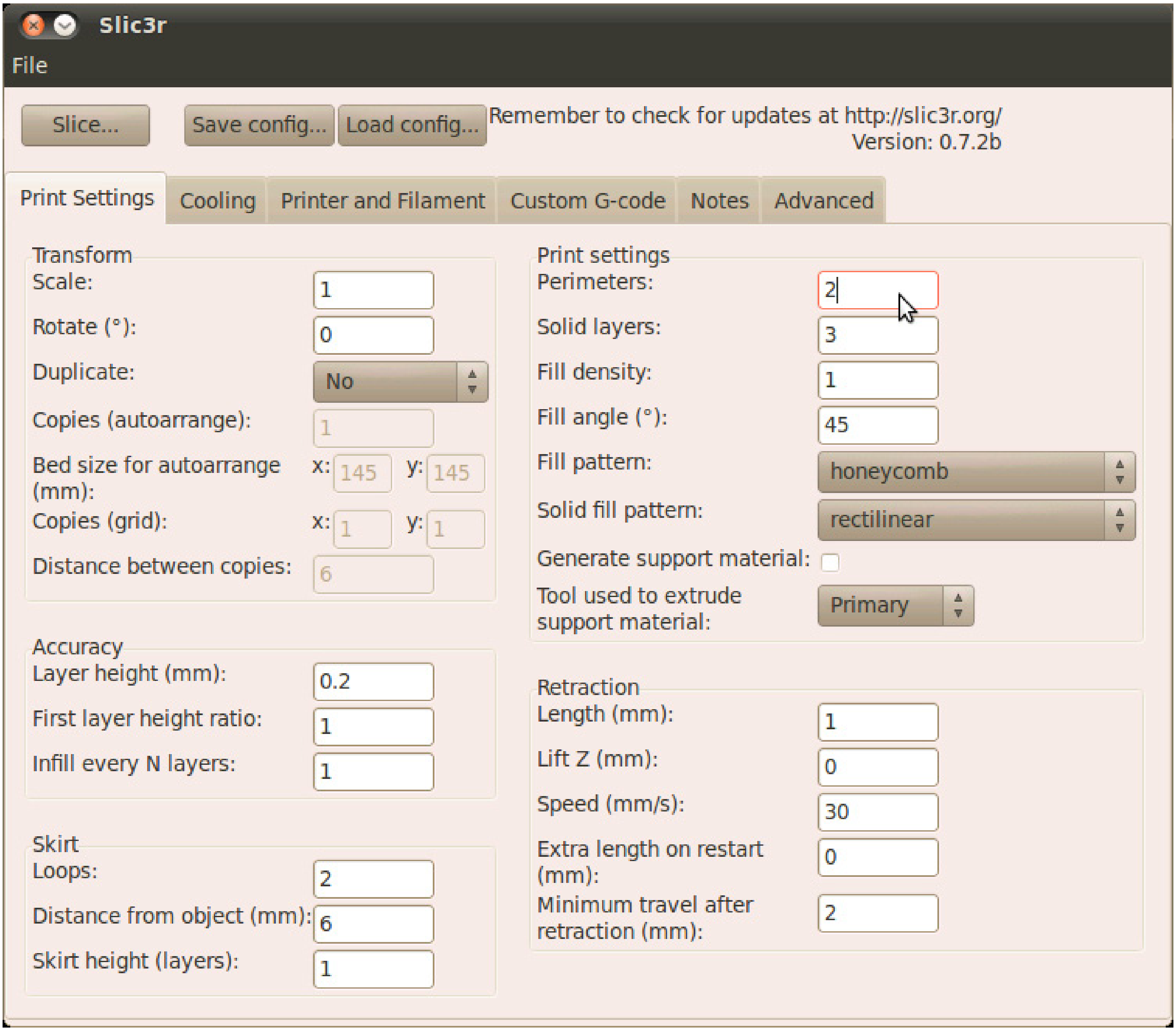
Figure 1. Marlin Configuration with Arduino Software

the dfu-programmer command-line package (which happened to be packaged for Debian-based distros).

Slicers

As I mentioned previously, the firmware accepts G-code as input and does the work of actually controlling the electronics. Generally speaking, when you print something out, you will need to convert some sort of 3-D diagram (usually an STL file) into this G-code though. The program that does this is known as a slicer, because it takes your 3-D diagram and slices it into individual layers of G-code that your printer can print.

Where the firmware settings are more concerned with stepper motors and acceleration settings, the slicer settings are more concerned with filament sizes and other settings you might want to tweak for each individual print. Other settings you control in the slicer include print layer heights, extruder and heated bed temperatures, print speeds, what fill percentage to use for solid parts, fan speeds and other settings that may change from object to object. For instance, you might choose small



Remember to check for updates at http://slic3r.org/
Version: 0.7.2b

Transform
Scale: 1
Rotate (°): 0
Duplicate: No
Copies (autoarrange): 1
Bed size for autoarrange (mm): x: 145 y: 145
Copies (grid): x: 1 y: 1
Distance between copies: 6

Accuracy
Layer height (mm): 0.2
First layer height ratio: 1
Infill every N layers: 1

Skirt
Loops: 2
Distance from object (mm): 6
Skirt height (layers): 1

Print settings
Perimeters: 2
Solid layers: 3
Fill density: 1
Fill angle (°): 45
Fill pattern: honeycomb
Solid fill pattern: rectilinear
Generate support material: □
Tool used to extrude support material: Primary

Retraction
Length (mm): 1
Lift Z (mm): 0
Speed (mm/s): 30
Extra length on restart (mm): 0
Minimum travel after retraction (mm): 2

Figure 2. Slic3r with the Default Print Settings Tab Open

layer heights (like .1mm) and slower print speeds for a very precise print, but for a large bottle opener, you might have a larger layer height and faster print speeds. For parts that need to be more solid, you may pick a higher fill percentage; whereas with parts where rigidity doesn't matter as much, you may pick a lower fill percentage. When printing the same object with either PLA or ABS, you will want to change your extruder and heated bed temperatures to match your material.

The two main slicing programs for Linux are Skeinforge and Slic3r. Skeinforge is included with the ReplicatorG user interface software and has been around longer than Slic3r. Skeinforge is considered to be a reliable slicer, although slow; whereas Slic3r (Figure 2) is much faster than Skeinforge, but it's newer, so it may not be quite as reliable with all STL files, at least not yet.

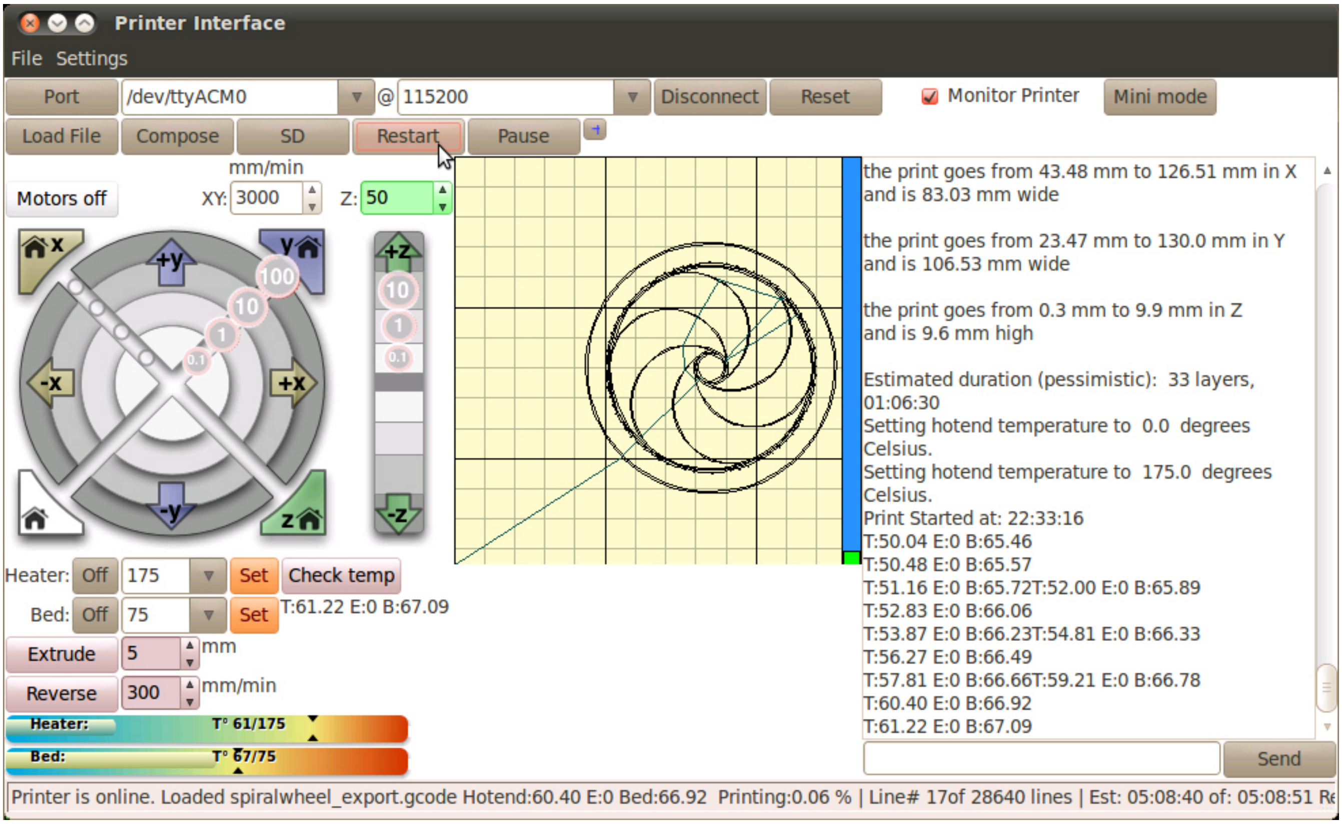
Slic3r is what I personally use with my Printrbot, and the work flow more or less is like this: I select what I want to print, and depending on whether I feel it needs slower speeds, more cooling or a smaller layer height, I tweak those settings in Slic3r and save them. Then, I go to my user interface software to run Slic3r and print the object. I also may tweak the settings whenever I switch plastic filament, as different filaments need different extrusion temperatures and have slightly different thicknesses. Slic3r calculates just how much plastic to extrude based on your filament thickness, so even if your printer uses 3mm filament, you might discover the actual diameter is 2.85mm. Slic3r also can create multiples of a particular item or scale an item up or down in size via its settings.

User Interface

At the highest level is a program that acts as a user interface for the printer. This software communicates with the printer over a serial interface (although most printers connect to the computer over a USB cable) and provides either a command-line or graphical interface you can use to move the printer along its axes and home it, control the temperature for extrusion or a heated bed (if you have one, it can be handy to help the first layer of the print stick to the print bed) and send G-code files to the printer.

The two most popular graphical user interfaces are ReplicatorG and Pronterface (part of the Printrun suite of software). ReplicatorG has been around longer, but Pronterface seems more popular today with the RepRap community. Generally, the user interface doesn't slice STL files itself but instead hands that off to another program. For instance, ReplicatorG uses Skeinforge as its slicer, and Pronterface defaults to Skeinforge but can also use Slic3r. Once the slicer generates the G-code, the user interface then sends that G-code to the printer and monitors its progress. In my case, I use Pronterface set to use Slic3r.

In Figure 3, you can see an example of Pronterface's GUI. On the left side of the window is a set of controls I can use to control my printer manually, so I can move it around each axis, extrude filament and manually set temperature settings. In the middle of the screen is a preview grid where I can see the object I've loaded, and during a print, I can see a particular slice. On the right side is an output section that tells me how much filament a print will use, approximately how long it might take to print and a place where I can send manual G-code commands. Finally, along the bottom is an area that displays the current status of a print,



Port: /dev/ttyACM0 @ 115200

XY: 3000 Z: 50

Heater: Off 175 Set

Bed: Off 75 Set

T:61.22 E:0 B:67.09

Extrude 5 mm

Reverse 300 mm/min

Heater: T° 61/175

Bed: T° 67/75

```
the print goes from 43.48 mm to 126.51 mm in X
and is 83.03 mm wide

the print goes from 23.47 mm to 130.0 mm in Y
and is 106.53 mm wide

the print goes from 0.3 mm to 9.9 mm in Z
and is 9.6 mm high

Estimated duration (pessimistic):  33 layers,
01:06:30
Setting hotend temperature to  0.0  degrees
Celsius.
Setting hotend temperature to  175.0  degrees
Celsius.
Print Started at: 22:33:16
T:50.04 E:0 B:65.46
T:50.48 E:0 B:65.57
T:51.16 E:0 B:65.72T:52.00 E:0 B:65.89
T:52.83 E:0 B:66.06
T:53.87 E:0 B:66.23T:54.81 E:0 B:66.33
T:56.27 E:0 B:66.49
T:57.81 E:0 B:66.66T:59.21 E:0 B:66.78
T:60.40 E:0 B:66.92
T:61.22 E:0 B:67.09
```

```
Printer is online. Loaded spiralwheel_export.gcode Hotend:60.40 E:0 Bed:66.92  Printing:0.06 % | Line# 17of 28640 lines | Est: 05:08:40 of: 05:08:51 Re
```

Figure 3. Pronterface's GUI

including my temperature settings and how far along it is in the print job.

I generally make my print job settings in Slic3r, save them, then go to Pronterface where I will load an STL file I want to print. Pronterface then calls Slic3r behind the scenes to generate the G-code. Once the file has been sliced, I click on the Print button, which sends the G-code to the printer. The G-code includes initial instructions to heat up the extruder and heated bed to a certain temperature before homing the printer and then starting the print. Then as the print starts, I just use Pronterface to keep an eye on the progress.

Although I expect you'll still need to do plenty of experimentation and research to choose a 3-D printer and use it effectively, after reading these articles, you should have a better idea of what 3-D printers and software are available and whether it is something you want to pursue. Like with Linux distributions, there really isn't a right 3-D printer and software suite for everyone, but hopefully, you should be able to find a combination of hardware and software that fits your needs and tastes.■

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

SHAWN POWERS

Webmin—the Sysadmin Gateway Drug

Manage your Linux server without ever touching the command line.

Whenever I introduce people to Linux, the first thing they bring up is how scary the command line is. Personally, I'm more disturbed by *not* having a command line to work with, but I understand a CLI can be intimidating. Thankfully, not only do many distributions offer GUI tools for some of their services, but Webmin also is available to configure almost every aspect of your server from the comfort of a GUI Web browser.

I have to be honest, many people dislike Webmin. They claim it is messy, or that it doesn't handle underlying services well, or that the whole concept of root-level access over a Web browser is too insecure. Some of those concerns are quite valid, but I think the benefits outweigh the risks, at least in many circumstances.

What Is Webmin?

Like the name implies, Webmin is a Web-based administration tool for Linux. It also supports UNIX, OS X and possibly even Windows, but I've only ever used it with Linux. At the core, Webmin is a dæmon process that provides a framework for modules. Those modules, in turn, offer a Web-based GUI for configuring and interacting with dæmons running on the underlying server. Modules also can be used to interact with user management, system backups and pretty much anything else a user with root access might want to control.

Webmin comes with a huge number of built-in modules that can manage a large selection of common server tasks. The infrastructure is such that authors also can write their own modules or download third-party contributed modules. With the nature of Webmin's root permissions, third-party modules can be a scary notion, so it's unwise to install them willy-nilly.

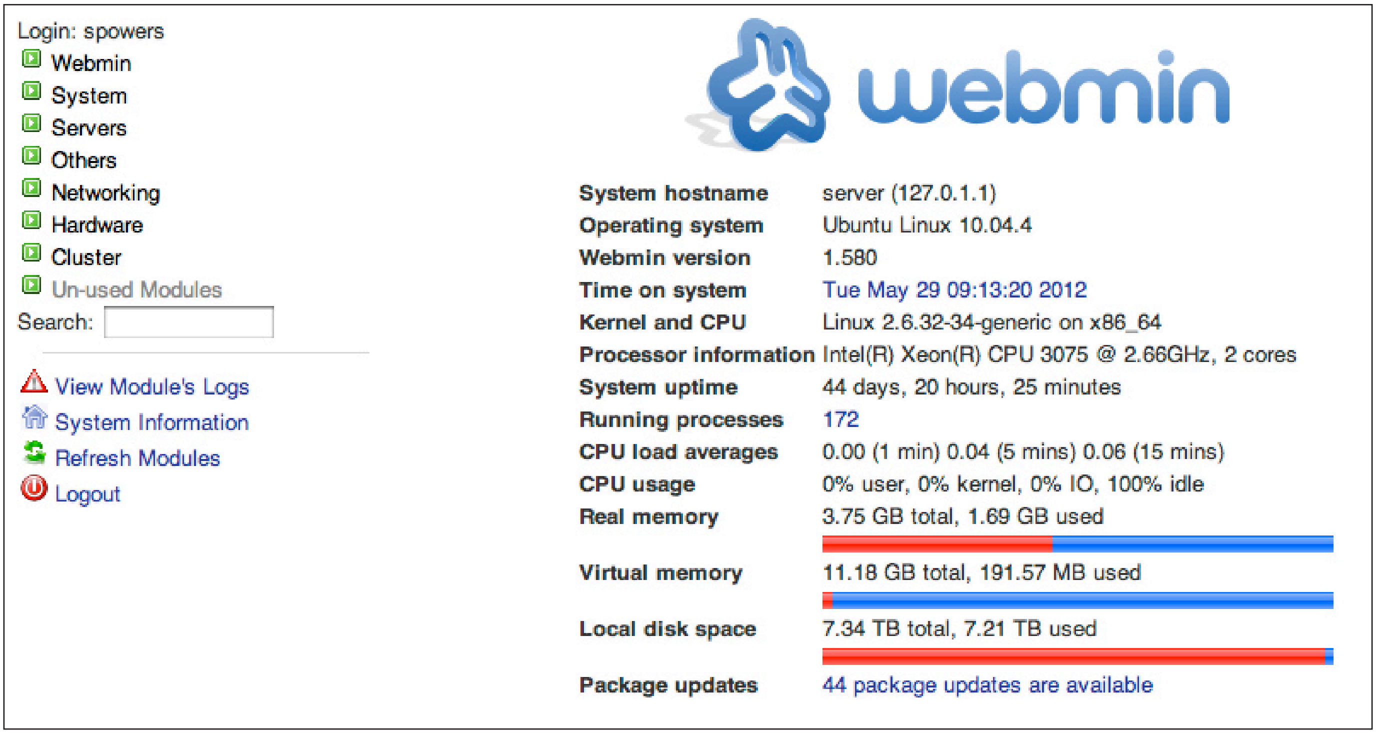
Installation

The Webmin installation instructions are on its Web site: **http://www.webmin.com**. You can download an RPM or deb file if your distribution supports it, but Webmin also supplies a tarball along with installation instructions for most systems. If you use the RPM or deb files, I highly recommend installing the APT or YUM repository rather than directly installing the downloaded package. Not only will that allow for dependency resolution, but it also means updates will occur with your system updates.

If you use the tarball for installation, the setup.sh script will walk you through all the configuration settings. This is the proper way to install Webmin for Linux distributions like Slackware, which don't support RPM or deb files. Be sure during the configuration process that you select your specific distribution, otherwise Webmin won't handle the config files for your various services properly.

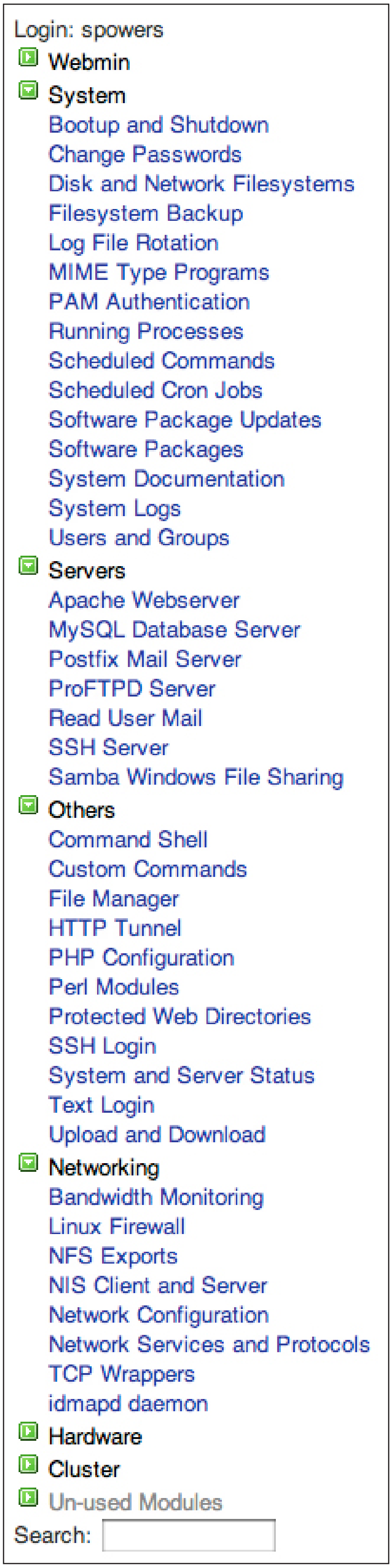
What's the Secret Sauce?

The thing I've always liked about Webmin is the lack of magic. The underlying configuration files on your system are configured using the appropriate syntax and can be edited by hand if you prefer. In fact, if you already have configured services on your server, Webmin usually will read the configuration properly.



System hostname	server (127.0.1.1)
Operating system	Ubuntu Linux 10.04.4
Webmin version	1.580
Time on system	Tue May 29 09:13:20 2012
Kernel and CPU	Linux 2.6.32-34-generic on x86_64
Processor information	Intel(R) Xeon(R) CPU 3075 @ 2.66GHz, 2 cores
System uptime	44 days, 20 hours, 25 minutes
Running processes	172
CPU load averages	0.00 (1 min) 0.04 (5 mins) 0.06 (15 mins)
CPU usage	0% user, 0% kernel, 0% IO, 100% idle
Real memory	3.75 GB total, 1.69 GB used
Virtual memory	11.18 GB total, 191.57 MB used
Local disk space	7.34 TB total, 7.21 TB used
Package updates	44 package updates are available

Figure 1. The dashboard is simple, but quite useful.



Login: spowers
Webmin
System
Bootup and Shutdown
Change Passwords
Disk and Network Filesystems
Filesystem Backup
Log File Rotation
MIME Type Programs
PAM Authentication
Running Processes
Scheduled Commands
Scheduled Cron Jobs
Software Package Updates
Software Packages
System Documentation
System Logs
Users and Groups
Servers
Apache Webserver
MySQL Database Server
Postfix Mail Server
ProFTPD Server
Read User Mail
SSH Server
Samba Windows File Sharing
Others
Command Shell
Custom Commands
File Manager
HTTP Tunnel
PHP Configuration
Perl Modules
Protected Web Directories
SSH Login
System and Server Status
Text Login
Upload and Download
Networking
Bandwidth Monitoring
Linux Firewall
NFS Exports
NIS Client and Server
Network Configuration
Network Services and Protocols
TCP Wrappers
idmapd daemon
Hardware
Cluster
Un-used Modules
Search:

Figure 2. The sheer number of Webmin modules is overwhelming, but awesome.

Sometimes it's a great way to learn the proper method for configuring a particular service by configuring it with Webmin and then looking at what changes were made to the config files. This is helpful if you can't remember (or don't want to be bothered with researching) the particular syntax. I've learned some pretty cool things about configuring virtual hosts in Apache by looking at how Webmin sets them up.

It's important to note that Webmin can be configured to work over non-encrypted HTTP, but because very sensitive data (including a user account with root access!) is transmitted via browser, SSL is enabled and forced by default. This means annoyance with unsigned certificates at first, but using standard HTTP is simply a horrible idea.

So What Does It Do?

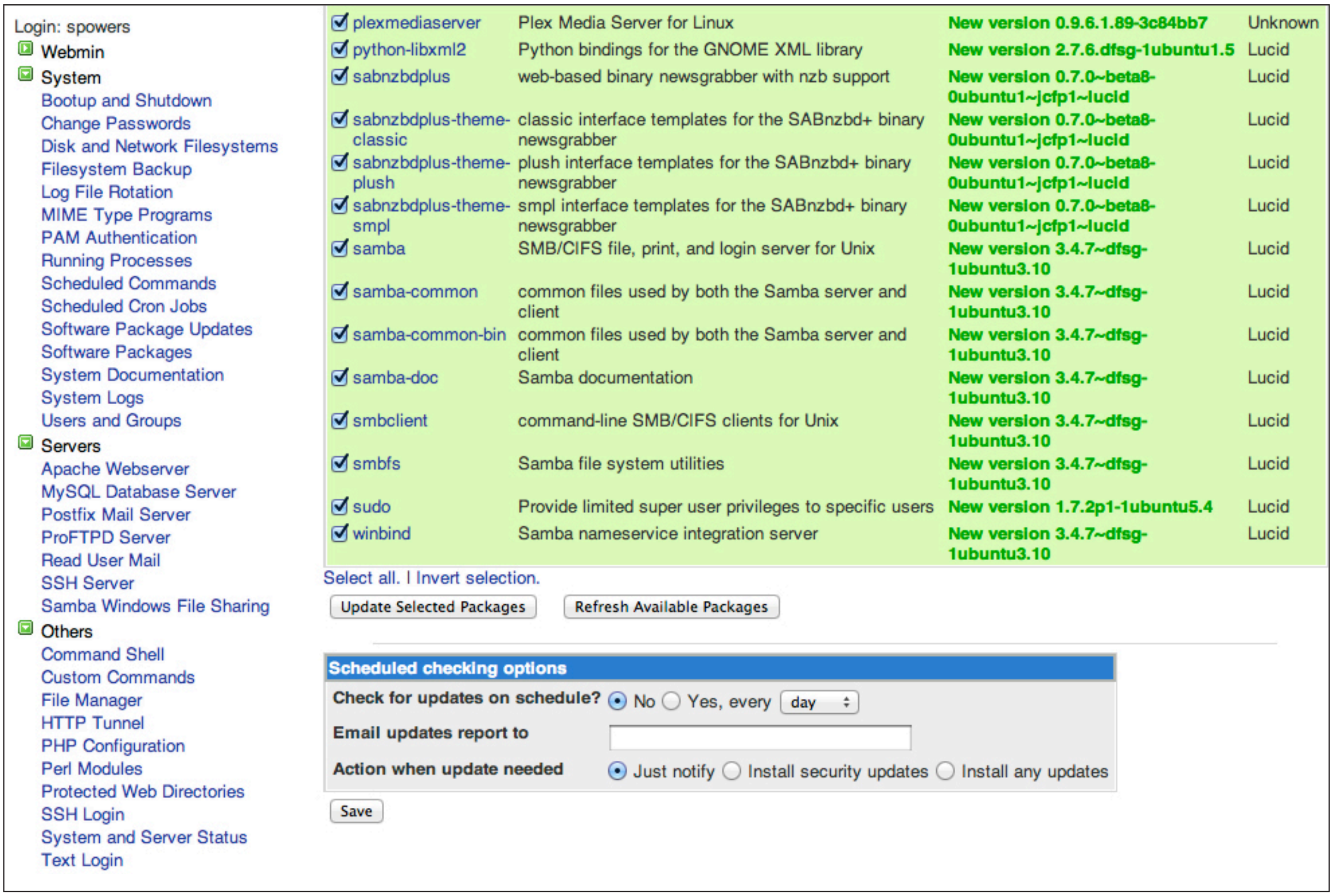
Once Webmin is installed, it should detect installed applications on your server and enable the appropriate modules. To log in, point your browser to https://server.ip.address:10000, and log in with either the root account or a user with full sudo privileges. The latter is preferable, as typing a root user/password into a Web form just gives me the willies.

The first page you'll see is a dashboard of sorts. Figure 1 shows the details of my home server. It's been 44 days since our last extended power outage (my uptime); I have some packages to update, and my file server is almost full. The dashboard doesn't offer earth-shattering information, but it's a nice collection of quick stats. The notification about 44 package updates available also is a hyperlink, which leads to the apt module. It makes for a very simple point-and-click way to keep your system updated.

Along the left side of the dashboard, you'll notice expandable menus separated into subject areas. I've never really liked the categories in Webmin, because so many modules naturally fit into more than one. Still, I appreciate the attempt at organization, and I just search the menus until I find the module I'm looking for. Figure 2 shows a mostly expanded screenshot of the menu system. These are merely the services and features Webmin detected when it was installed. There is still the "Un-used Modules" menu, which contains countless other modules for applications I don't have installed.

The Mounds of Modules

Going back to those packages that need to be updated, clicking on the "Software Package Updates" module (or just



☑ plexmediaserver	Plex Media Server for Linux	New version 0.9.6.1.89-3c84bb7	Unknown
☑ python-libxml2	Python bindings for the GNOME XML library	New version 2.7.6.dfsg-1ubuntu1.5	Lucid
☑ sabnzbdplus	web-based binary newsgrabber with nzb support	New version 0.7.0~beta8-0ubuntu1~jcfp1~lucid	Lucid
☑ sabnzbdplus-theme-classic	classic interface templates for the SABnzbd+ binary newsgrabber	New version 0.7.0~beta8-0ubuntu1~jcfp1~lucid	Lucid
☑ sabnzbdplus-theme-plush	plush interface templates for the SABnzbd+ binary newsgrabber	New version 0.7.0~beta8-0ubuntu1~jcfp1~lucid	Lucid
☑ sabnzbdplus-theme-smpl	smpl interface templates for the SABnzbd+ binary newsgrabber	New version 0.7.0~beta8-0ubuntu1~jcfp1~lucid	Lucid
☑ samba	SMB/CIFS file, print, and login server for Unix	New version 3.4.7~dfsg-1ubuntu3.10	Lucid
☑ samba-common	common files used by both the Samba server and client	New version 3.4.7~dfsg-1ubuntu3.10	Lucid
☑ samba-common-bin	common files used by both the Samba server and client	New version 3.4.7~dfsg-1ubuntu3.10	Lucid
☑ samba-doc	Samba documentation	New version 3.4.7~dfsg-1ubuntu3.10	Lucid
☑ smbclient	command-line SMB/CIFS clients for Unix	New version 3.4.7~dfsg-1ubuntu3.10	Lucid
☑ smbfs	Samba file system utilities	New version 3.4.7~dfsg-1ubuntu3.10	Lucid
☑ sudo	Provide limited super user privileges to specific users	New version 1.7.2p1-1ubuntu5.4	Lucid
☑ winbind	Samba nameservice integration server	New version 3.4.7~dfsg-1ubuntu3.10	Lucid

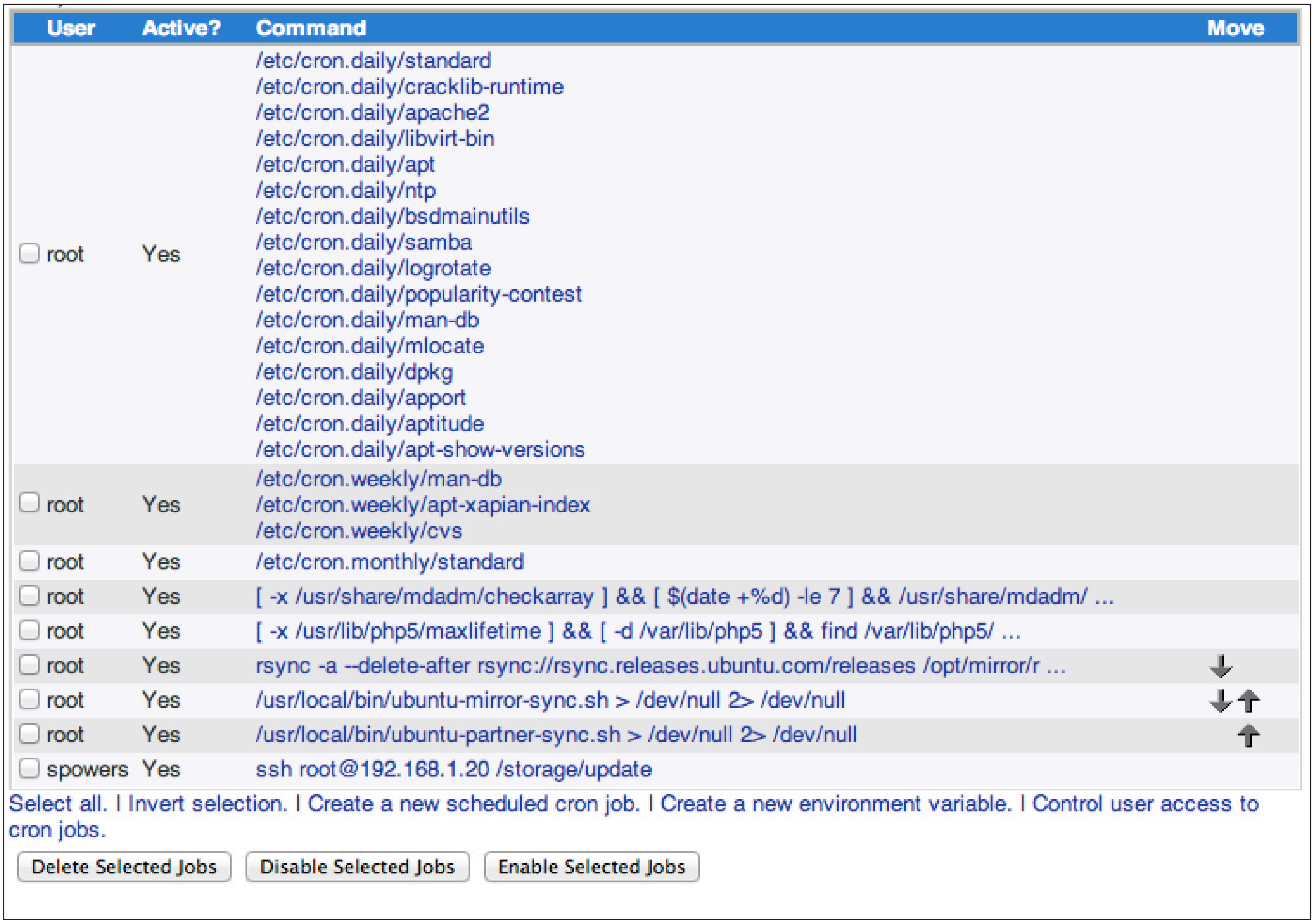
Select all. | Invert selection.
Update Selected Packages
Refresh Available Packages
Scheduled checking options

Check for updates on schedule? ● No ○ Yes, every day

Email updates report to

Action when update needed ● Just notify ○ Install security updates ○ Install any updates

Figure 3. A GUI tool for updates on a headless server is very nice.



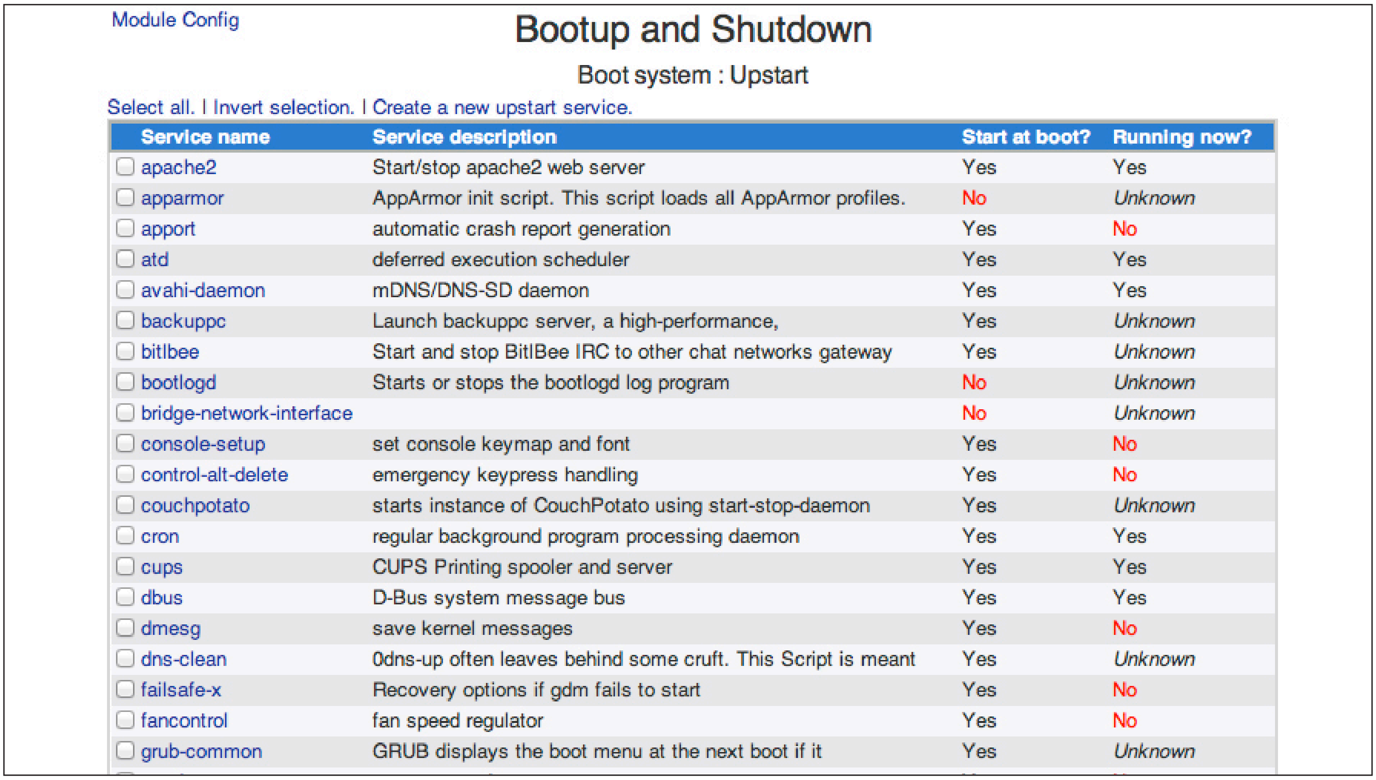
User	Active?	Command	Move
□ root	Yes	/etc/cron.daily/standard /etc/cron.daily/cracklib-runtime /etc/cron.daily/apache2 /etc/cron.daily/libvirt-bin /etc/cron.daily/apt /etc/cron.daily/ntp /etc/cron.daily/bsdmainutils /etc/cron.daily/samba /etc/cron.daily/logrotate /etc/cron.daily/popularity-contest /etc/cron.daily/man-db /etc/cron.daily/mlocate /etc/cron.daily/dpkg /etc/cron.daily/apport /etc/cron.daily/aptitude /etc/cron.daily/apt-show-versions	
□ root	Yes	/etc/cron.weekly/man-db /etc/cron.weekly/apt-xapian-index /etc/cron.weekly/cvs	
□ root	Yes	/etc/cron.monthly/standard	
□ root	Yes	[-x /usr/share/mdadm/checkarray] && [$(date +%d) -le 7] && /usr/share/mdadm/ ...	
□ root	Yes	[-x /usr/lib/php5/maxlifetime] && [-d /var/lib/php5] && find /var/lib/php5/ ...	
□ root	Yes	rsync -a --delete-after rsync://rsync.releases.ubuntu.com/releases /opt/mirror/r ...	↓
□ root	Yes	/usr/local/bin/ubuntu-mirror-sync.sh > /dev/null 2> /dev/null	↓↑
□ root	Yes	/usr/local/bin/ubuntu-partner-sync.sh > /dev/null 2> /dev/null	↑
□ spowers	Yes	ssh root@192.168.1.20 /storage/update	

Select all. | Invert selection. | Create a new scheduled cron job. | Create a new environment variable. | Control user access to cron jobs.
Delete Selected Jobs
Disable Selected Jobs
Enable Selected Jobs

Figure 4. Cron jobs are simple to edit with Webmin.

clicking the hyperlink on the dashboard) will give you a listing of the outdated packages. Figure 3 shows my system. I've scrolled down to the bottom of the list to show some of the little extras Webmin offers. There is a button to refresh the package list, which upon clicking would execute `sudo apt-get update` in the background and then refresh the page with whatever updates are available. The same sort of thing happens when pressing the "Update Selected Packages" button; it just offers a quick-and-clicky way to run `apt-get update`. Below those buttons, you can see a nifty scheduling option for installing updates automatically. Like most things with Webmin, this isn't some proprietary scheduler, it simply runs cron jobs in the underlying system.

Other common system configuration tasks are available as modules too. Figure 4 shows the crontab configuration tool. Figure 5 shows the upstart configuring (which dæmons are started on boot), and Figure 6 shows the interface for viewing log files. All of these things are configurable from the command line, but



Module Config

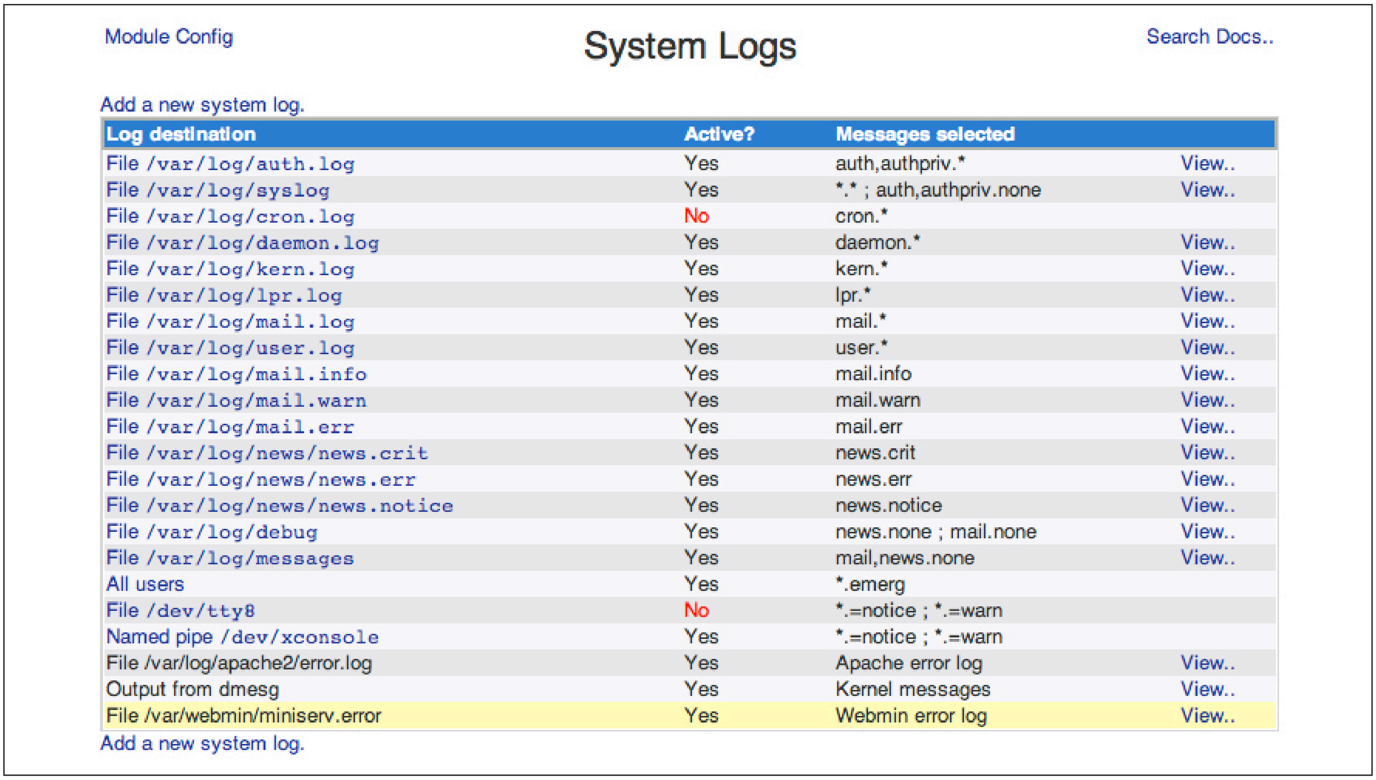
Bootup and Shutdown

Boot system : Upstart

Select all. | Invert selection. | Create a new upstart service.

Service name	Service description	Start at boot?	Running now?
□ apache2	Start/stop apache2 web server	Yes	Yes
□ apparmor	AppArmor init script. This script loads all AppArmor profiles.	No	*Unknown*
□ apport	automatic crash report generation	Yes	No
□ atd	deferred execution scheduler	Yes	Yes
□ avahi-daemon	mDNS/DNS-SD daemon	Yes	Yes
□ backuppc	Launch backuppc server, a high-performance,	Yes	*Unknown*
□ bitlbee	Start and stop BitlBee IRC to other chat networks gateway	Yes	*Unknown*
□ bootlogd	Starts or stops the bootlogd log program	No	*Unknown*
□ bridge-network-interface		No	*Unknown*
□ console-setup	set console keymap and font	Yes	No
□ control-alt-delete	emergency keypress handling	Yes	No
□ couchpotato	starts instance of CouchPotato using start-stop-daemon	Yes	*Unknown*
□ cron	regular background program processing daemon	Yes	Yes
□ cups	CUPS Printing spooler and server	Yes	Yes
□ dbus	D-Bus system message bus	Yes	Yes
□ dmesg	save kernel messages	Yes	No
□ dns-clean	0dns-up often leaves behind some cruft. This Script is meant	Yes	*Unknown*
□ failsafe-x	Recovery options if gdm fails to start	Yes	No
□ fancontrol	fan speed regulator	Yes	No
□ grub-common	GRUB displays the boot menu at the next boot if it	Yes	*Unknown*

Figure 5. It got confusing when Ubuntu switched to upstart from sysv, but Webmin handles it just fine.



Module Config

System Logs

Search Docs..

Add a new system log.

Log destination	Active?	Messages selected	
File `/var/log/auth.log`	Yes	auth,authpriv.*	View..
File `/var/log/syslog`	Yes	*.* ; auth,authpriv.none	View..
File `/var/log/cron.log`	No	cron.*	
File `/var/log/daemon.log`	Yes	daemon.*	View..
File `/var/log/kern.log`	Yes	kern.*	View..
File `/var/log/lpr.log`	Yes	lpr.*	View..
File `/var/log/mail.log`	Yes	mail.*	View..
File `/var/log/user.log`	Yes	user.*	View..
File `/var/log/mail.info`	Yes	mail.info	View..
File `/var/log/mail.warn`	Yes	mail.warn	View..
File `/var/log/mail.err`	Yes	mail.err	View..
File `/var/log/news/news.crit`	Yes	news.crit	View..
File `/var/log/news/news.err`	Yes	news.err	View..
File `/var/log/news/news.notice`	Yes	news.notice	View..
File `/var/log/debug`	Yes	news.none ; mail.none	View..
File `/var/log/messages`	Yes	mail,news.none	View..
All users	Yes	*.emerg	
File `/dev/tty8`	No	*.=notice ; *.=warn	
Named pipe `/dev/xconsole`	Yes	*.=notice ; *.=warn	
File /var/log/apache2/error.log	Yes	Apache error log	View..
Output from dmesg	Yes	Kernel messages	View..
File /var/webmin/miniserv.error	Yes	Webmin error log	View..

Add a new system log.

Figure 6. Not only can you view logs, you can manage them as well.

the simple, consistent interface can be a time-saver, especially for folks unfamiliar with configuring the different aspects of their system.

Servicing Servers

I've been a sysadmin for 17+ years, and I still need to search the manual in order to get Apache configuration directives right. I think it's very good for sysadmins to know how programs like Apache work, but I also think it's nice to have a tool like the Webmin module (Figure 7) to make changes. Whether you need to add a virtual host or want to configure global cgi-bin permissions, Webmin is a quick way to get the right syntax in the right place.

The MySQL Module, shown in Figure 8, is a very functional alternative to both the command-line MySQL interface and the popular phpmyadmin package. I've found it to be a little less robust than phpmyadmin, but it has the convenience of being contained within the Webmin system.

I won't list every service available, but here are a few of the really handy ones:

- SSH Server: great for managing user access and system authentication keys.

- Postfix/Sendmail: e-mail can be tricky to configure, but the GUI interface makes it simple.

- Samba: there are a few other Web-based Samba configuration tools, but Webmin is very functional and straightforward.

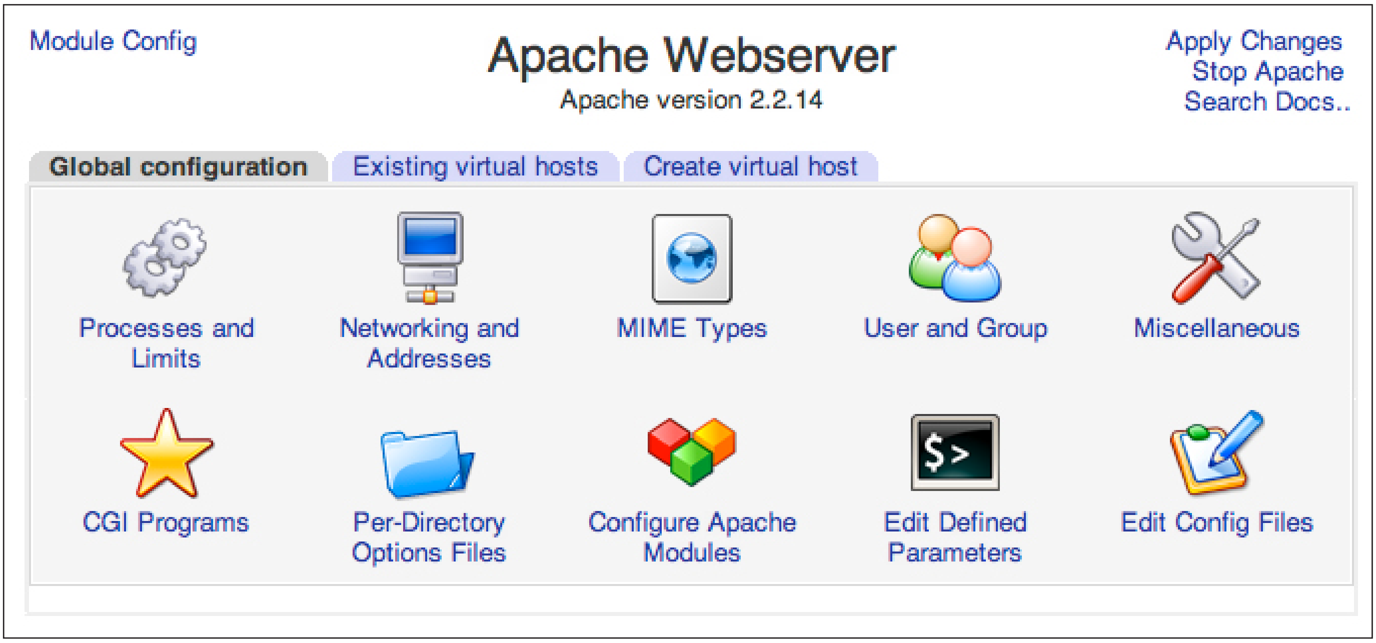
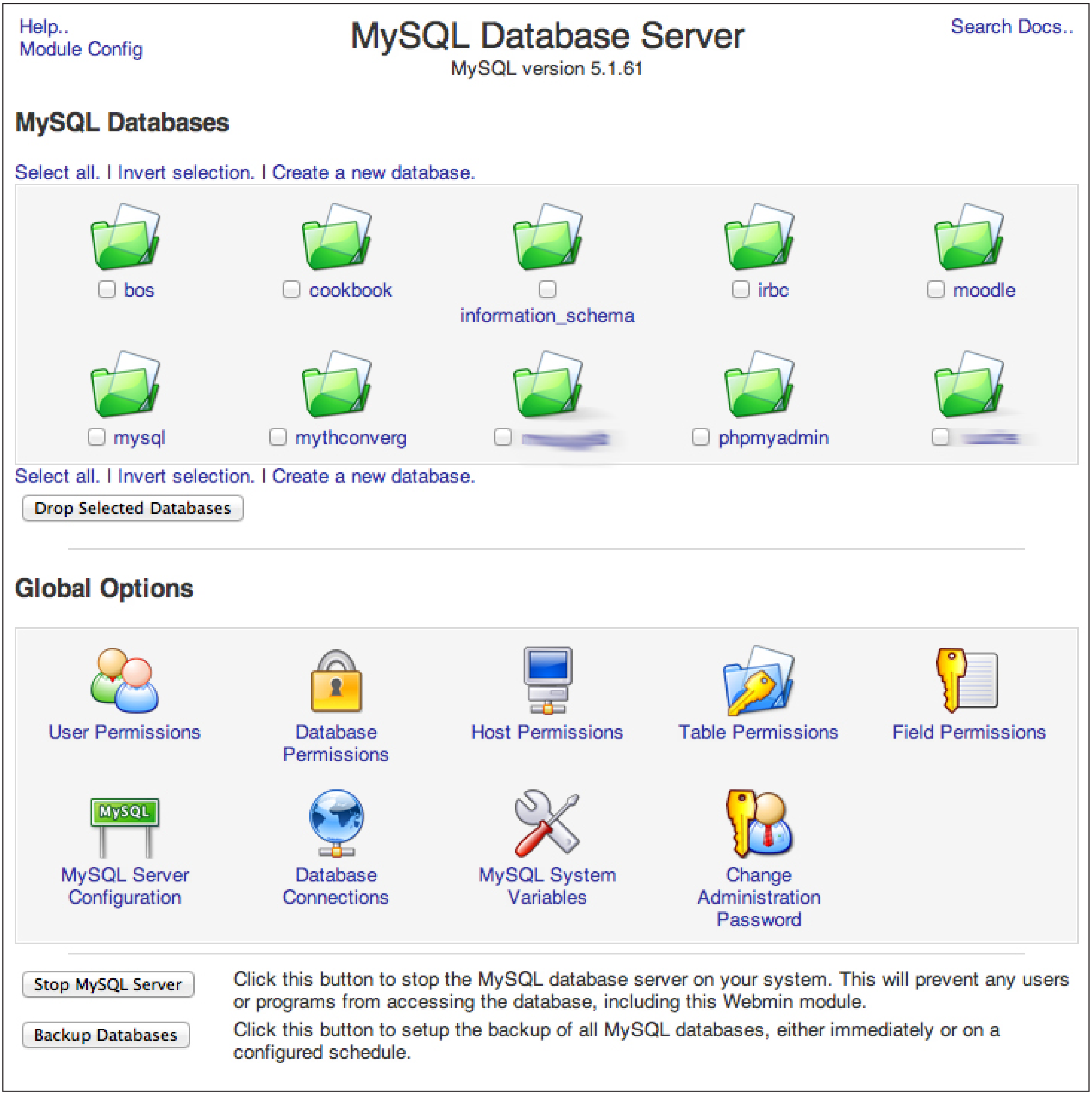


Figure 7. Apache has so many options, keeping track of them can be like herding cats. Webmin helps.



Click this button to stop the MySQL database server on your system. This will prevent any users or programs from accessing the database, including this Webmin module.

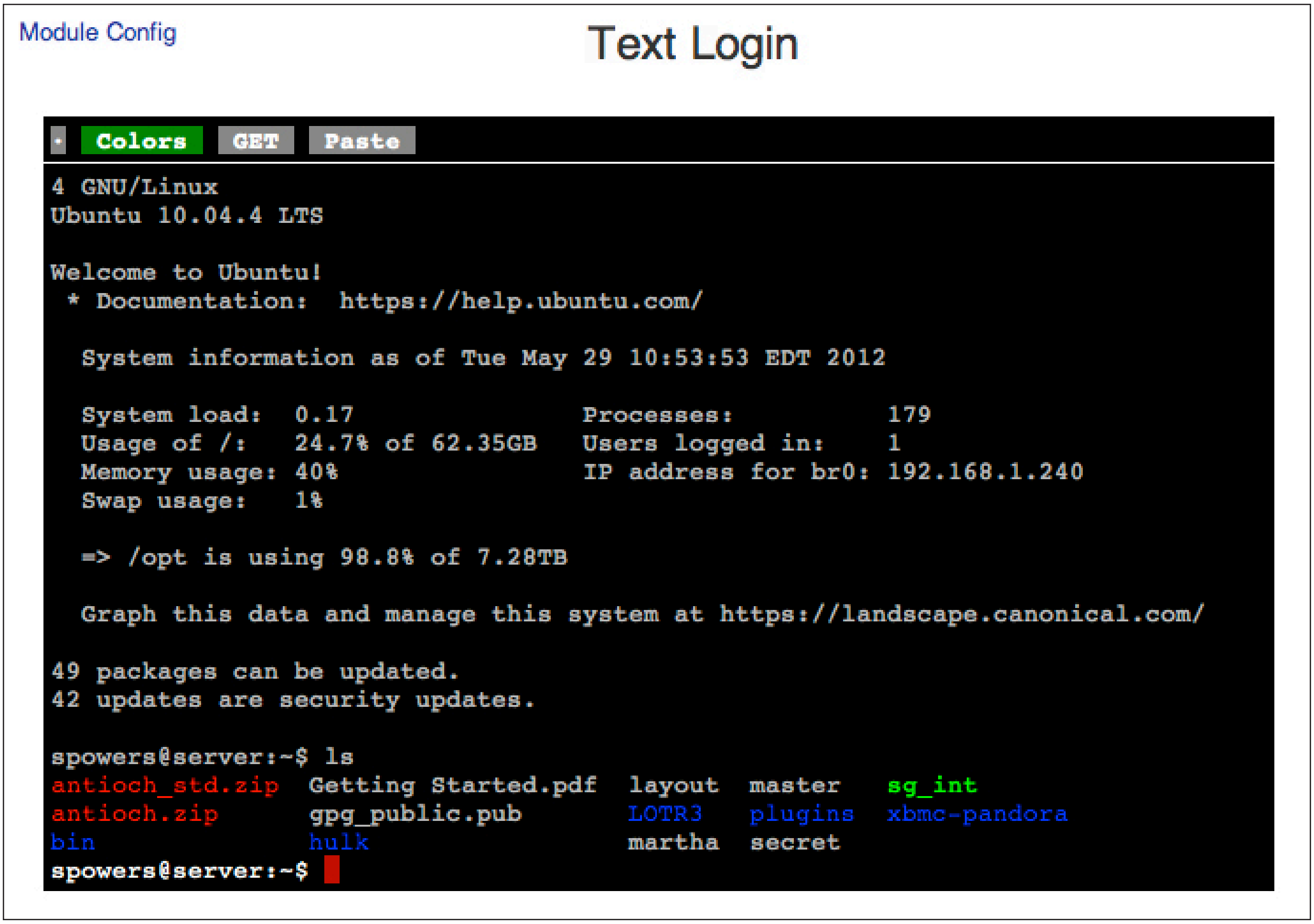
Click this button to setup the backup of all MySQL databases, either immediately or on a configured schedule.

Figure 8. The MySQL module is very functional, with a consistent interface.

When Configuration Isn't Enough

It's clear that Webmin is a powerful and convenient tool for system configuration. However, some other features are just as useful.

If you look back at Figure 2, you'll notice a bunch of modules in the "Others" section. Most are fairly straightforward, like the File Manager. Modules like the Java-based



```
4 GNU/Linux
Ubuntu 10.04.4 LTS

Welcome to Ubuntu!
 * Documentation:  https://help.ubuntu.com/

  System information as of Tue May 29 10:53:53 EDT 2012

  System load:  0.17               Processes:           179
  Usage of /:   24.7% of 62.35GB   Users logged in:     1
  Memory usage: 40%                IP address for br0: 192.168.1.240
  Swap usage:   1%

  => /opt is using 98.8% of 7.28TB

  Graph this data and manage this system at https://landscape.canonical.com/

49 packages can be updated.
42 updates are security updates.

spowers@server:~$ ls
antioch_std.zip  Getting Started.pdf  layout  master   sg_int
antioch.zip      gpg_public.pub       LOTR3   plugins  xbmc-pandora
bin              hulk                 martha  secret
spowers@server:~$
```

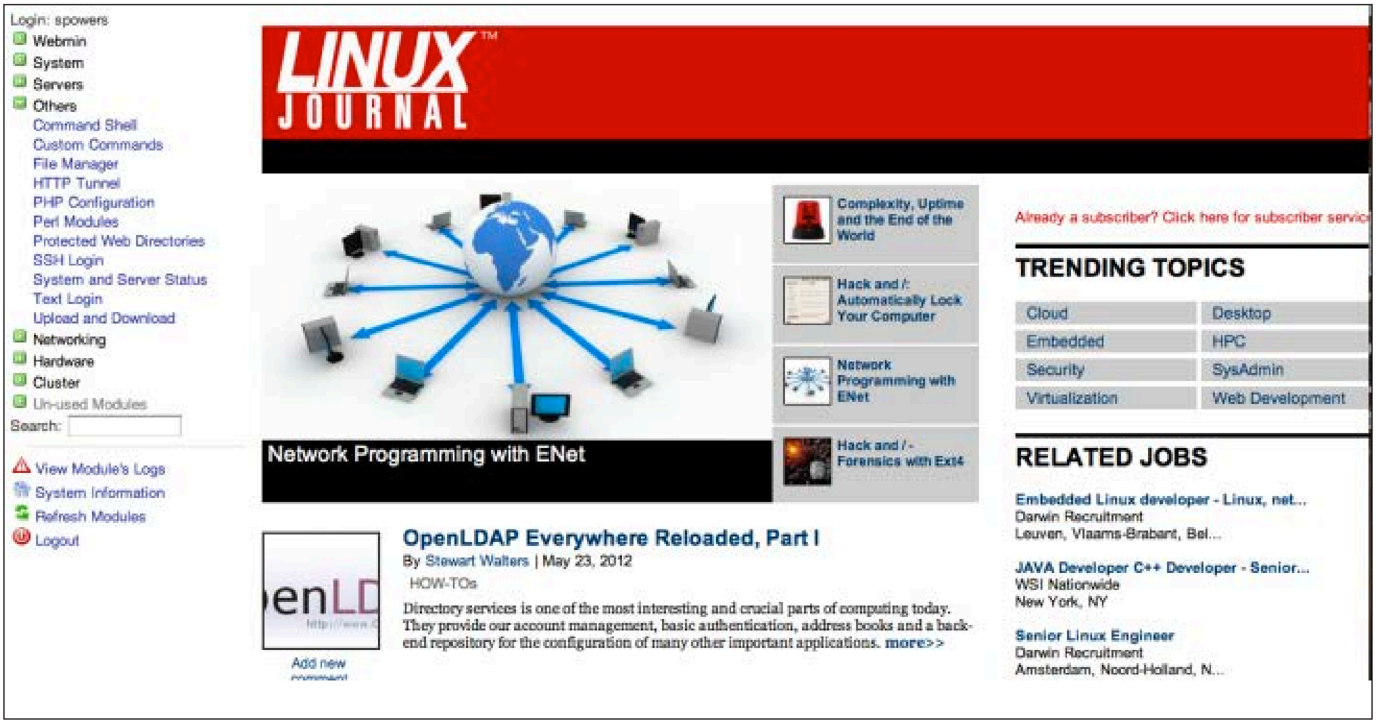
Figure 9. The command line in a browser is helpful in a pinch, but too slow for regular use.

SSH Login or the AJAX-based Text Login are very useful if you need to get to a command line on your server, but don't have access to a terminal (like when you are on your uncle's Windows 98 machine at Thanksgiving dinner and your server crashes, but that's another story).

Another nifty module is the HTTP Tunnel tool (Figure 10), which allows you to browse the Web through a tunnel. This certainly could be used for nefarious purposes if you're trying to get around a Web filter, but it has righteous value as well. Whether you're testing connectivity from a remote site or avoiding geographic restrictions while abroad, the HTTP Tunnel module can be a life-saver.

When Webmin Isn't Enough!

If you were thinking how great Webmin is for the sysadmin, but you wish there were something end users could use for managing their accounts, you're in luck. Usermin is a separate program that runs on the server and allows users to log in



OpenLDAP Everywhere Reloaded, Part I

By Stewart Walters | May 23, 2012

HOW-TOs

Directory services is one of the most interesting and crucial parts of computing today. They provide our account management, basic authentication, address books and a back-end repository for the configuration of many other important applications. more>>

Embedded Linux developer - Linux, net...
Darwin Recruitment
Leuven, Vlaams-Brabant, Bel...

JAVA Developer C++ Developer - Senior...
WSI Nationwide
New York, NY

Senior Linux Engineer
Darwin Recruitment
Amsterdam, Noord-Holland, N...

Figure 10. The HTTP Tunnel is a cool feature, but it can be slow if you have a slow Internet connection on your server.

and configure items specific to their accounts. If users need to set up their .forward file or create a procmail recipe for sorting incoming mail, Usermin has modules to support that. It will allow users to configure their .htaccess files for Apache, change their passwords, edit their cron jobs and even manage their own MySQL databases. Usermin basically takes the concept of Webmin and applies it to the individual user. Oh, and how do you configure the Usermin dæmon? There's a Webmin module for that!

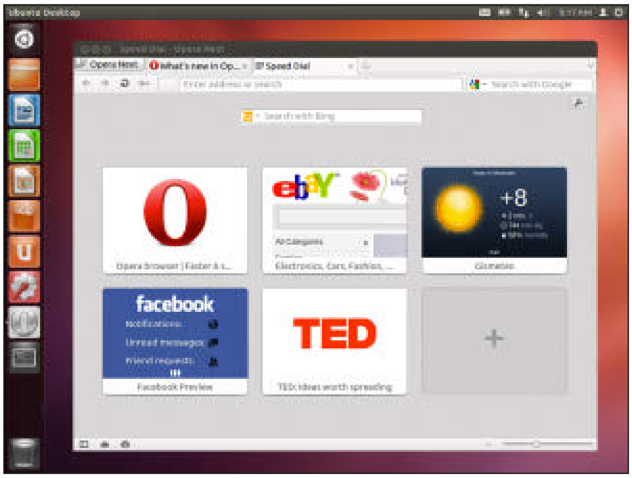
Webmin is a tool that people either love or hate. Some people are offended by the transmission of root-level information over a browser, and some people think the one-stop shop for system maintenance is unbeatable. I'm a teacher at heart, so for me, Webmin is a great way to configure a system and then show people what was done behind the scenes in those scary configuration files. If Webmin is the gateway drug to Linux system administration, I think I'm okay with that.■

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.

cPacket Networks' cVu

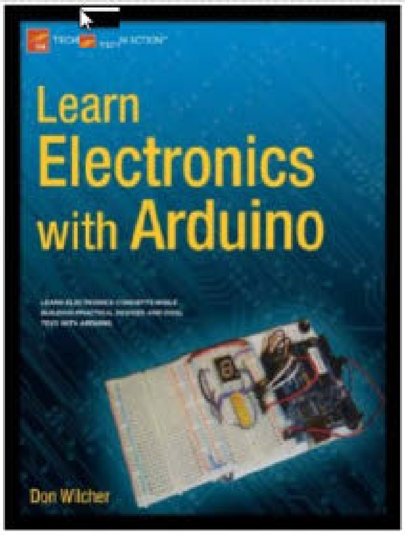
Data centers are getting faster and more complicated. In order to enable the higher levels of network intelligence that is needed to keep up with these trends, without adding undue complexity, cPacket Networks has added a new feature set to its cVu product family. The company says that cVu enables unprecedented intelligence for traffic monitoring and aggregation switches, which significantly improves the efficiency of operations teams running data centers and sophisticated networks. The cVu family offers enhanced pervasive real-time network visibility, which includes granular performance monitoring, microburst auto-detection and filtering of network traffic based on complete packet-and-flow inspection or pattern matching anywhere inside the packet payload. An additional innovation involves utilizing the traffic-monitoring switch as a unified performance monitoring and "tool hub".
http://www.cpacket.com

Opera 12 Browser



Opera recently announced its new Opera 12 browser—code-named Wahoo—with a big "woo-hoo"! The folks at Opera say that the latest entry in the company's long line of desktop Web browsers "is both smarter and faster than its predecessors and introduces new features for both developers and consumers to play with". Key new features include browser themes, a separate process for plugins for added stability, optimized network SSL code for added speed, an API that enables Web applications to use local hardware, paged media support, a new security badge system and language support for Arabic, Farsi, Urdu and Hebrew. Opera says that the paged media project has the potential to change the way browsers handle content, and camera support shows how Web applications can compete with native apps. Opera 12 runs on Linux, Mac OS and Windows.
http://www.opera.com

Don Wilcher's *Learn Electronics with Arduino* (Apress)



If you are a home-brew electronics geek who hasn't tried Arduino yet, what the heck are you waiting for? Get yourself an open-source Arduino microcontroller board and pair it with Don Wilcher's new book *Learn Electronics with Arduino*. Arduino is inarguably changing the way people think about do-it-yourself tech innovation. Wilcher's book uses the discovery method, getting the reader building prototypes right away with solderless breadboards, basic components and scavenged electronic parts. Have some old blinky toys and gadgets lying around? Put them to work! Readers discover that there is no mystery behind how to design and build circuits, practical devices, cool gadgets and electronic toys. On the road to becoming electronics gurus, readers learn to build practical devices like a servo motor controller, a robotic arm, a sound effects generator, a music box and an electronic singing bird.

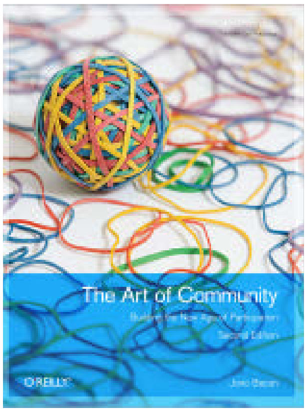
http://www.apress.com

Moxa's ioLogik W5348-HSDPA-C

Industrial automation specialist Moxa recently announced availability of its new product ioLogik W5348-HSDPA-C, a C/C++ programmable 3G remote terminal unit (RTU) controller adapted for data acquisition and condition monitoring that leverages a Linux/GNU platform. This integrated 3G platform, which is designed for remote monitoring applications where wired communication devices are not available, combines cellular modem, I/O controller and data logger into one compact device. Moxa emphasizes the product's open, user-friendly SDKs, which reduce programming overhead in key areas, such as I/O control and condition monitoring, interoperability with SCADA/DB and improving smart communication controls, including cellular connection and SMS. The result, says Moxa, is that engineers can create imaginative, user-defined programs that integrate with localized domains, giving end users considerable additional value.

http://www.moxa.com

Jono Bacon's *The Art of Community*, 2nd ed. (O'Reilly Media)



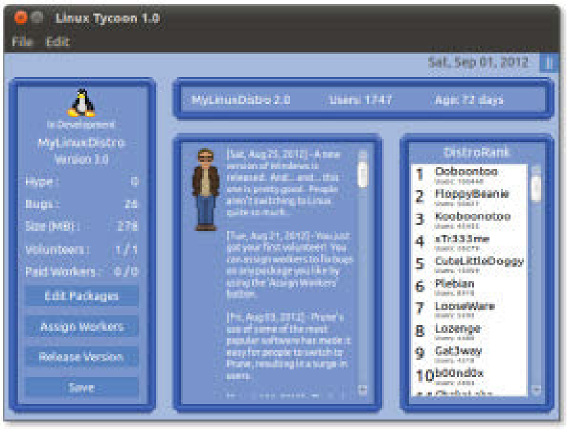
Huge need for your groundbreaking open-source app? Check. Vision for changing the world? Check. Development under way? Check. Participation by a talented group of collaborators? Inconvenient pause. Well don't worry, mate, because Ubuntu community manager, Jono Bacon, is here to help with the updated second edition of his book *The Art of Community: Building the New Age of Participation*. So that you don't have to re-invent the wheel, Bacon distills his own decade-long experience at Ubuntu as well as insights from numerous other successful community management leaders. Bacon explores how to recruit members to your own community, and motivate and manage them to become active participants. Bacon also offers insights on tapping your community as a reliable support network, a valuable source of new ideas and a powerful marketing force. This expanded edition adds content on using social-networking platforms, organizing summits and tracking progress toward goals. A few of the other numerous topics include collaboration techniques, tools and infrastructure, creating buzz, governance issues and managing outsized personalities.

http://www.oreilly.com

BGI's EasyGenomics

Scientific inquiry will continue to advance exponentially as more solutions like BGI's EasyGenomics come on-line. EasyGenomics is a recently updated, cloud-based SaaS application that allows scientists to perform data-heavy "omics"-related research quickly, reliably and intuitively. BGI adds that EasyGenomics integrates various popular next-generation sequencing (NGS) analysis work flows including whole genome resequencing, exome resequencing, RNA-Seq, small RNA and de novo assembly, among others. The back-end technology includes large databases for storing vast datasets and a robust resource management engine that allows precise distribution of computational tasks, real-time task monitoring and prompt response to errors. Thanks to Aspera's integrated fast high-speed file transferring technology and Connect Server Data, transmission rates are 10–100 times faster than common methods, such as FTP. BGI is the world's largest genomics organization.

http://www.genomics.cn/en



[Sat, Aug 25, 2012] - A new version of Windows is released. And... and... this one is pretty good. People aren't switching to Linux quite so much.

[Tue, Aug 21, 2012] - You just got your first volunteer! You can assign workers to fix bugs on any package you like by using the 'Assign Workers' button.

[Fri, Aug 03, 2012] - Prune's use of some of the most popular software has made it easy for people to switch to Prune, resulting in a surge in users.

Bryan Lunduke's *Linux Tycoon*

Bryan Lunduke gave us the official shout that *Linux Tycoon*—"the premier Linux Distro Building Simulator game in the universe"—has arrived at the coveted "One-Point-Oh" status. In this so-called "nerdiest simulation game ever conceived", players simulate building and managing their own Linux distro... without actually building or managing their own Linux distro. Remove the actual work, bug fixing and programming parts, and wham-o!, you've got *Linux Tycoon*. Of course, *Linux Tycoon* runs on Linux, but Mac and Windows users also have the irresistible chance to simulate being a Linux user. Features in progress include Android, iOS and Maemo versions, as well as an on-line, multiplayer game, which is currently in limited beta. *Linux Tycoon* is DRM-free.
http://lunduke.com

Nginx Inc.'s NGINX

NGINX, the second-most-popular Web server for active sites on the Internet, recently released a version 1.2 milestone release with myriad improvements and enhancements. Functionality of the open-source, light-footprint NGINX (pronounced "engine x") includes HTTP server, HTTP and mail reverse proxy, caching, load balancing, compression, request throttling, connection multiplexing and reuse, SSL offload and HTTP media streaming. Version 1.2 is a culmination of NGINX's annual development and extensive quality assurance cycle, led by the core engineering team and user community. Some of the 40 new features include reuse of keepalive connections to upstream servers, consolidation of multiple simultaneous requests to upstream servers, improved load balancing with synchronous health checks, HTTP byte-range limits, extended configuration for connection and request throttling, PCRE JIT optimized regular expressions and reduced memory consumption with long-lived and TLS/SSL connections, among others. Developer Nginx, Inc., says that NGINX now serves more than 25% of the top 1,000 Web sites, more than 10% of all Web sites on the Internet and 70 million Web sites overall.
http://www.nginx.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.

RECONNAISSANCE of a LINUX NETWORK STACK

The Linux kernel is in a military zone with guaranteed punishments for all trespassers. Let's emulate the kernel and study packet flow in the network stack.

RATHEESH KANNOTH

Linux is a free operating system, and that's a boon to all computer-savvy people. People like to know how the kernel works. Many books and tutorials are available, but until you have hands-on experience, you won't gain any solid knowledge. The Linux kernel is a highly secure and powerful operating system kernel. If you try doing anything fishy, the kernel will kill your program. Suppose your program tries to access any memory location of the kernel, the kernel will send a SIGSEGV signal, and your program will core-dump by a segmentation fault. Similarly, you might come across many other examples of the kernel's punishments.

The kernel has defined a set of interfaces, and users can avail the kernel's services only through those interfaces. Those interfaces are called system calls. All system calls have a stub code to verify all the arguments passed. A verification failure will result in the program to core-dump, so it is very difficult to experiment with the kernel.

Kernel modules provide an easy way to execute programs in kernel space, but this is risky, because any faulty kernel module can mess up the operating system, and you will have to hard-reboot the machine.

All these difficulties make the kernel more mysterious. You can't easily peep into the system.

But, UML (User-Mode Linux) comes to the rescue. UML is just a process, an emulation of a Linux kernel, that acts like a Linux machine. Because it is a process, you can manipulate kernel memory and variables' values without any harm to the native Linux machine. You can attach UML to the gdb debugger and do a step-by-step execution of the kernel. If you mess up with UML, and it goes bad, you can kill that process and restart UML at any point of time.

I like to call the UML process a UML machine, because it acts like a different machine altogether. The native Linux machine is nothing but the host Linux machine where you run all these UML processes.

I've been working in the Linux networking domain for the last five years. I found it very difficult to debug kernel modules (in the network stack) because: 1) the kernel is in a highly protected zone, and 2) you need a setup of two or more machines and routers to create a packet flow. Therefore, I created a network of UML machines to overcome this problem, which not only cut down the cost but also saved a lot of time.

This article is not about building UML machines from scratch. Instead, here you will learn how to build a UML network and debug kernel modules effectively without spending resources on additional machines.

The UML source code is available with the Linux kernel. Let's download the 2.6.38 kernel from **http://www.kernel.org** and build a UML kernel. A UML kernel is a process that is in ELF-executable format. Because UML emulates an entire Linux machine, it requires a virtual disk partition to hold small programs, libraries and files, and this virtual disk partition is called the UML filesystem. The UML kernel boots up and mounts this filesystem image as its root partition. You either can create your own or download a UML filesystem from any popular distribution site.

I have done this demo on an Ubuntu 64-bit Lucid operating system (on an Intel

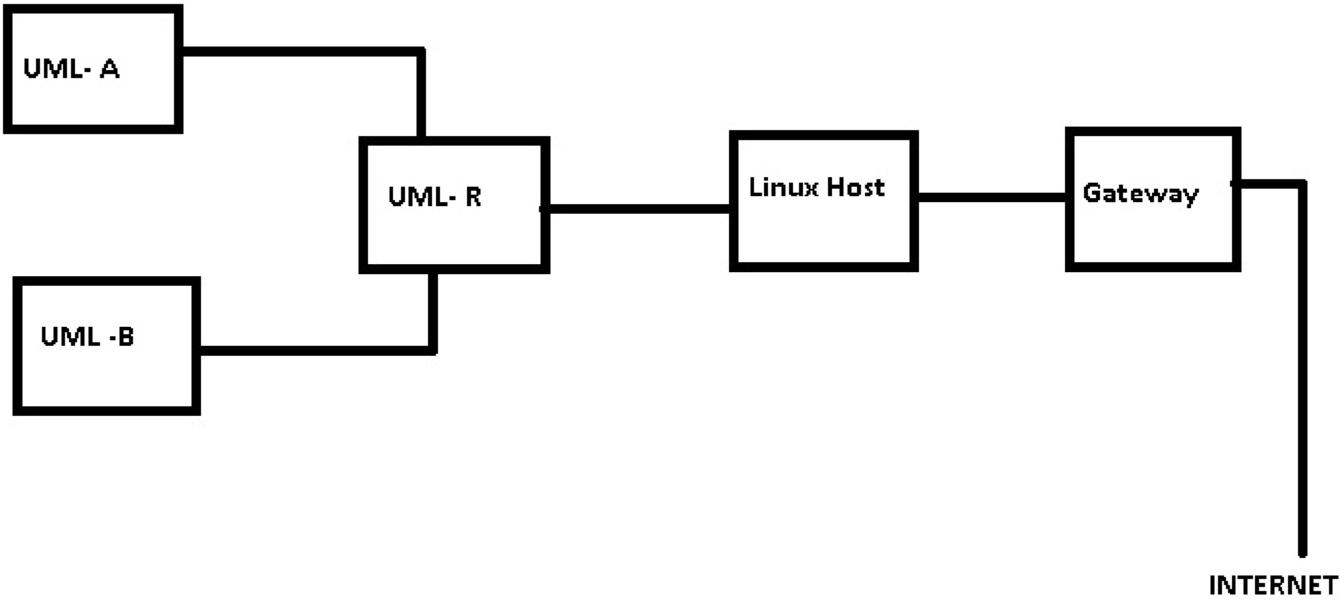


Figure 1. High-Level Block Diagram of the Example UML Setup

Pentium processor). Don't worry if you are using a different Linux distribution or architecture. Just make sure that you download the 2.6.38 kernel and build a UML kernel.

You can configure the kernel using `make menuconfig`. Don't forget to enable CONFIG_DEBUG_INFO and CONFIG_FRAME_POINTER in the config file, as that's necessary for this demo.

I used the following command to build a 32-bit UML kernel:

```
root@ubuntu-lucid:~/$ make ARCH=um SUBARCH=i386
```

Let's build a network of three UML machines, and let's name those machines UML-A, UML-B and UML-R. UML-A and UML-B will behave as normal Linux clients in different IP subnets, but UML-R will be the router machine. UML-R is the default gateway machine for UML-A and UML-B. If you ping the IP address of UML-A from UML-B, the icmp packet should flow through UML-R. Let's make the host Linux machine as the default gateway machine for UML-R. Then, if you ping www.google.com from UML-A, the packet will flow as shown in Figure 1.

Let's make three copies of the UML kernel and the UML filesystem for these three UML machines. It is better to create three directories and keep each copy of the UML kernel and the UML filesystem in each directory:

```
root@ubuntu-lucid:~/root$ mkdir machineA machineB machineR
root@ubuntu-lucid:~/root$ cp uml-filesystem-image
 ➥MachineA/uml-filesystem-image-A
root@ubuntu-lucid:~/root$ cp uml-filesystem-image
```

```
➥MachineB/uml-filesystem-image-B
root@ubuntu-lucid:~/root$ cp uml-filesystem-image
➥MachineR/uml-filesystem-image-R
root@ubuntu-lucid:~/root$ cp linux /test/machineA/
root@ubuntu-lucid:~/root$ cp linux /test/machineB/
root@ubuntu-lucid:~/root$ cp linux /test/machineR/
```

If you boot up all these UML machines, they will look exactly same. So, how do you identify each of the UML machines? To differentiate between them, you can give them different hostnames. The /etc/hostname file contains the machine's hostname, but this file is part of the UML filesystem. You can mount the UML filesystem locally and edit this file to change the hostname:

```
root@ubuntu-lucid:~/root$ mkdir /mnt/mount-R
root@ubuntu-lucid:~/root$ mount -o loop
➥./uml-filesystem-image-R /mnt/mount-R
root@ubuntu-lucid:~/root$ cd /mnt/mount-R
root@ubuntu-lucid:~/root$ echo "MachineR" > etc/hostname
```

Now the UML-R machine's hostname is Machine-R. You can use the same commands and mount uml-filesystem-image-A and uml-filesystem-image-B locally and change the hostnames as "MachineA" and "MachineB", respectively.

Let's boot UML-A and observe:

```
root@ubuntu-lucid:~/root$ ./linux  ubda=./uml-filesystem-image-A
➥mem=256M  umid=myUmlId  eth0=tuntap,,,192.168.50.1
```

UML-A boots up and shows a console prompt. This command configures a tap interface (tap0) on the host Linux machine and an eth0 interface on UML-A. The tap interface is a virtual interface. There is no real hardware attached to it. This is a feature provided by Linux for doing userspace networking. And, this is the right candidate for our network (imagine that the tap0 and eth0 interfaces are like two ends of a water pipe). Refer to the UML Wiki to learn more about the UML kernel command-line options.

The above command assigns the 192.168.50.1 IP address to the tap0 interface on the host Linux machine. You can check this with the ifconfig command on the host Linux machine. The next task is to assign an IP address to the eth0 interface in UML-A. You can assign an IP address to the eth0 interface with ifconfig, but that configuration dies with the UML process. It becomes a repetitive task to assign an IP address every time the UML machine boots up, so you can use an init script to automate that task.

UML-A and UML-B require only one interface because these are just clients, but UML-R needs three interfaces. One interface is to communicate with UML-A, and the second is to communicate with UML-B. The last one is to communicate with the host Linux machine.

Let's bring up the UML machines one

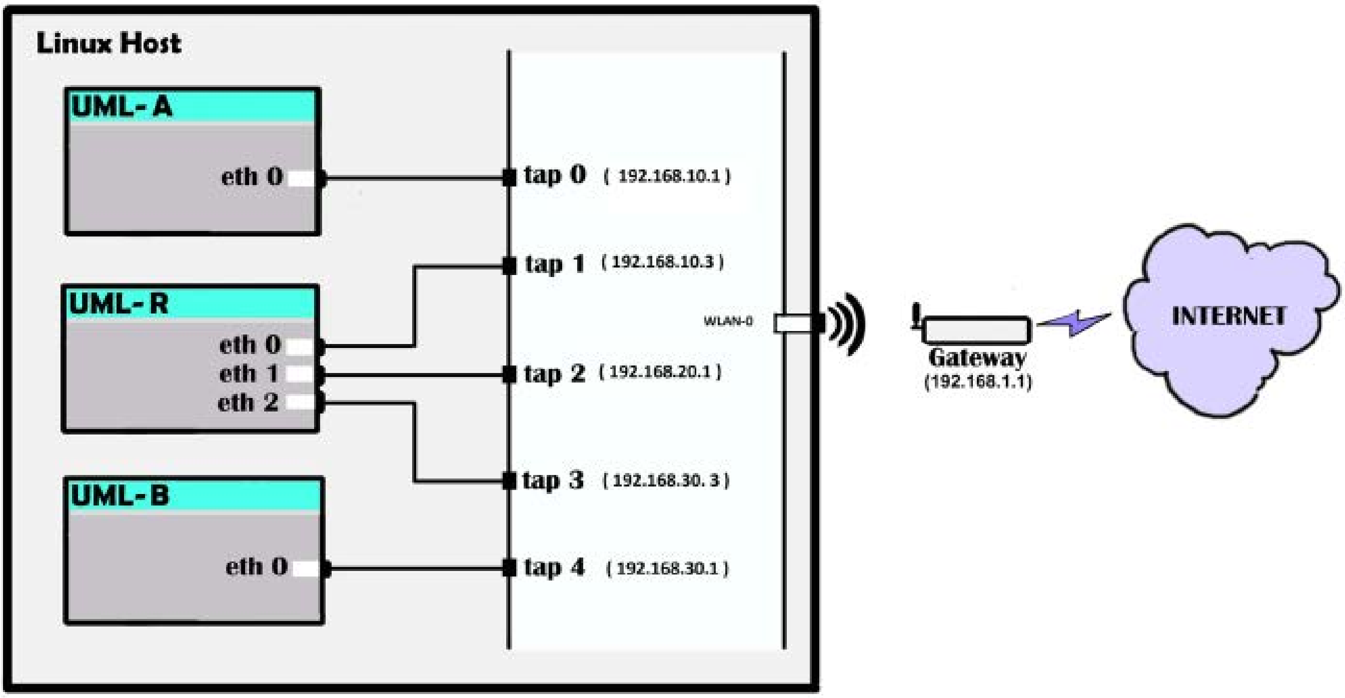


Figure 2. The Three UML Machines Once Booted Up

by one using the commands below (you need to start UML-A, UML-R and then UML-B in that exact order):

```
root@ubuntu-lucid:~/root$ ./linux  ubda=./uml-filesystem-image-A
 ➥mem=256M  umid=client-uml-A eth0=tuntap,,,192.168.10.1
root@ubuntu-lucid:~/root$ ./linux  ubda=./uml-filesystem-image-R
 ➥mem=256M  umid=router-uml-R  eth0=tuntap,,,192.168.10.3
 ➥eth1=tuntap,,,192.168.20.1 eth2=tuntap,,,192.168.30.3
root@ubuntu-lucid:~/root$ ./linux  ubda=./uml-filesystem-image-B
 ➥mem=256M  umid=client-uml-B eth0=tuntap,,,192.168.30.1
```

The IP address of the tap0 interface is 192.168.10.1. Let's assign an IP address from the same subnet to eth0 (in UML-A) and eth0 (in UML-R). Similarly, the IP address of the tap4 interface is 192.168.30.1. Assign the same subnet IP address to eth0 (in UML-B) and eth2 (in UML-R). You can add these commands in an init script to automate these configurations.

Add the commands below to the /etc/rc.local file in uml-filesystem-image-A. These commands will configure the "eth0" interface on UML-A with the IP address 192.168.10.2 and configure the gateway as 192.168.10.50 (the IP address of the eth0 interface in UML-R) on bootup:

```
ifconfig eth0 192.168.10.2  netmask 255.255.255.0 up
route add default gw 192.168.10.50
```

Similarly, add the commands below to

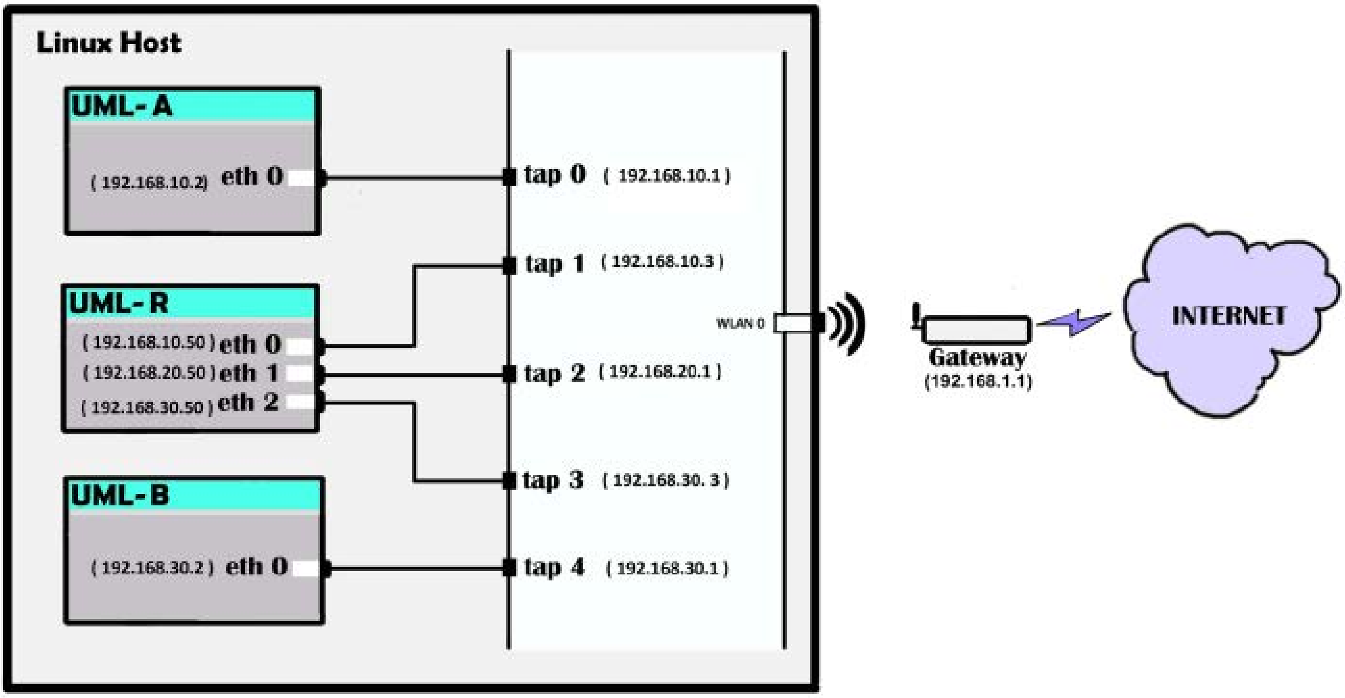


Figure 3. UML Machines, after Interfaces Are Assigned IP Addresses

/etc/rc.local in uml-filesystem-image-B. This command configures the "eth0" interface on UML-B with the 192.168.30.2 IP address and configures the gateway as 192.168.30.50 (the IP address of the eth2 interface in UML-R) on bootup:

```
ifconfig eth0 192.168.30.2  netmask 255.255.255.0 up
route add default gw 192.168.30.50
```

Let's configure one interface on UML-R with the 192.168.10.0/24 subnet IP address and another with the 192.168.30.0/24 subnet IP address. These interfaces are the gateways of UML-A and UML-B. Packets from UML-A and UML-B will route through these interfaces on UML-R. The last interface of UML-R is in the 192.168.20.0/24 subnet. The gateway of UML-R should be an IP address on the host machine, because you ultimately need packets to reach the host machine and route through the host machine's default gateway to the Internet. Because UML-R is the gateway for UML-A and UML-B, you have to turn on ip_forward and add an iptable NAT rule in UML-R. ip_forward tells the kernel stack to allow forwarding of packets. The iptable NAT rule is to masquerade packets.

Add the commands below to /etc/rc.local in uml-filesystem-image-R for this configuration on every UML-R bootup:

```
ifconfig eth0 192.168.10.50 netmask 255.255.255.0  up
ifconfig eth1 192.168.20.50 netmask 255.255.255.0 up
ifconfig eth2 192.168.30.50 netmask 255.255.255.0 up
route add default gw 192.168.20.1

echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

The next task is to bridge the tap0 and tap1 interfaces and the tap3 and tap4 interfaces and assign IP addresses to these bridges. A bridge is a device that links two or more network segments. This is very similar to a network hub device. You can create a software bridge device on Linux using the brctl utility. You can add or delete interfaces to a bridge.

As I mentioned earlier, whatever you send in the eth interface, you can see in its corresponding tap interface. You have three UML machines up and running. Now it's time to configure the host Linux machine to route packets correctly.

1. Create a bridge (br0), add the tap interface of UML-A and one tap interface of UML-R to br0.

2. Create a bridge (br1), add the tap interface of UML-B and one tap interface of UML-R to br1.

3. Assign an IP address to br0 from the same subnet of UML-A's eth0 interface IP address.

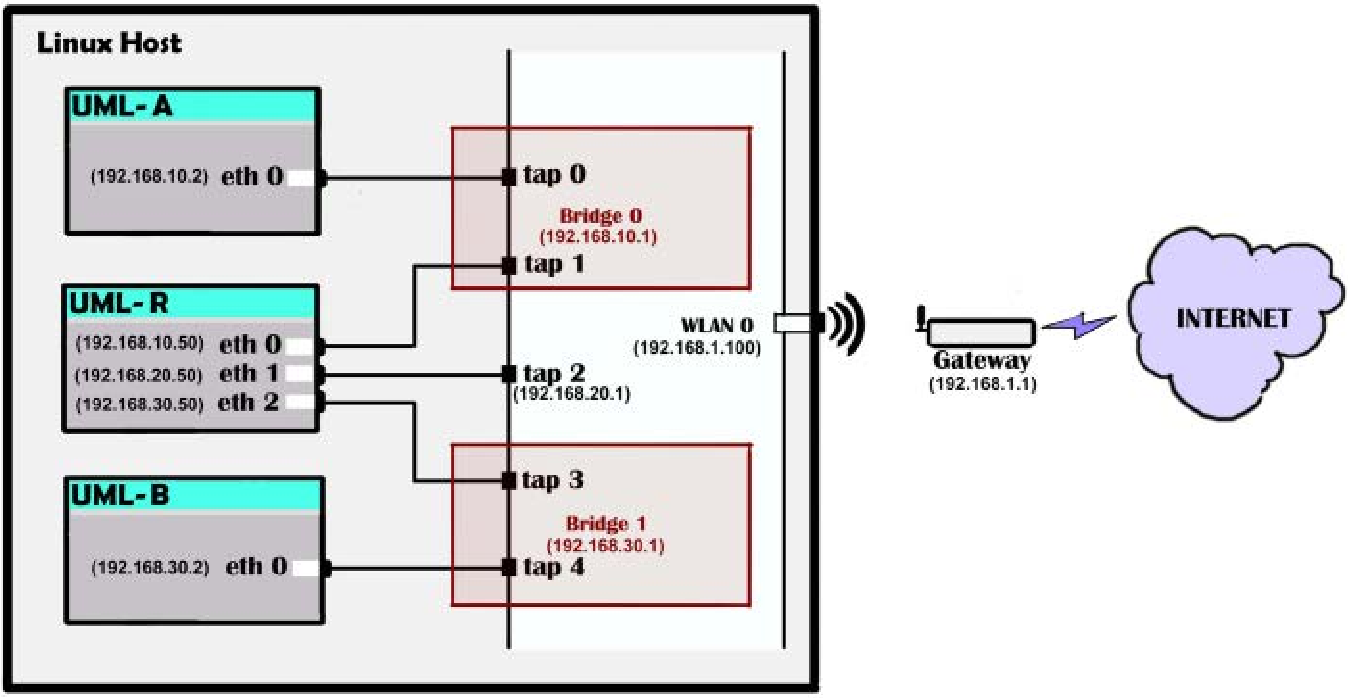


Figure 4. UML Machines, after Executing the setup_network_connections.sh Script

4. Assign an IP address to br1 from the same subnet of UML-B's eth0 interface IP address.

5. Assign an IP address to the third interface of UML-R and its tap interface from the same subnet.

6. Flush the iptables filter rule on the host Linux machine so that the firewall won't drop any packets.

7. Add the Masquerade NAT rule on the host Linux machine.

8. Enable ip_forward on the host Linux machine.

Executing steps 1 through 5—bridge tap0, tap1 to br0 and assign the 192.168.10.1 IP address (the gateway IP address of UML-R):

```
root@ubuntu-lucid:~/root$ brctl addbr br0
root@ubuntu-lucid:~/root$ brctl addif br0 tap0
root@ubuntu-lucid:~/root$ brctl addif br0 tap1
root@ubuntu-lucid:~/root$ ifconfig br0 192.168.10.1
 ➥netmask 255.255.255.0    up
```

Bridge tap3, tap4 to br1 and assign an 192.168.30.1 IP address:

```
root@ubuntu-lucid:~/root$ brctl  addbr br1
root@ubuntu-lucid:~/root$ brctl addif br1 tap3
root@ubuntu-lucid:~/root$ brctl addif br1 tap4
root@ubuntu-lucid:~/root$ ifconfig br1 192.168.30.1
 ➥netmask 255.255.255.0    up
```

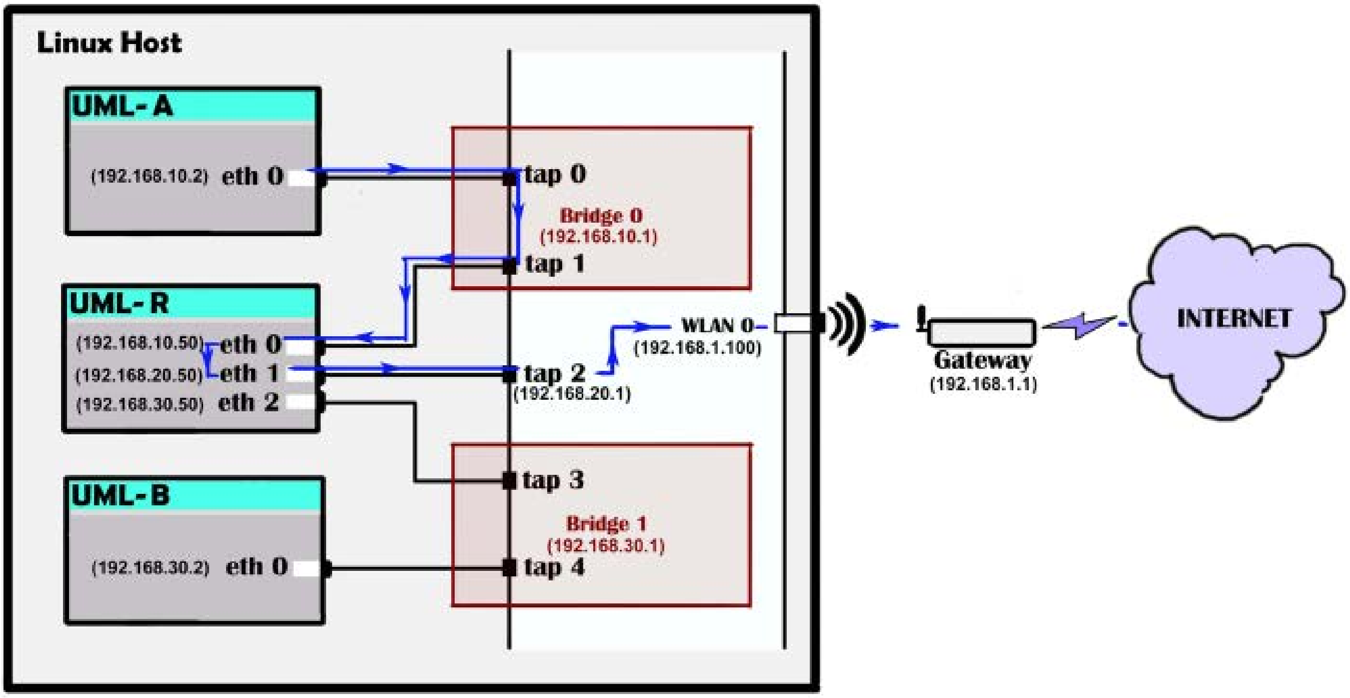


Figure 5. Packet Flow in the UML Network

Listing 1. setup_network_connections.sh

```
###### create the br0 and br1 bridge with the brctl utility
brctl  addbr br0
brctl  addbr br1

##### delete all old configurations if they exist
ifconfig br0 0.0.0.0 down
brctl  delif br0 tap0
brctl  delif br0 tap1
ifconfig br1 0.0.0.0 down
brctl  delif br1 tap3
brctl  delif br1 tap4

##### flush all filter and nat rules
iptables -t nat -F
iptables  -F


##### turn on debug prints
set -x

####  make all tap interfaces up.
ifconfig tap0 0.0.0.0 up
ifconfig tap1 0.0.0.0 up
ifconfig tap3 0.0.0.0 up
ifconfig tap4 0.0.0.0 up

#### add tap0 and tap1 to br0 bridge
brctl addif br0 tap0
brctl addif br0 tap1

#### add tap3 and tap4 to br1 bridge
brctl addif br1 tap3
brctl addif br1 tap4

##### assign br0 with 192.168.10.1 ip and make it up
ifconfig br0 192.168.10.1 netmask 255.255.255.0    up

##### assign br1 with 192.168.30.1 ip and make it up
ifconfig br1 192.168.30.1 netmask 255.255.255.0    up

##### assign tap2 interface with 192.168.20.1 ip and make it up
ifconfig tap2 192.168.20.1  netmask 255.255.255.0  up

##### enable ip forward
echo 1 > /proc/sys/net/ipv4/ip_forward

##### make the default policy of the forward chain as accept
##### to avoid any possibility of dropping packets in filter chain
iptables -P FORWARD ACCEPT

##### add a NAT rule to Masquerade packets from uml-R to the host machine.
iptables -t nat -A POSTROUTING -o wlan0 -j MASQUERADE
```

Assign the tap2 IP address with 192.168.20.1:

```
root@ubuntu-lucid:~/root$ ifconfig tap2 192.168.20.1
 ➥netmask 255.255.255.0  up
```

Flush out the firewall rules in the host machine:

```
root@ubuntu-lucid:~/root$iptables -t nat -F
root@ubuntu-lucid:~/root$ipables  -F
```

At the end of step 5, you will get a setup like the one shown in Figure 4.

I have written a script (Listing 1) to automate all these tasks with comments added for easy readability. All you need to do is start UML-A, UML-R and UML-B in the same order and run the script on the host Linux machine. Note that "wlan0" is my host machine's default gateway interface; you will need to modify that with the correct interface name before executing this script.

Now the setup is ready, so if you ping www.google.com from UML-A, the icmp packet follows a path as shown in Figure 5.

How do you verify that packets are getting routed through UML-R? A utility called traceroute. The `traceroute` command will show all the hops in its path until the destination. Let's `traceroute` www.google.com from UML-A. Because www.google.com is a domain name, you have to resolve the domain name into a valid IP address. Add some valid DNS server names to the /etc/resolv.conf file in UML-A and UML-B.

I executed `traceroute` to 192.168.0.1 (my host machine's default gateway IP address) from UML-A. You can see from the output snapshot below that packets are routed through UML-R (192.168.10.50 is an IP address in the UML-R machine) then to the host machine (192.168.20.1 is an IP address in the host machine):

```
MachineA@/root# traceroute 192.168.0.1
traceroute to 192.168.0.1 (192.168.0.1), 30 hops max, 40 byte packets
 1  192.168.10.50 (192.168.10.50)  0.364 ms  0.232 ms  0.242 ms
 2  192.168.20.1 (192.168.20.1)  0.326 ms  0.293 ms  0.291 ms
 3  192.168.0.1 (192.168.0.1)  1.364 ms  1.375 ms  1.466 ms
```

Building Modules

It is not easy to develop or enhance a kernel module, because it is in kernel space (as I mentioned previously). UML helps here also. You can attach GDB to UML and do a step-by-step execution. Let's debug the ipt_REJECT.ko module in machine-R. ipt_REJECT.ko is a target module for iptable rules. Let's add filter rules on the UML-R machine. Filter rules are firewall rules by which you can selectively REJECT packets.

First, you need to make sure that ipt_REJECT is not built as part of the UML-R kernel. If it is part of the UML-R kernel, you need to run `make menuconfig` and unselect this module, and then rebuild the UML-R kernel again.

It is very easy to build a kernel module. You need three entities for a kernel module build:

1. Source code of the module.

2. Makefile.

3. Linux kernel source code.

ipt_REJECT.c is the source code of the ipt_REJECT.ko module. This file is part of the Linux kernel source code. Let's copy this file to a directory. You need to create a Makefile in the same directory. You can build this module and `scp` the module to the UML-R machine. There are two ways to copy files between UML and the host machine. One is with `scp` and the other is by mounting the UML filesystem locally and copying files to this mounted directory. The good part is that you can mount the UML filesystem even though the UML machine is running.

Here are the commands to build the ipt_REJECT.ko module:

```
root@ubuntu-lucid:~/root$ mkdir  /workout/
root@ubuntu-lucid:~/root$ cd /workout/
root@ubuntu-lucid:~/workout$ cp /workspace/linux-2.6.38/
➥net/ipv4/netfilter/ipt_REJECT.c  ./ipt_REJECT.c
root@ubuntu-lucid:~/workout$ echo "obj-m := ipt_REJECT.o"
 ➥> ./Makefile
root@ubuntu-lucid:~/workout$ make -C /workspace/linux-2.6.38/
 ➥M=`pwd` modules ARCH=um SUBARCH=i386
root@ubuntu-lucid:~/workout$ scp ipt_REJECT.ko
 ➥root@192.168.10.50:/tmp/
```

Let's see the capability of the REJECT target module. Remove all the filter rules in UML-R:

```
MachineR@/root# iptables -F
```

Ping www.google.com from MachineA:

```
MachineA@/root$ ping www.google.com
```

You can ping www.google.com because there are no filter rules loaded in the UML-R machine. UML-R is the default gateway machine for UML-A.

Now, `insmod` the REJECT module, and add a rule in the filter table to block all icmp packets in the UML-R machine:

```
MachineR@/root# insmod /tmp/ipt_REJECT.ko
MachineR@/root# iptables -A FORWARD -p icmp -j REJECT
```

Try to ping www.google.com from UML-A again:

```
MachineA@/root#  ping www.google.com
```

`ping` would fail as the REJECT rule

You can attach GDB to UML because UML is just a user-mode process.

blocks ping packets (icmp packets). If you flush out the rules in UML-R (using `iptables -F`), icmp packets will start flowing again.

Running GDB on the Kernel

You can attach GDB to UML because UML is just a user-mode process. You need to know the UML's pid to attach to GDB. You can find the pid easily from umid (umid is nothing but an argument passed to the UML kernel):

```
root@ubuntu-lucid:/$ ./linux  ubda=uml-machine-R,./
➥uml-filesystem-image-R mem=256M  umid=router-uml-R
 ➥eth2=tuntap,,,192.168.10.3 eth3=tuntap,,,192.168.20.1
 ➥eth4=tuntap,,,192.168.30.3
```

Here, the umid is client-uml-R. The ~/.uml/router-uml-R/pid file contains the pid of the UML-R process.

Let's attach GDB to UML-R:

```
root@ubuntu-lucid:/$ pid=$(cat ~/.uml/router-uml-R/pid)
root@ubuntu-lucid:/$ gdb ./linux  $pid
```

The moment you attach GDB to UML-R, the Uml-R console stops execution. You can't type anything in UML-R. You can type `c` ("continue") on the GDB prompt to make the UML-R prompt active:

```
(gdb) c
```

Detach GDB with the command `q` ("quit") at the GDB prompt:

```
(gdb) q
```

Step-by-Step Execution of a Module

You already have seen that the control reaches ipt_REJECT.ko when you pinged www.google.com from UML-A after loading an iptable REJECT rule in UML-R. You can attach GDB to UML-R and set a breakpoint in the ipt_REJECT.ko module code. ipt_REJECT.ko is an ELF file. ELF is an executable file format in the Linux OS. An ELF binary has many sections, and you can display those sections using the `readelf` command. In order to set a breakpoint, you need to load debug symbols to GDB and inform GDB about the ".text" section address of the module. ".text" is a code segment of the ELF binary.

You can find the code segment address from either the proc or sysfs file entry:

1. The proc entry: in the file /proc/modules.

2. The sysfs entry: in the file /sys/module/<module-name>/sections/.text.

Let's load the debug symbols and address of .text to GDB:

```
(gdb) add-symbol-file /workout/ipt_REJECT.ko  <address_of_.text>
```

Now you can set the breakpoint in the ipt_REJECT.ko module. Open the ipt_REJECT.c file and check the functions available. Whenever an icmp packet flows through UML-R, the reject_tg() function gets called. Let's put a breakpoint in this function and try pinging from UML-A:

```
(gdb) b reject_tg
(gdb) c
```

```
MachineA@/root# ping www.google.com
```

Now control will hit the breakpoint, and it's time to print some variable in the module.

List the source code of the module:

```
(gdb) list
```

Print the sk_buff structure. sk_buff is the structure that holds a network packet. Each packet has an sk_buff structure (**http://lxr.linux.no/#linux+v2.6.38/include/linux/skbuff.h#L319**). Let's print all the fields in this structure:

```
(gdb) p *(struct sk_buff *)skb
```

You can use GDB's s command to do step execution. Press c or q to continue execution or to detach GDB from UML.

Conclusion

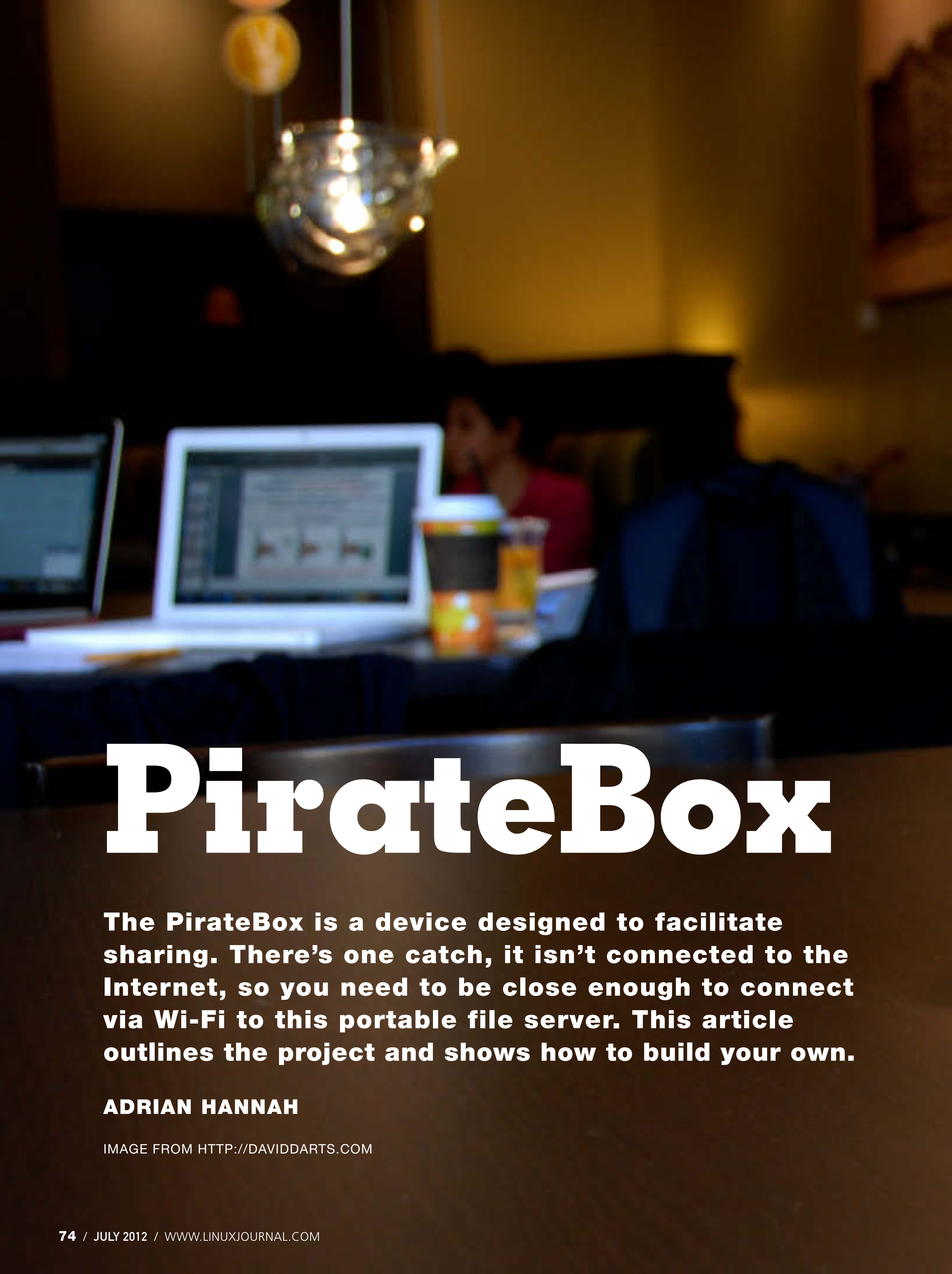
UML is a very versatile tool. You can create different kinds of network nodes using UML. You can debug most parts of the Linux kernel using UML. I don't consider UML to be a good tool for debugging device drivers, which has a direct dependency on a particular hardware. But certainly, it is an intelligent tool for understanding the TCP/IP stack, debugging kernel modules and so on. You can play with UML and learn a lot without doing any harm to your Linux machine. I bet you can become a Linux network expert in the near future.■

Ratheesh Kannoth is a senior software engineer with Cisco Systems. You can reach him at ratheesh.ksz@gmail.com.

Resources

The User-Mode Linux Kernel Home Page: **http://user-mode-linux.sourceforge.net**

User-Mode Linux—Ubuntu Documentation: **https:/help.ubuntu.com/community/UserModeLinux**



PirateBox

The PirateBox is a device designed to facilitate sharing. There's one catch, it isn't connected to the Internet, so you need to be close enough to connect via Wi-Fi to this portable file server. This article outlines the project and shows how to build your own.

ADRIAN HANNAH

IMAGE FROM HTTP://DAVIDDARTS.COM

In days of yore (the early- to mid-1990s) those of us using the "Internet", as it was, delighted in our ability to communicate with others and share things: images, MIDI files, games and so on. These days, although file sharing still exists, that feeling of community has been leeched away from the same activities, and people are somewhat skeptical of sharing files on-line anymore for fear of a lawsuit or who's watching.

Enter David Darts, the Chair of the Art Department at NYU. Darts, aware of the Dead Drops (**http://deaddrops.com**) movement, was looking for a way for his students to be able to share files easily in the classroom. Finding nothing on the market, he designed the first iteration of the PirateBox.

> "Protecting our privacy and our anonymity is closely related to the preservation of our freedoms."—David Darts

The PirateBox is a self-contained file-sharing device that is designed to be simple to build and use. At the same time, Darts wanted something that would be private and anonymous.

The PirateBox doesn't connect to the Internet for this reason. It is simply a local file-sharing device, so the only thing you can do when connected to it is chat with other people connected to the box or share files. This creates an interesting social dynamic, because you are forced to interact (directly or indirectly) with the people connected to the PirateBox.

The PirateBox doesn't log any information. "The PirateBox has no tool to track or identify users. If ill-intentioned people—or the police—came here and seized my box, they will never know who used it", explains Darts. This means the only information stored about any users by the PirateBox is any actual files uploaded by them.

The prototype of the PirateBox was a plug computer, a wireless router and a battery fit snugly into a metal lunchbox. After releasing the design on the Internet, the current iteration of the PirateBox (and the one used by Darts himself) is built onto a Buffalo AirStation wireless router (although it's possible to install it on anything running OpenWRT), bringing the components down to only the router and a battery. One branch of the project is working on porting it to the Android OS, and another is working on building a PirateBox using only open-source components.

How to Build a PirateBox

There are several tutorials on the PirateBox Web site (**http://wiki.daviddarts.com/PirateBox_DIY**) on how to set up a PirateBox based on what platform you are planning on using. The simplest (and recommended) way of setting it up is on an OpenWRT router. For the purpose of this article, I assume this is the route you are taking. The site suggests using a TP-Link MR3020 or a TP-Link TL-WR703N, but it should work on any router with OpenWRT installed that also has a USB port. You also need a USB Flash drive and a USB battery (should you want to be fully mobile).

Assuming you have gone through the initial OpenWRT installation (I don't go into this process in this article), you need to make some configuration changes to allow your router Internet access initially (the PirateBox software will ensure that this is locked down later).

First, you should set a password for the root account (which also will enable SSH). Telnet into the router, and run `passwd`.

The next thing you need to do is set up your network interfaces. Modify /etc/config/network to look similar to this:

```
config interface 'loopback'
        option ifname 'lo'
        option proto 'static'
        option ipaddr '127.0.0.1'
        option netmask '255.0.0.0'

config interface 'lan'
        option ifname 'eth0'
        option type 'bridge'
        option proto 'static'
        option ipaddr '192.168.2.111'
        option netmask '255.255.255.0'
        option gateway '192.168.2.1'
```

Dead Drops

Dead Drops is an off-line peer-to-peer file-sharing network in public. In other words, it is a system of USB Flash drives embedded in walls, curbs and buildings. Observant passersby will notice the drop and, hopefully, connect a device to it. They then are encouraged to drop or collect any files they want on this drive. For more information, comments and a map of all Dead Drops worldwide, go to **http://deaddrops.com**.

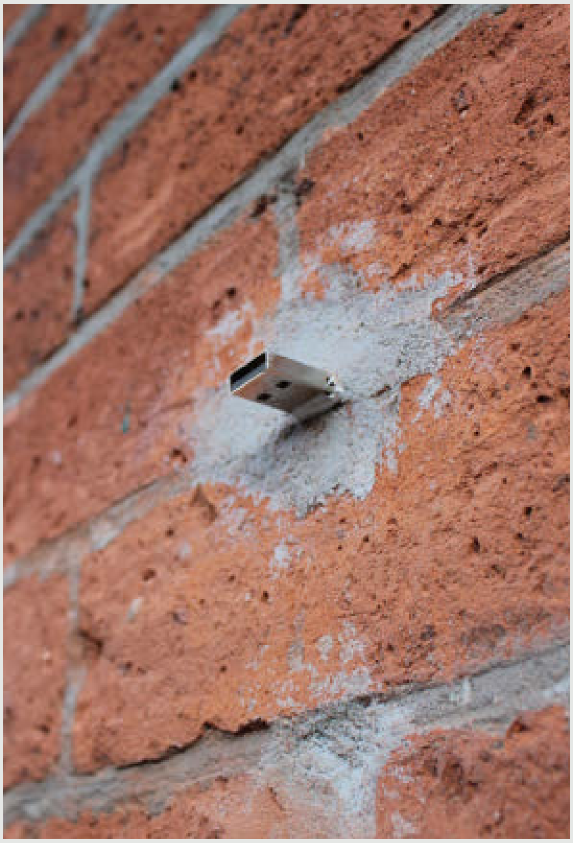


IMAGE FROM HTTP://DEADDROPS.COM

```
        list dns '192.168.2.1'
        list dns '8.8.8.8'
```

assuming that the router's IP address will be 192.168.2.111 and your gateway is at 192.168.2.1.

Next, modify the beginning of the firewall config file (/etc/config/firewall) to look like this:

```
config defaults
        option syn_flood     '1'
        option input         'ACCEPT'
        option output        'ACCEPT'
        option forward       'ACCEPT'
#Uncomment this line to disable ipv6 rules
#       option disable_ipv6  1

config zone
        option name          'lan'
        option network       'lan'
        option input         'ACCEPT'
        option output        'ACCEPT'
        option forward       'ACCEPT'

config zone
        option name          'wan'
        option network       'wan'
        option input         'ACCEPT'
        option output        'ACCEPT'
        option forward       'ACCEPT'
        option masq          '1'
        option mtu_fix       '1'
```

Leave the rest of the file untouched.

What Does David Darts Keep on His PirateBox?

- **A collection of stories by Cory Doctorow.**
- **Abbie Hoffman's *Steal This Book*.**
- **DJ Danger Mouse's *The Grey Album*.**
- **Girl Talk's *Feed the Animals*.**
- **A collection of songs by Jonathan Coulton.**
- **Some animations by Nina Paley.**

(All freely available and released under some sort of copyleft protection.)

The point of the PirateBox is to be integrated easily into a public space with zero effort on the part of the end user; otherwise, no one ever would use it!

In /etc/config/wireless, find the line that reads "option disabled" and change it to "option disabled 0" to enable wireless. At this point, you need to reboot the router.

Now, connect a FAT32-partitioned USB Flash drive to the router, and run the following commands on the router:

```
cd /tmp
wget http://piratebox.aod-rpg.de/piratebox_0.3-2_all.ipk
opkg update && opkg install piratebox*
```

When you restart the device, you should see a new wireless network called "PirateBox - Share Freely". Plug your router in to a USB battery, and place everything into an enclosure of some kind (preferably something black with the Jolly Roger emblazoned on the side). Congratulations! With little to no hassle, you've created a mobile, anonymous sharing device!

Using the PirateBox

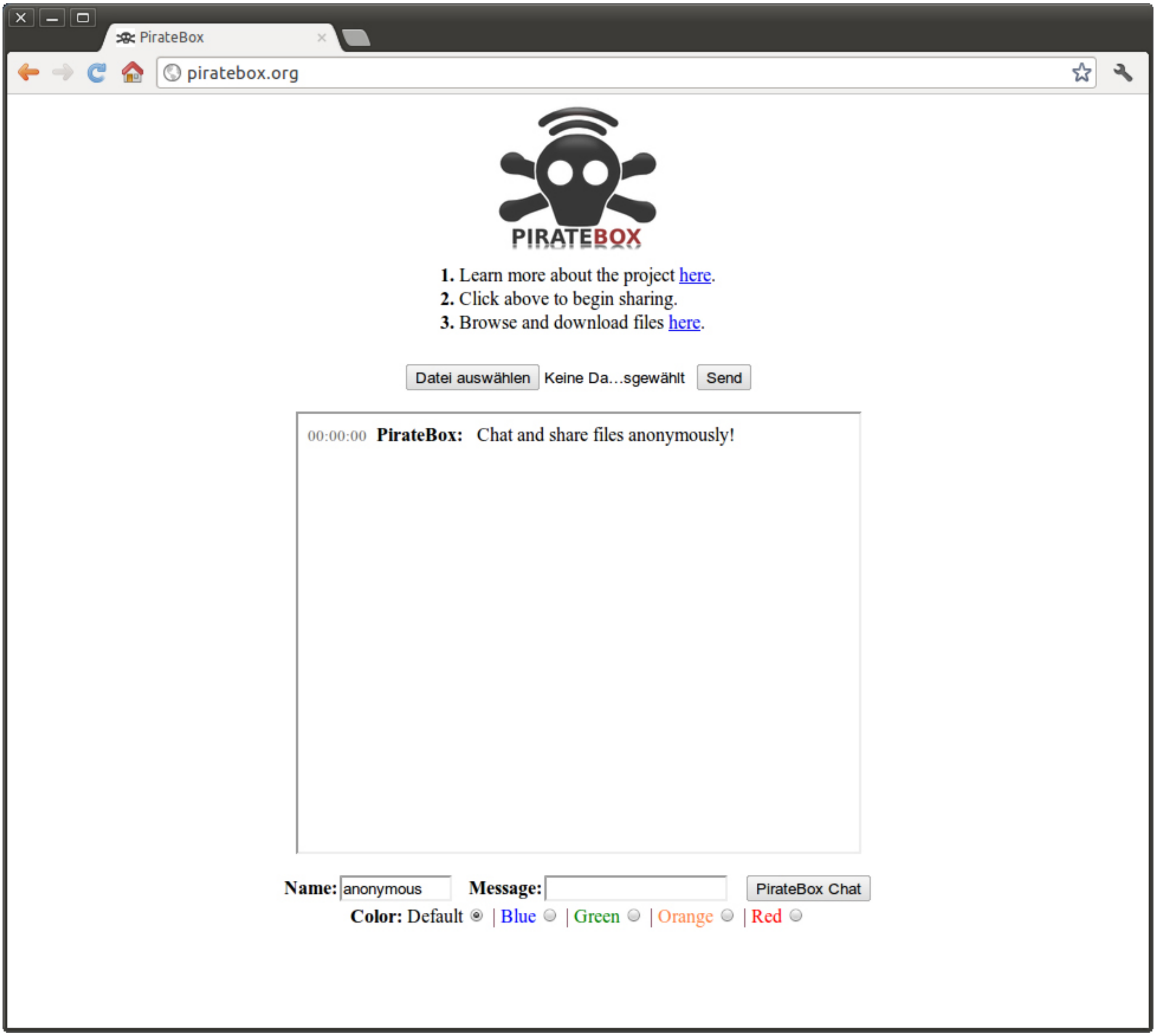
The point of the PirateBox is to be integrated easily into a public space with zero effort on the part of the end user; otherwise, no one ever would use it! This means using it has to be incredibly simple, and it is. If you are connected to the "PirateBox - Share Freely" network and you try to open a Web page, you automatically will be redirected to this page (Figure 1).

As you can see, you are given choices

Adding USB Support to OpenWRT

USB support can be added by running the following commands:

```
opkg update
opkg install kmod-usb-uhci
insmod usbcore
insmod uhci
opkg install kmod-usb-ohci
insmod usb-ohci
```



1. Learn more about the project here.
2. Click above to begin sharing.
3. Browse and download files here.

00:00:00 **PirateBox:** Chat and share files anonymously!

Name: anonymous **Message:**

Color: Default ● | Blue ○ | Green ○ | Orange ○ | Red ○

Figure 1. PirateBox Home Screen

as to what you wish to do: browse and download files, upload files or chat with other users—all of which is exceedingly easy to do. Go build your own PirateBox and get sharing! ■

Adrian Hannah has spent the last 15 years bashing keyboards to make computers do what he tells them. He currently is working as a system administrator for the federal government. He is a jack of all trades and a master of none. Find out more at http://about.me/adrianhannah.

TCP Thin-Stream Modifications: Reduced Latency for Interactive Applications

Sometimes your interactive TCP-based applications lag. This article shows you how to reduce the worst latency.

ANDREAS PETLUND

Are you tired of having to wait for seconds for your networked real-time application to respond? Did you know that Linux has recently added mechanisms that will help reduce the latency? If you use Linux for VNC, SSH, VoIP or on-line games, you should read this article. Two little-known TCP modifications can reduce latency by several seconds in cases where retransmissions are needed to recover lost data. In this article, I introduce these new techniques that can be enabled per stream or machine-wide without any modifications to the application. I show how these modifications have improved maximum latencies by several seconds in *Age of Conan*, an MMORPG game by Funcom.

Background

The communication system in Linux provides heaps of configuration options. Still, many users keep them at the default settings, which serves most causes nicely. In some cases, however, the performance experienced by the application can be improved significantly by turning a few knobs.

Most services today use a variant of TCP. In the course of many years, TCP has been optimized for bulk download, such as file transfers and Web browsing. These days, we use more and more interactive applications over the Internet, and many of those rely on TCP, although most traditional TCP implementations handle them badly. For several reasons, they recover lost packets for these applications much more slowly than for download traffic, often longer than is acceptable. The Linux kernel has recently included enhanced system support for interactive services by modifying TCP's packet loss recovery schemes for thin-stream traffic. But, it is up to the developers and administrators to use it.

Thin-Stream Applications

A large selection of networked interactive applications are characterized by a low packet rate combined with small packet payloads. These are called thin streams. Multiplayer on-line games, IP telephony/audio conferences, sensor networks, remote terminals, control systems, virtual reality systems, augmented reality systems and stock exchange systems are all common examples of such applications, and all have millions of users every day.

Compared to bulk data transfers like HTTP or FTP, thin-stream applications send very few packets, with small

Table 1. Examples of thin- (and bulk-) stream packet statistics based on analysis of real-world packet traces. All traces are one-way (no ACKs are recorded) packet traffic.

Application	Payload Size (bytes):			Packet Interarrival Time (ms):						Avg Bandwidth Used	
	avg	min	max	avg	med	min	max	1%	99%	(pps)	(bps)
VNC (from client)	8	1	106	34	8	<1	5451	<1	517	29.412	17K
Skype (2 users)	236	14	1267	34	40	<1	1671	4	80	29.412	69K
SSH text session	48	16	752	323	159	<1	76610	32	3616	3.096	2825
Anarchy Online	98	8	1333	632	449	7	17032	83	4195	1.582	2168
World of Warcraft	26	6	1228	314	133	<1	14855	<1	3785	3.185	2046
Age of Conan	80	5	1460	86	57	<1	1375	24	386	11.628	12K
HTTP download	1447	64	1448	<1	<1	<1	186	<1	8	>1000	14M
FTP download	1447	40	1448	<1	<1	<1	339	<1	<1	>1000	82M

payloads, but many of them are interactive and users become annoyed quickly when they experience large latencies. Just how much latency users can accept has been investigated for few applications. ITU-T (International Telecommunication Union's Telecomunication Standarization Sector—a standardization organization) has done it for telephony and audio conferencing and defined guidelines for the satisfactory one-way transmission delay: quality is bad when the delay exceeds 150–200ms, and the maximum delay should not exceed 400ms.

Similarly, experiments show that for on-line games, some latency is tolerable, as long as it does not exceed the threshold for playability. Latency limits for on-line games depend on the game type and ranges from 100ms to 1,000ms. For other kinds of interactive applications, such as SSH shells and VNC remote control, we all know how a lag can be a real pain. It also has been shown that pro-gamers can adapt to larger lag than newbies, but that they are much more annoyed by it.

A Representative Example: *Anarchy Online*

We had been wondering for a long time how game traffic looked when one saw a lot of streams at once. Could one reduce lag by shaping game traffic into constant-sized TCP streams? Would it be possible to see when avatars interacted?

To learn more about this, we monitored the game traffic from Funcom's *Anarchy Online*. We captured all traffic from one of the game servers using tcpdump. We soon found that we were asking the wrong questions and analyzed the latencies that players experienced. Figure 1 shows statistics for delay and loss.

In Figure 1a, I have drawn a line at 500ms. It is an estimate of the delay that the majority of players finds just acceptable in a role-playing game like *Anarchy*. Everybody whose value is above that line probably has experienced annoying lag. The graph shows that nearly half the measured streams during this hour of game play had high-latency events, and that these are closely related to packet losses (Figure 1b). The worst case in this one-hour, one-region measurement is the connection where the user experienced six consecutive retransmissions resulting in a delay of 67 (!) seconds.

New TCP Mechanisms

The high delays you can see in the previous section stem from the default TCP loss recovery mechanisms. We have experimented with all the available TCP variants in Linux to find the TCP flavor that is best suited for low-latency,

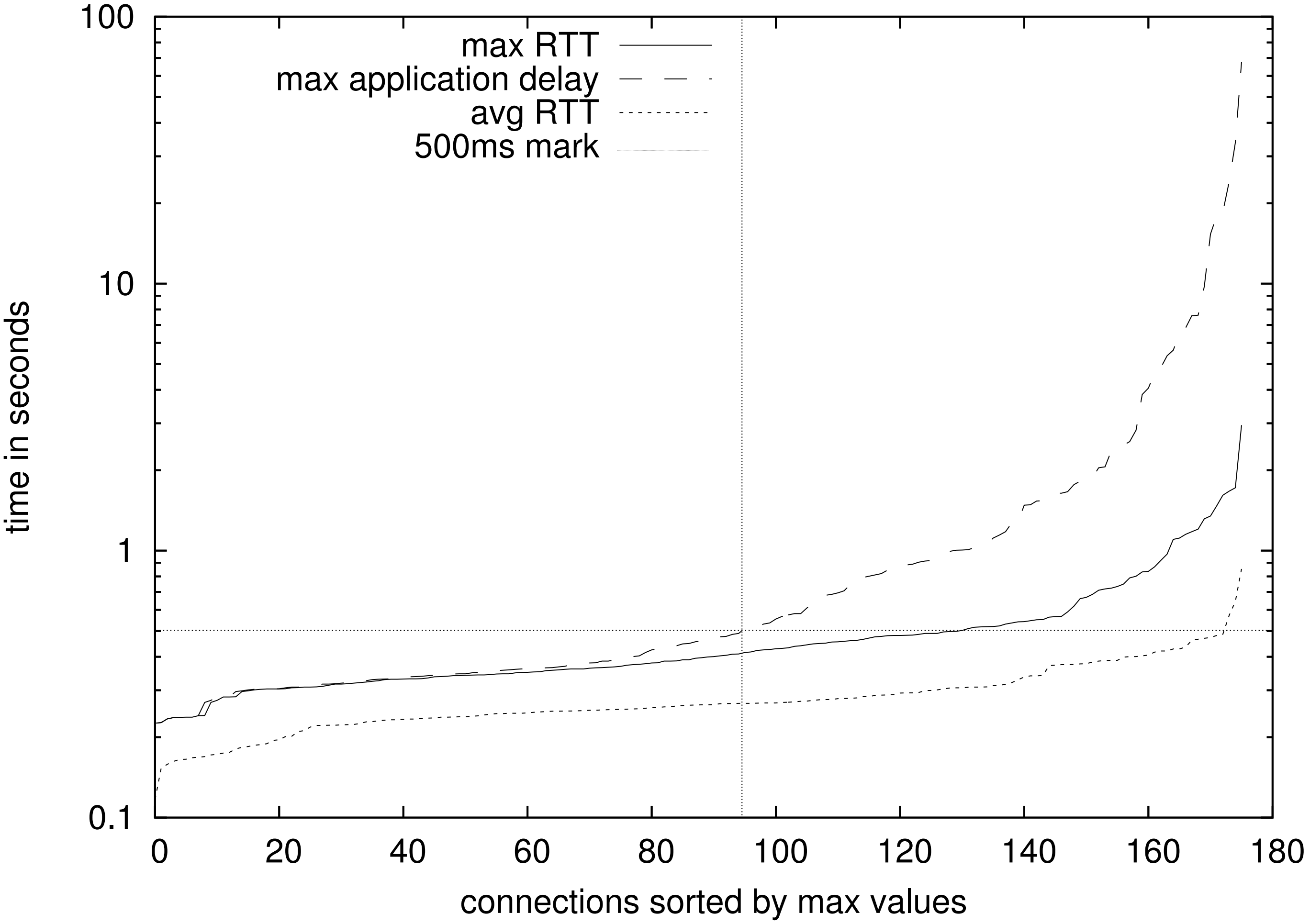


Figure 1a. Round-Trip Time vs. Maximum Application Delay (Analysis of Trace from *Anarchy Online*)

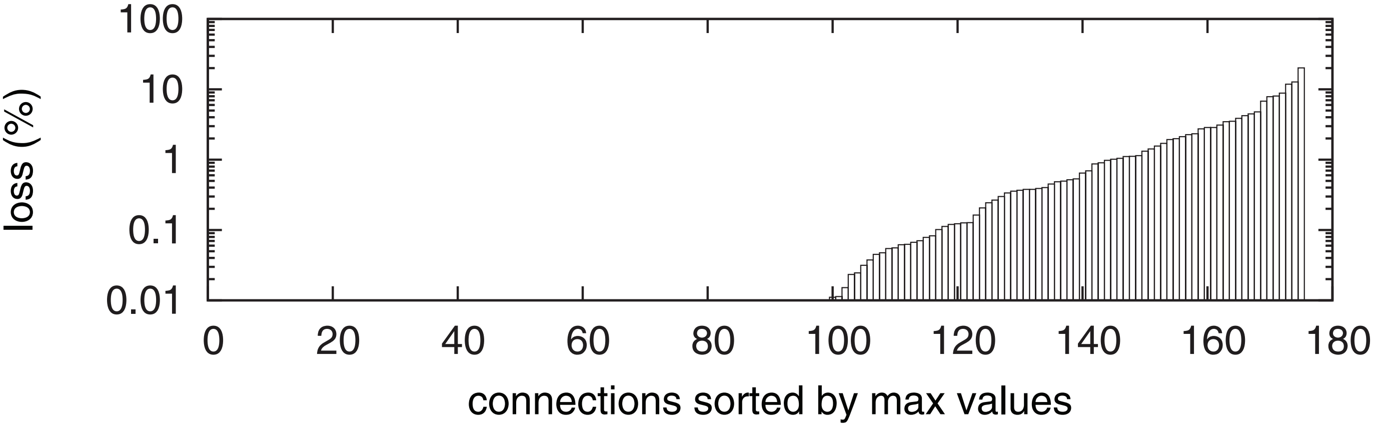


Figure 1b. Per-Stream Loss Rate (Analysis of Trace from *Anarchy Online*)

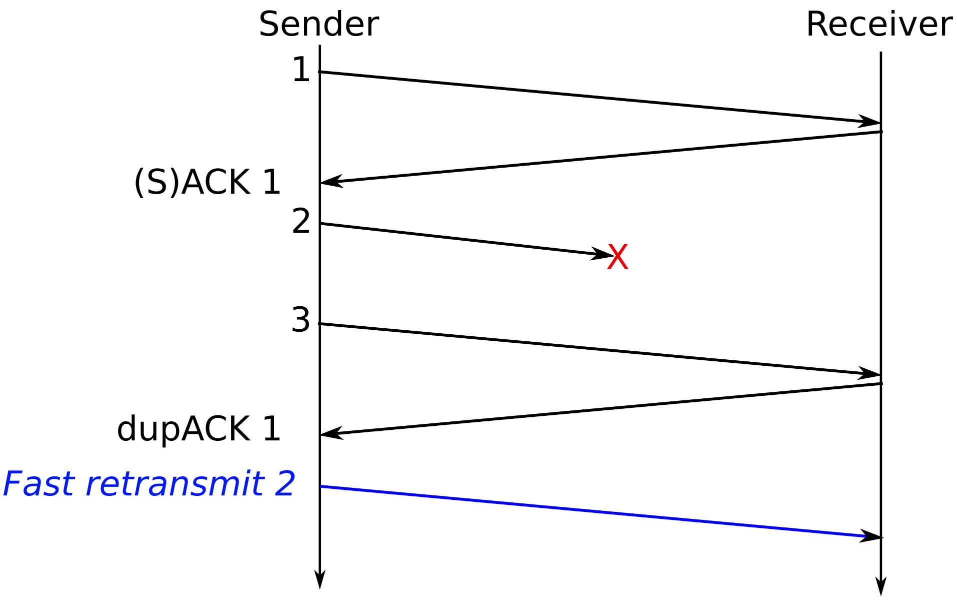


Figure 2. Thin Fast Retransmit

thin-stream applications. The result was disheartening: all TCP variants suffer from long retransmission delays for thin-stream traffic.

We wanted to do something about this and implemented several modifications to Linux TCP. Since version 2.6.34, the Linux kernel includes the *linear timeouts* and the *thin fast retransmit* modifications we proposed as replacements for the exponential backoff and fast retransmit mechanisms in TCP. The modifications behave normally whenever a TCP stream is not thin and retransmit faster when it is thin. They are sender-side only and, thus, can be used with unmodified receivers. We have tested the mechanisms with Linux, FreeBSD, Mac OS X and Windows receivers, and all platforms successfully receive, and benefit from, the packet recovery enhancements.

Thin Fast Retransmit

TCP streams that are always busy—as they are for downloading—use fast retransmit to recover packet losses. When a sender receives three (S)ACKs for the same segment in a row, it assumes the following segment is lost and retransmits it. Segment interarrival times for thin-stream applications are very high, and in most cases, a timeout will happen before three (S)ACKs can arrive. To deal with this problem, you trigger a fast retransmission when the first duplicate (S)ACK arrives, as illustrated in Figure 2. Even if this causes a few unintended retransmissions, it leads to better latency. The overhead of this modification is minimal, because the thin stream sends very few packets anyway.

Linear Timeouts

When packets are lost and so few (S)ACKs are received by the sender that fast retransmission doesn't work, a timeout is triggered to retransmit the oldest lost

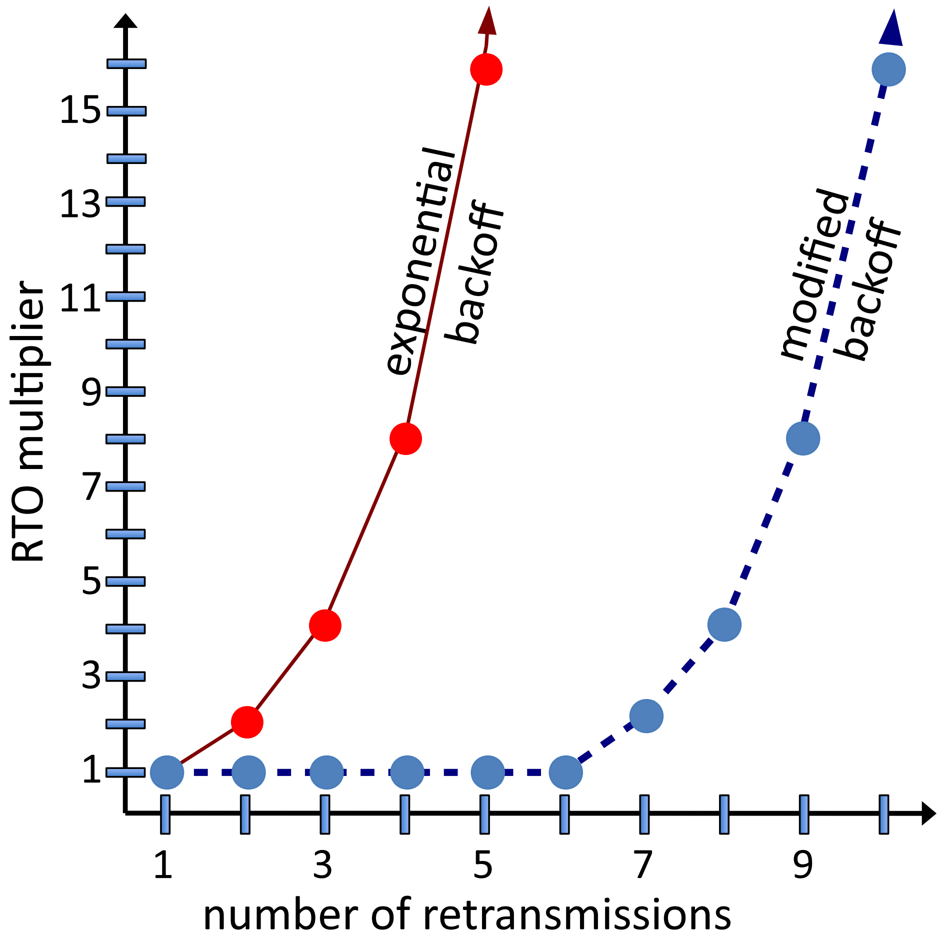


Figure 3. Modified and Standard Exponential Backoff

packet. This is not supposed to happen unless the network is heavily congested, and the retransmission timer is doubled every time it is triggered again for the same packet to avoid adding too much to the problem. When a stream is thin, these timeouts handle most packet losses simply because the application sends too little data to trigger fast transmissions. TCP doubles the timer, and latency grows exponentially when the same packet is lost several times in a row. When modification is turned on, linear timeouts are enabled when a thin stream is detected (shown in Figure 3). After six linear timeouts, exponential backoff is resumed. A packet still not recovered within this period is most likely dropped due to prevailing heavy congestion, and in that case, the linear timeout modification does not help.

Limiting Mechanism Activation

As the modifications can have a negative effect on bulk data streams (they do trigger retransmissions faster), we have implemented a test in the TCP stack to count the non-ACKed packets of a stream, and then apply the enhanced mechanisms only if a thin stream is detected. A stream is classified as thin if there are so few packets in transit that they *cannot* trigger a fast retransmission (less than four packets on the wire). Linux uses this “test” to decide when the stream is thin and, thus, when to apply the enhancements. If the test fails (the stream is able to trigger fast retransmit), the default TCP mechanisms are used. The number of dupACKs needed to trigger a fast retransmit can vary between implementations and transport protocols, but RFC 2581 advocates fast retransmit upon receiving the third dupACK. In the Linux kernel TCP implementation, "packets in transit" is an already-available variable (the `packets_out` element of the `tcp_sock` struct), and, thus, the overhead to detecting the thin-stream properties is minimal.

Enabling Thin-Stream Modifications for Your Software

The modifications are triggered dynamically based on whether the system currently identifies the stream as thin, but the mechanisms have to be enabled using switches: 1) system-wide by the administrator using syscontrol or 2) for a particular socket using I/O-control from the application.

The Administrator's View

Both the linear timeout and the thin fast retransmit are enabled using boolean switches. The administrator can set the `net.ipv4.tcp_thin_linear_timeouts` and `net.ipv4.tcp_thin_dupack` switches in order to enable linear timeout and the thin fast retransmit, respectively. As an example, linear timeouts can be configured using sysctl like this:

```
$ sysctl net.ipv4.tcp_thin_linear_timeouts=1
```

The above requires sudo or root login or using the exported kernel variables in the /proc filesystem like this:

```
$ echo "1" >  /proc/sys/net/ipv4/tcp_thin_linear_timeouts
```

(The above requires root login.)

The thin fast retransmit is enabled in a similar way using the `tcp_thin_dupack` control. If enabled in this way by the system administrator, the mechanisms are applied to *all* TCP streams of the machine, but of course, if and only if, the system identifies the stream

NOTE: If you care about thin-stream retransmission latency, there are two other socket options that you should turn on using I/O-control: 1) TCP_NODELAY disables Nagle's algorithm (delaying small packets in order to save resources by sending fewer, larger packets), and 2) TCP_QUICKACK disables the "delayed ACK" algorithm (cumulatively ACKing only every second received packet, thus saving ACKs). Both of these mechanisms reduce the feedback available for TCP when trying to figure out when to retransmit, which is especially damaging to thin-stream latency since thin streams have small packets and large intervals between each packet (see Table 1).

as thin. In this case, no modifications are required to the sending (or receiving) application.

The Application Developer's View

The thin-stream mechanisms also may be enabled on a per-socket basis by the application developer. If so, the programmer must enable the mechanism with I/O-control using the `setsockopt` system call and the `TCP_THIN_LINEAR_TIMEOUTS` and `TCP_THIN_DUPACK` option names. For example:

```
int flag = 1;
int result = setsockopt(sock, IPPROTO_TCP,
                        TCP_THIN_LINEAR_TIMEOUTS,
                        (char *) &flag, sizeof(int));
```

enables the linear timeouts. The thin fast retransmit is enabled in a similar way using the `TCP_THIN_DUPACK` option name. In this case, the programmer explicitly tells the application to use the modified TCP at the sender side, and the modifications are applied to the particular application/connection only.

OLPC | Arduino | Trim-Slice | vDSO | Elmer | MPD
LINUX JOURNAL
ONE LAPTOP PER CHILD STATUS REPORT
USE YOUR PHONE TO LOCK YOUR SCREEN AUTOMATICALLY
HACK THIS
+ HOW TO: ADD A CUSTOM VDSO TO A LINUX KERNEL
Get Started Programming the Arduino
Create a Unified Boot Environment on a USB Drive
NETWORKING

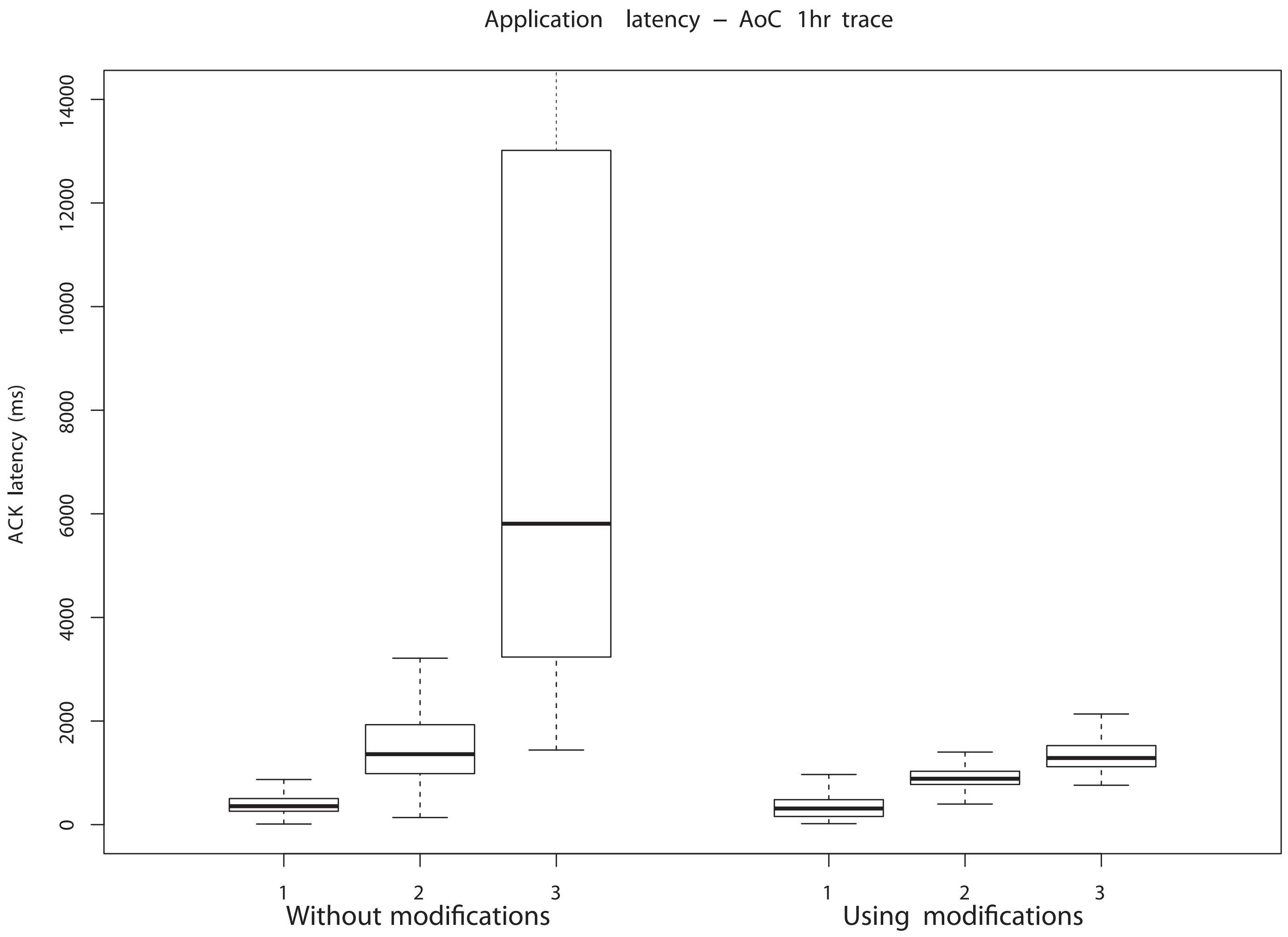


Figure 4. Modified vs. Traditional TCP in *Age of Conan*. The box shows the upper and lower quartiles and the average values. Maximum and minimum values (excluding outliers) are shown by the drawn line. The plot shows statistics for the first, second and third retransmissions.

The Mechanisms Applied in the *Age of Conan* MMORPG

We've successfully tested the thin-stream modifications for many scenarios like games, remote terminals and audio conferencing (for more information, see the thin-stream Web page listed under Resources). The example I use here to show the effect of the modifications is from a game server, a typical thin-stream application.

Funcom enabled the modifications on some of its servers running *Age of Conan*, one of its latest MMORPG games. The network traffic was captured using tcpdump. The difference in retransmission latency between the modified and the traditional TCP is shown in Figure 4.

During a one-hour capture from one of the machines in the server park, we saw more than 700 players (746 for the traditional and 722 for the modified TCP tests), where about 300 streams in each experiment experienced loss rates between 0.001% and 10%. Figure 4 shows the results from an analysis of the three first retransmissions. Having only one retransmission is fine, also when the modifications are not used. The average and worst-case latencies are still within the bounds of a playable game. However, as the users start to experience second and third retransmissions, severe latencies are observed in the traditional TCP scenario, whereas the latencies in the modified TCP test are significantly lower. Thus, the perceived quality of the game services should be greatly improved by applying the new Linux TCP modifications.

The Tools Are at Your Fingertips

If you have a kernel later than 2.6.34, the modifications are available and easy to use when you know about them. Since you now know, turn them on for your interactive thin-stream applications and remove some of the worst latencies that have been annoying you. We're currently digging deeper into thin-stream behavior—watch our blog for updates on how to reduce those latencies further.

Acknowledgements

Testing and traces by Funcom: Pal Frogner Hansen, Rui Casais and Torbjorn Linddgren. Scientific work: Carsten Griwodz and Pal Halvorsen.■

Andreas Petlund works at Simula Research Laboratory in Oslo. He finished his PhD in transport protocol latency improvements for interactive applications in 2009.

Resources

Documentation from the Linux Kernel Source: Documentation/networking/tcp-thin.txt

Thin-Stream Resource Page: **http://heim.ifi.uio.no/apetlund/thin**

Funcom Web Page: **http://www.funcom.com**

MPG Blog Page: **http://mpg.ndlab.net**

Claypool et al. "Latency and player actions in online games". *Communications of the ACM* 49, 11 (Nov. 2005), 40–45.

C. Griwodz and P. Halvorsen. "The Fun of using TCP for an MMORPG". In: *Network and Operating System Support for Digital Audio and Video* (NOSSDAV 2006), ed. by Brian Neil Levine and Mark Claypool, pp. 1–7, ACM Press (ISBN: 1-59593-285-2), 2006.

OpenLDAP Everywhere Reloaded, Part II

Now that core network services were configured in Part I, let's look at different methods for replicating the Directory between the server pair.

STEWART WALTERS

This multipart series covers how to engineer an OpenLDAP Directory Service to create a unified login for heterogeneous environments. With current software and a modern approach to server design, the aim is to reduce the number of single points of failure for the directory. In this installment, I discuss the differences between single and multi-master replication. I also describe how to configure OpenLDAP for single master replication between two servers. [See the April 2012 issue for Part I of this series or visit **http://www.linuxjournal.com/content/openldap-everywhere-reloaded-part-i**.]

On both servers, use your preferred package manager to install the slapd and ldap-utils packages if they haven't been installed already.

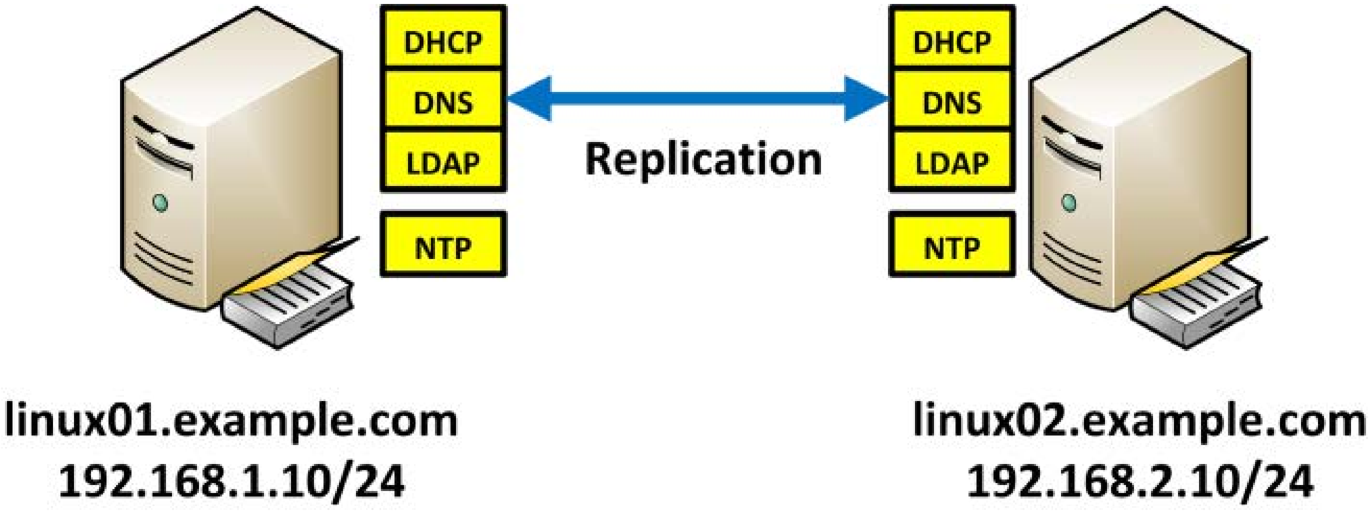


Figure 1. Example redundant server pair—in Part I of the series, NTP, DNS and DHCP were configured.

OpenLDAP 2.4 Overview

OpenLDAP 2.3 offered the start of a dynamic configuration back end to replace the traditional slapd.conf and schema files. This dynamic configuration engine (also known as cn=config) is now the default method in OpenLDAP 2.4 to store the slapd(8) configuration.

The benefits for using cn=config over traditional slapd.conf(5) are namely:

- Changes have immediate effect—you no longer need to restart slapd(8) on a production server just to make a minor ACL change or add a new schema file.

- Changes are made using LDIF files. If you already have experience with modifying LDAP using LDIF files, there is no major learning curve (other than knowing the new cn=config attributes).

OpenLDAP 2.4 still can be configured through slapd.conf(5) for now; however, this functionality may be removed from a future release of OpenLDAP. If you have an existing OpenLDAP server configured via slapd.conf, now is the time to get acquainted with cn=config.

OpenLDAP 2.4 changes the terminology in regard to replication. Replication nodes no longer are referred to as either “master” or “slave”. They are instead referred to as either a “provider” (a node that provides directory updates) or a “consumer” (a node that consumes directory updates

The benefit of MMR is that it removes the single point of failure for Directory writes.

from the provider or sometimes another consumer). The change is subtle but important to note.

In addition to LDAP Sync Replication (aka Syncrepl), which uses a Single Master Replication (SMR) model, OpenLDAP 2.4 introduces new replication types, such as N-Way Multi-Master Replication.

N-Way Multi-Master Replication, as the name suggests, uses a Multi-Master Replication (MMR) model. It is akin in operation to 389 Directory Server's replication of similar name. Multiple providers can write changes to the Directory Information Tree (DIT) concurrently.

For more information on the changes in OpenLDAP 2.4, consult the OpenLDAP 2.4 Software Administrator's Guide (see Resources).

SMR vs. MMR: Which Replication Model Is Better?

Neither replication model is better than the other per se. They both have their own benefits and drawbacks. It's really just a matter of which benefits and drawbacks are better aligned to your individual needs.

The benefit of SMR (via Syncrepl) is that it guarantees data consistency. Data will not corrupt or conflict because only one provider is allowed to make changes to the DIT. All other consumers, in effect, just make a read-only shadow copy of the DIT. Should the single provider go off-line, clients still can read from the shadow copy on the consumer.

This benefit also can be its drawback. SMR removes the single point of failure for Directory reads, but it still has the disadvantage of a single point of failure for Directory writes. If a client tries to write to the Directory when the provider is off-line, it will be unable to do so and will receive an error.

Generally speaking, this might not be a problem if the data within LDAP is very static or the outage is corrected in a relatively short amount of time. After all, a Directory by its very nature is intended to be *read from* far more than it ever will be *written to*.

But, if the provider's outage lasts for a significant amount of time, this can cause some sticky problems with account management. While the provider is unavailable, users are unable to change their expired or forgotten passwords, which might cause problems with logins. If an employee is terminated, you cannot

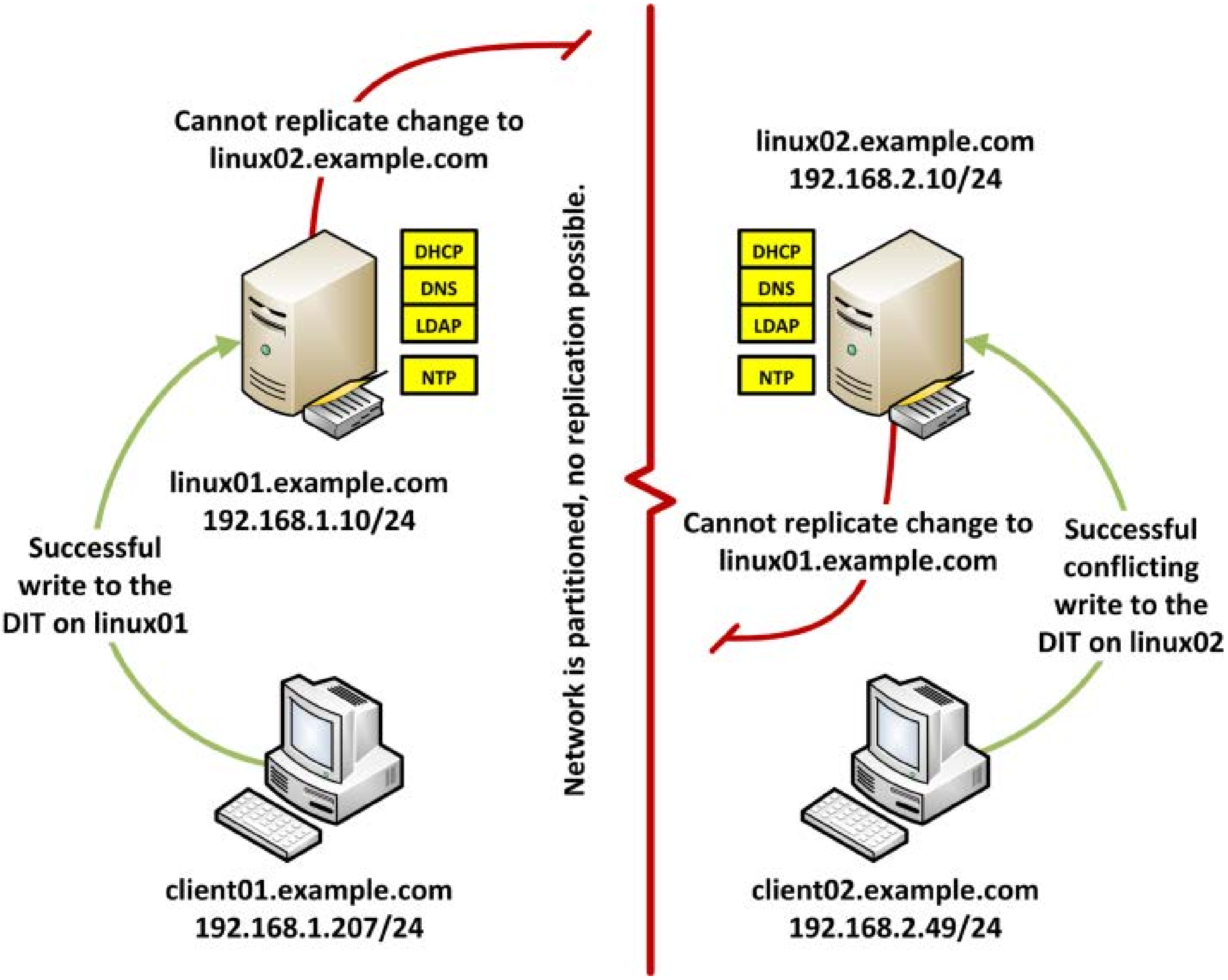

Cannot replicate change to linux02.example.com
linux02.example.com
192.168.2.10/24
DHCP
DNS
LDAP
NTP
DHCP
DNS
LDAP
NTP
linux01.example.com
192.168.1.10/24
Network is partitioned, no replication possible.
Cannot replicate change to linux01.example.com
Successful write to the DIT on linux01
Successful conflicting write to the DIT on linux02
client01.example.com
192.168.1.207/24
client02.example.com
192.168.2.49/24

Figure 2. An over-simplified view of the split-brain problem: replication fails between the two servers despite the local network still being available.

disable that person's account in LDAP until the provider is returned to service. Additionally, employees will be unable to change address-book data (although most users would not consider this an urgent problem).

The benefit of MMR is that it removes the single point of failure for Directory writes. If one provider goes off-line, the other provider(s) still can make changes to the DIT. Those changes will be replicated back to the failed provider when it comes back on-line. However, as is the case with all high-availability clusters, this can introduce what is referred to as the "split-brain" problem.

The split-brain problem is where

neither provider has failed, but network communication between the two has been disrupted. The "right side" of the split can modify the DIT blindly without consideration of what the "left side" already had changed (and vice versa). This can cause damage or corruption to the shared data store that is supposed to be consistent between both providers.

As time goes on, the two independent copies of the DIT start to diverge further and further from each other, and they become inconsistent. When the split is repaired, there is no automagic way for either provider to know which server has the truly correct copy of the DIT. At this point, a system administrator must intervene manually to repair any divergence between the two servers.

As Directories are *read from* more than they are *written to*, you may perceive the risk of divergence during split-brain to be very low. In this case, N-Way Multi-Master Replication is a good way to remove the single point of failure for Directory writes.

On the other hand, the single point of failure for Directory writes may be

only a minor nuisance if you can avoid the hassles of data inconsistency. In this case, Syncrepl is the better option.

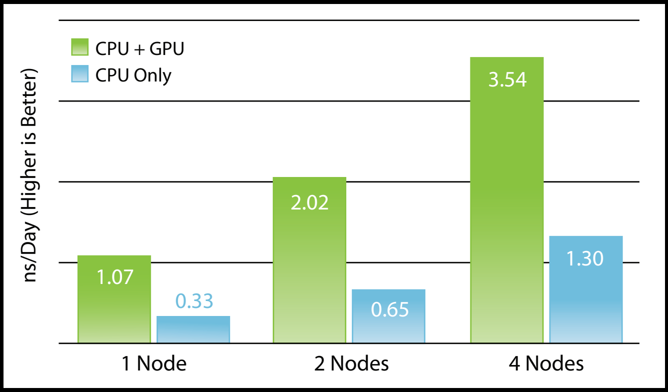
It's all a matter of which risk you perceive to have a bigger impact on your organization. You'll need to make an assessment as to which of the two replication methods is more appropriate, then implement one or the other—*but not both*!

Initial Configuration of slapd after Installation

After Debian installs the slapd package, it asks you for the "Administrator" password. It preconfigures the Directory Information Tree (DIT) with a top-level namespace of dc=nodomain if getdomainname(2) was not configured locally. The RootDN becomes `cn=admin,dc=nodomain`, which is a Debian-ism and a departure from OpenLDAP's default of `cn=Manager,$BASEDN`.

dc=nodomain is not actually useful though. The Debian OpenLDAP maintainers essentially leave it up to the user to re-create a more appropriate namespace.

You can delete the dc=nodomain



The question about “DNS domain name” has nothing to do with DNS; it is a Debian-ism.

DIT and start again with the `dpkg-reconfigure slapd` command. Run this on both linux01.example.com and linux02.example.com. The reconfigure scripts for the slapd package will ask you some questions. I’ve provided the answers I used as an example. Of course, select more appropriate values where you see fit:

```
"Omit OpenLDAP server configuration" = No
"DNS domain name" = example.com
"Organisation name" = Example Corporation
"Administrator password" = linuxjournal
"Confirm Administrator password" = linuxjournal
"Database backend to use" = HDB
"Do you want the database to be removed when slapd is purged?" = No
"Move old database?" = Yes
"Allow LDAPv2 protocol?" = No
```

The question about “DNS domain name” has nothing to do with DNS; it is a Debian-ism. The answer supplied as a domain name will be converted to create the top-level namespace ($BASEDN) of the DIT. For example, if you intend to use `dc=pixie,dc=dust` as your top-level namespace, enter `pixie.dust` for the answer.

The questions about “Administrator password” refer to the OpenLDAP RootDN password, aka RootPW, aka olcRootPW. Here you will set the password for the cn=admin,$BASEDN account, which in this example is `cn=admin,dc=example,dc=com`.

If you run the slapcat(8) command, it now shows a very modest DIT, with only dc=example,dc=com and cn=admin,dc=example,dc=com populated.

OpenLDAP by default (for performance reasons) does not log a large amount information to syslog(3). You might want to increase OpenLDAP’s log levels to assist the diagnosis of any replication problems that occur:

```
# set_olcLogLevel.ldif
#
# Run on linux01 and linux02
#
dn: cn=config
changetype: modify
replace: olcLogLevel
olcLogLevel: acl stats sync
```

Modify cn=config on both servers with the `ldapmodify -Q -Y EXTERNAL -H ldapi:/// -f set_olcloglevel.ldif` command to make this change effective.

Option 1: Single Master Replication (Using Syncrepl)

If you have chosen to use LDAP Sync Replication (Syncrepl), the instructions below demonstrate a way to replicate dc=example,dc=com between both servers using one provider (linux01.example.com) and one consumer (linux02.example.com).

As Syncrepl is a *consumer-side* replication engine, it requires the consumer to bind to the provider with a security object (an account) to complete its replication operations.

To create a new security object on linux01.example.com, create a new text file called smr_create_security_object.ldif, and populate it as follows:

```
# smr_create_security_object.ldif
#
# Run on linux01
#
# 1. Create an OU for all replication accounts
dn: ou=Replicators,dc=example,dc=com
description: Security objects (accounts) used by
 Consumers that will replicate the DIT.
objectclass: organizationalUnit
objectclass: top
ou: Replicators

# 2. Create security object for linux02.example.com
dn: cn=linux02.example.com,ou=Replicators,dc=example,dc=com
cn: linux02.example.com
description: Security object used by linux02.example.com
 for replicating dc=example,dc=com.
objectClass: simpleSecurityObject
objectClass: organizationalRole
userPassword: {SSHA}qzhCiuIJb3NVJcKoy8uwHD8eZ+IeU5iy
# userPassword is 'linuxjournal' in encrypted form.
```

The encrypted password was obtained with the `slappasswd -s <password>` command. Use ldapadd(1) to add the security object to dc=example,dc=com:

```
root@linux01:~# ldapadd -x -W -H ldapi:/// \
> -D cn=admin,dc=example,dc=com \
> -f smr_create_security_object.ldif
Enter LDAP Password:
adding new entry "ou=Replicators,dc=example,dc=com"

adding new entry "cn=linux02.example.com,ou=
➥Replicators,dc=example,dc=com"

root@linux01:~#
```

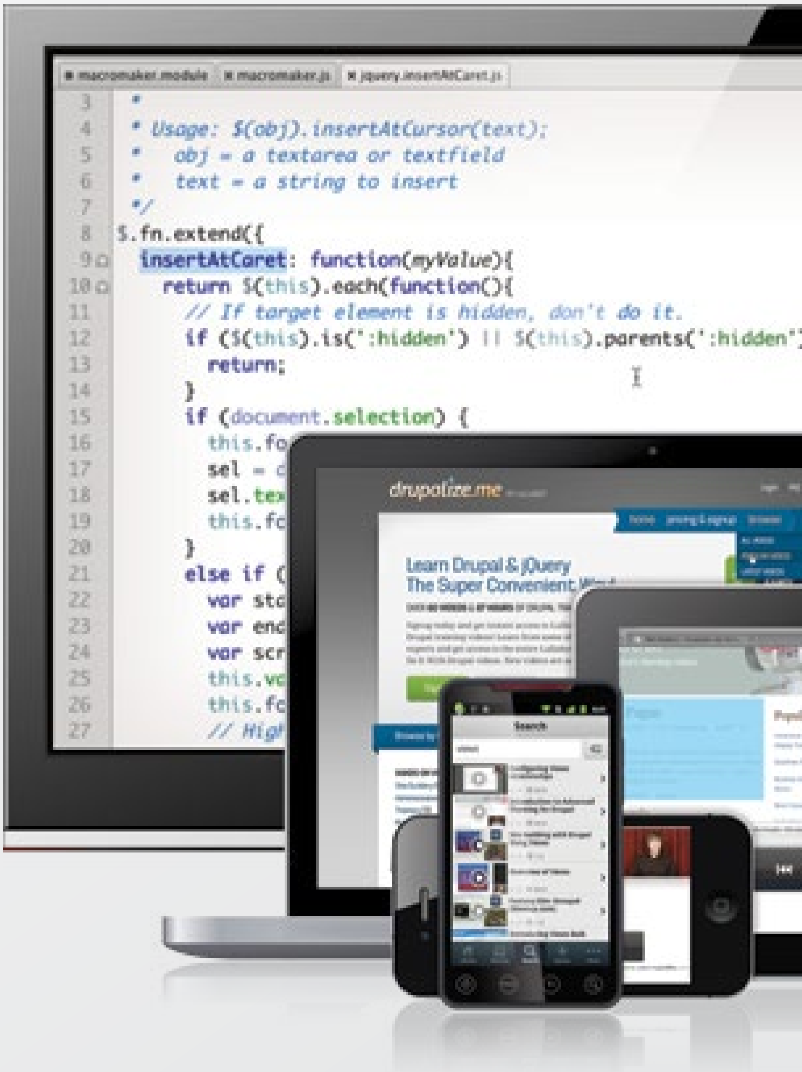
If you encounter an error, there may be a typographical error in the LDIF file. Be careful to note lines that are broken with a single preceding space on the second line. If in doubt, see the Resources section for a copy of smr_create_security_object.ldif.

Run slapcat(8) to show the security object and the OU it's contained by.

On linux01.example.com, create a new text file called smr_set_dcexample_provider.ldif, and populate it as follows:

```
# smr_set_dcexample_provider.ldif
#
# Run on linux01
#
# 1. Load the Sync Provider (syncprov) Module
```

```
dn: cn=module{0},cn=config
changetype: modify
add: olcModuleLoad
olcModuleLoad: syncprov

# 2. Enable the syncprov overlay on
#    dc=example,dc=com
dn: olcOverlay=syncprov,olcDatabase={1}hdb,cn=config
changetype: add
objectClass: olcOverlayConfig
objectClass: olcSyncProvConfig
olcOverlay: syncprov
olcSpCheckpoint: 100 10
olcSpSessionlog: 100
# olcSpCheckpoint (syncprov-checkpoint) every 100
#   operations or every 10 minutes, whichever is
#   first
# olcSpSessionlog (syncprov-sessionlog) maximum
#   100 session log entries

# 3.1.1. Delete the existing ACL for
#        userPassword/shadowLastChange
dn: olcDatabase={1}hdb,cn=config
changetype: modify
delete: olcAccess
olcAccess: {0}to attrs=userPassword,shadowLastChange
  by self write
  by anonymous auth
  by dn="cn=admin,dc=example,dc=com" write
  by * none
-
# 3.1.2. Add a new ACL to allow the replication
#        security object read access to
#        userPassword/shadowLastChange
add: olcAccess
olcAccess: {0}to attrs=userPassword,shadowLastChange
  by self write
  by anonymous auth
  by dn="cn=admin,dc=example,dc=com" write
  by dn="cn=linux02.example.com,ou=Replicators,dc=ex
➥ample,dc=com" read
  by * none
-
# 3.2. Indices can speed searches up. Though, every
#      index used, adds to slapd's memory
#      requirements
add: olcDbIndex
#
# Required indices
olcDbIndex: entryCSN eq
olcDbIndex: entryUUID eq
#
# Not quite required, not quite optional. The logs
# fill up without this index present
olcDbIndex: uid pres,sub,eq
#
# Optional indices
olcDbIndex: cn pres,sub,eq
olcDbIndex: displayName pres,sub,eq
olcDbIndex: givenName pres,sub,eq
olcDbIndex: mail pres,eq
olcDbIndex: sn pres,sub,eq
#
# Debian already includes an index for
# objectClass eq, which is also a requirement
-
# 3.3. Allow Replicator account limitless searches
add: olcLimits
olcLimits: dn.exact="cn=linux02.example.com,ou=Repli
 cators,dc=example,dc=com"
  time.soft=unlimited
  time.hard=unlimited
  size.soft=unlimited
  size.hard=unlimited
```



```
 *
 * Usage: $(obj).insertAtCursor(text);
 *   obj = a textarea or textfield
 *   text = a string to insert
 */
$.fn.extend({
  insertAtCaret: function(myValue){
    return $(this).each(function(){
      // If target element is hidden, don't do it.
      if ($(this).is(':hidden') || $(this).parents(':hidden'
        return;
      }
      if (document.selection) {
```

When this LDIF file is applied, it will tell slapd(8) to load the syncprov (Sync Provider) module and will enable the syncprov overlay on the database that contains dc=example,dc=com. It will modify Debian's default password ACL to allow the newly created security object read access (so it can replicate passwords to linux02.example.com). It also adds some required and optional indices, and removes any time and size limits for the security object (so as not to restrict it when it queries linux01.example.com).

Apply this LDIF file on linux01.example.com with ldapmodify(1) as follows:

```
root@linux01:~# ldapmodify -Q -Y EXTERNAL \
> -H ldapi:/// \
> -f smr_set_dcexample_provider.ldif
modifying entry "cn=module{0},cn=config"

adding new entry "olcOverlay=syncprov,olcDatabase={1}hdb,cn=config"

modifying entry "olcDatabase={1}hdb,cn=config"

root@linux01:~#
```

Again, if there are errors, they could be typographical errors. Be sure to note which lines in the file are broken with a preceding single space or a preceding double space. Also, be sure to note which sections are separated with a blank line and which are separated with a single dash (-) character. If in doubt, see the Resources section for a copy of smr_set_dcexample_provider.ldif.

Now, on linux02.example.com, create a text file called smr_set_dcexample_consumer.ldif, and populate it with the following:

```
# smr_set_dcexample_consumer.ldif
#
# Run on linux02
#
# 1.1.
dn: olcDatabase={1}hdb,cn=config
changetype: modify
add: olcSyncRepl
olcSyncRepl: rid=001
  provider=ldap://linux01.example.com/
  type=refreshAndPersist
  retry="5 6 60 5 300 +"
  searchbase="dc=example,dc=com"
  schemachecking=off
  bindmethod=simple
  binddn="cn=linux02.example.com,ou=Replicators,dc=example,dc=com"
  credentials=linuxjournal
# retry every 5 seconds for 6 times (30 seconds),
#  then every 60 seconds for 5 times (5 minutes)
#  then every 300 seconds (5 minutes) thereafter
# schemachecking=off as checking gets done on
#  linux01. we do not want records received from
#  linux01 ignored because they fail the ill-
#  defined (or missing) schemas on linux02.
-
# 1.2.1. Delete the existing ACL for
#        userPassword/shadowLastChange
delete: olcAccess
olcAccess: {0}to attrs=userPassword,shadowLastChange
  by self write
  by anonymous auth
```

21st USENIX SECURITY SYMPOSIUM

Bellevue, WA • August 8–10, 2012

The USENIX Security Symposium brings together researchers, practitioners, system administrators, system programmers, and others interested in the latest advances in the security of computer systems and networks.

USENIX Security '12 will feature:

Keynote Address given by:

- **Dickie George**, Johns Hopkins Applied Physics Laboratory

A 3-day Technical Program including:

More than 40 refereed papers covering the latest research, including topics such as browser security, privacy enhancing technologies, and botnets and Web security. Plus:

- Invited Talks
- Panel Discussions
- Poster Session
- Rump Session
- Birds-of-a-Feather sessions (BoFs)

Stay Connected...

http://www.usenix.org/facebook

http://twitter.com/USENIXSecurity

Co-Located Workshops Include:

EVT/WOTE '12: 2012 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections
August 6–7, 2012

HealthSec '12: 3rd USENIX Workshop on Health Security and Privacy
August 6–7, 2012

WOOT '12: 6th USENIX Workshop on Offensive Technologies
August 6–7, 2012

CSET '12: 5th Workshop on Cyber Security Experimentation and Test
August 6, 2012

FOCI '12: 2nd USENIX Workshop on Free and Open Communications on the Internet
August 6, 2012

HotSec '12: 7th USENIX Workshop on Hot Topics in Security
August 7, 2012

MetriCon 7.0: Seventh Workshop on Security Metrics
August 7, 2012

Register by July 18 and Save!

www.usenix.org/sec12

```
  by dn="cn=admin,dc=example,dc=com" write
  by * none
-
# 1.2.2. Add a new ACL which removes all write
#        access
add: olcAccess
olcAccess: {0}to attrs=userPassword,shadowLastChange
  by anonymous auth
  by * none
-
# 1.3.1. Delete the existing ACL for *
delete: olcAccess
olcAccess: {2}to *
  by self write
  by dn="cn=admin,dc=example,dc=com" write
  by * read
-
# 1.3.2. Add a new ACL for * removing all write
#        access
add: olcAccess
olcAccess: {2}to *
  by * read
-
# 1.4. Indices can speed searches up. Though, every
#      index used, adds to slapd's memory
#      requirements
add: olcDbIndex
#
# Required indices
olcDbIndex: entryCSN eq
olcDbIndex: entryUUID eq
#
# Not quite required, not quite optional. The logs
# fill up without this index present
olcDbIndex: uid pres,sub,eq
#
# Optional indices
olcDbIndex: cn pres,sub,eq
olcDbIndex: displayName pres,sub,eq
olcDbIndex: givenName pres,sub,eq
olcDbIndex: mail pres,eq
olcDbIndex: sn pres,sub,eq
#
# Debian already includes an index for
# objectClass eq, which is also a requirement
-
# 1.5. If a LDAP client attempts to write changes
#      on linux02, linux02 will return with a
#      referral error telling the client to direct
#      the change at linux01 instead.
add: olcUpdateRef
olcUpdateRef: ldap://linux01.example.com/
-
# 1.6.1. Rename cn=admin to cn=manager.
#        Modifications are only made by linux01
replace: olcRootDN
olcRootDN: cn=manager
-
# 1.6.2. Remove the local olcRootPW. Modifications
#        are only made on linux01
delete: olcRootPW
```

When this LDIF file is applied, it configures slapd(8) to use LDAP Sync Replication (olcSyncRepl) to replicate from linux01.example.com. It authenticates with the newly created security object. As this is a read-only copy of dc=example,dc=com, it replaces two existing ACLs with ones that remove all write access. It adds some required and optional indices, adds a referral URL for linux01.example.com and (in effect) cripples the RootDN on linux02.example.com (because no modifications to the DIT will occur here).

Apply smr_set_dcexample_consumer.ldif on linux02.example.com with ldapmodify(1) as follows:

```
root@linux02:~# ldapmodify -Q -Y EXTERNAL \
> -H ldapi:/// \
> -f smr_set_dcexample_consumer.ldif
modifying entry "olcDatabase={1}hdb,cn=config"

root@linux02:~#
```

Finally, on linux02.example.com, stop slapd(8), delete the database files created by the `dpkg-reconfigure slapd` command run earlier, and restart slapd(8). This will allow slapd(8) to regenerate the database files in light of the new configuration:

```
root@linux02:~# /etc/init.d/slapd stop
Stopping OpenLDAP: slapd.
root@linux02:~# rm /var/lib/ldap/*
root@linux02:~# /etc/init.d/slapd start
Starting OpenLDAP: slapd.
root@linux02:~#
```

To show that the replication works,

Sinatra | HTML5 | Catalyst | Drupal 7 | Moodle
LINUX JOURNAL
INTRO TO THE SINATRA MICRO-FRAMEWORK
THE BEST WAY TO CLIENT CONCURRENCY
WEB DEVELOPMENT
CREATE Web Apps with Catalyst and Perl
NEW COLUMN: THE OPEN-SOURCE CLASSROOM

To show that the replication works, you can add something to the DIT on linux01.example.com and use slapcat(8) on linux02.example.com to see if it arrives there.

you can add something to the DIT on linux01.example.com and use slapcat(8) on linux02.example.com to see if it arrives there.

Create a text file on linux01.example.com called set_dcexample_test.ldif, and populate it with some dummy records:

```
# set_dcexample_test.ldif
#
# Run on linux01
#
dn: ou=People,dc=example,dc=com
description: Testing dc=example,dc=com replication
objectclass: organizationalUnit
objectclass: top
ou: People

dn: ou=Soylent.Green.is,ou=People,dc=example,dc=com
description: Chuck Heston would be proud
objectclass: organizationalUnit
ou: Soylent.Green.is
```

Use ldapadd(1) to add the entries to the DIT:

```
root@linux01:~# ldapadd -x -W -H ldapi:/// \
> -D cn=admin,dc=example,dc=com \
> -f set_dcexample_test.ldif
Enter LDAP Password:
adding new entry "ou=People,dc=example,dc=com"

adding new entry "ou=Soylent.Green.is,ou=People,
➥dc=example,dc=com"

root@linux01:~#
```

On linux02.example.com, use slapcat(8) to see that the records are present:

```
root@linux02:~# slapcat | grep -i soylent
dn: ou=Soylent.Green.is,ou=People,dc=example,dc=com
ou: Soylent.Green.is
root@linux02:~#
```

On linux01.example.com, create a new text file called unset_dcexample_test.txt, and populate it as follows:

```
ou=Soylent.Green.is,ou=People,dc=example,dc=com
ou=People,dc=example,dc=com
```

Use the command `ldapdelete -x -W -H ldapi:/// -D cn=admin,dc=example,dc=com -f unset_dcexample_test.txt` to delete the test entries.

A Few Last Things

Once replication is working properly between the two servers, you should remove the change to the logging level (olcLogLevel) performed earlier, so that queries to LDAP do not affect server performance.

On both linux01.example.com and linux02.example.com create a text file called unset_olcLogLevel.ldif, and populate it as follows:

```
# unset_olcLogLevel.ldif
#
# Run on linux01 and linux02
#
dn: cn=config
changetype: modify
delete: olcLogLevel
```

Then, use it to remove olcLogLevel with the `ldapmodify -Q -Y EXTERNAL -H ldapi:/// -f unset_olcLogLevel.ldif` command.

Also, configure the LDAP clients to point at the LDAP servers. Modify /etc/ldap/ldap.conf on both servers, and add the following two lines:

```
BASE    dc=example,dc=com
URI     ldap://linux01.example.com/ ldap://linux02.example.com/
```

If you opted for MMR, use the above two lines for /etc/ldap/ldap.conf on linux01.example.com only. On linux02.example.com, use the following two lines instead:

```
BASE    dc=example,dc=com
URI     ldap://linux02.example.com/ ldap://linux01.example.com/
```

I'll continue this in Part III of this series, where I describe how to configure the two OpenLDAP servers to replicate using N-Way Multi-Master Replication instead.■

Stewart Walters is a Solutions Architect with more than 15 years' experience in the Information Technology industry. Among other industry certifications, he is a Senior-Level Linux Professional (LPIC-3). Where possible, he tries to raise awareness of the "Parkinson-Plus" syndromes, such as crippling neurodegenerative diseases like Progressive Supranuclear Palsy (PSP) and Multiple System Atrophy (MSA). He can be reached for comments at stewart.walters@googlemail.com.

Resources

Example Configuration Files for This Article: **http://ftp.linuxjournal.com/pub/lj/listings/issue218/11292.tgz**

"OpenLDAP Everywhere Reloaded, Part I" by Stewart Walters, *LJ*, April 2012: **http://www.linuxjournal.com/content/openldap-everywhere-reloaded-part-i**

OpenLDAP Release Road Map: **http://www.openldap.org/software/roadmap.html**

OpenLDAP Software 2.4 Administrator's Guide: **http://www.openldap.org/doc/admin24**

Chapter 18: "Replication—from OpenLDAP Software 2.4 Administrator's Guide": **http://www.openldap.org/doc/admin24/replication.html**

Appendix A: "Changes Since Previous Release"—from OpenLDAP Software 2.4 Administrator's Guide: **http://www.openldap.org/doc/admin24/appendix-changes.html**

OpenLDAP Technical Mailing List: **http://www.openldap.org/lists/mm/listinfo/openldap-technical**

OpenLDAP Technical Mailing List Archives Interface: **http://www.openldap.org/lists/openldap-technical**

LDAP Data Interchange Format Wikipedia Page: **http://en.wikipedia.org/wiki/LDAP_Data_Interchange_Format**

RFC2849—The LDAP Data Interchange Format (LDIF)—Technical Specification: **http://www.ietf.org/rfc/rfc2849**

Internet Draft—Using LDAP Over IPC Mechanisms: **http://tools.ietf.org/html/draft-chu-ldap-ldapi-00**

OpenLDAP Consumer on Debian Squeeze: **http://www.rjsystems.nl/en/2100-d6-openldap-consumer.php**

OpenLDAP Provider on Debian Squeeze: **http://www.rjsystems.nl/en/2100-d6-openldap-provider.php**

OpenLDAP Server from the Ubuntu Official Documentation: **https://help.ubuntu.com/11.04/serverguide/C/openldap-server.html**

Samba 2.0 Wiki: Configuring LDAP: **http://wiki.samba.org/index.php/2.0:_Configuring_LDAP#2.2.2._slapd.conf_Master_delta-syncrepl_Openldap2.3**

Zarafa LDAP cn config How To: **http://www.zarafa.com/wiki/index.php/Zarafa_LDAP_cn_config_How_To**

Man Page for getdomainname(2): **http://linux.die.net/man/2/getdomainname**

Man Page for ldapadd(1): **http://linux.die.net/man/1/ldapadd**

Man Page for ldapdelete(1): **http://linux.die.net/man/1/ldapdelete**

Man Page for ldapmodify(1):
http://linux.die.net/man/1/ldapmodify

Man Page for ldif(5):
http://linux.die.net/man/5/ldif

Man Page for slapcat(8):
http://linux.die.net/man/8/slapcat

Man Page for slapd(8):
http://linux.die.net/man/8/slapd

Man Page for slapd.access(5):
http://linux.die.net/man/5/slapd.access

Man Page for slapd.conf(5):
http://linux.die.net/man/5/slapd.conf

Man Page for slapd.overlays:
http://linux.die.net/man/5/slapd.overlays

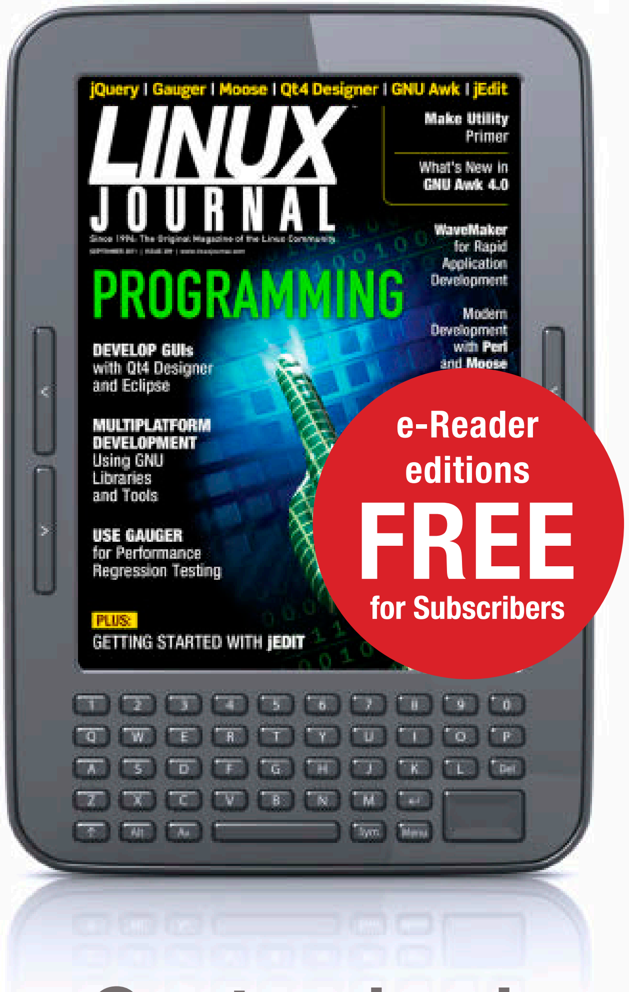
Man Page for slapd-config(5):
http://linux.die.net/man/5/slapd-config

Man Page for slapo-syncprov(5):
http://linux.die.net/man/5/slapo-syncprov

Man Page for slapindex(8):
http://linux.die.net/man/8/slapindex

Man Page for slappasswd(8):
http://linux.die.net/man/8/slappasswd

Man Page for syslog(3):
http://linux.die.net/man/3/syslog


jQuery | Gauger | Moose | Qt4 Designer | GNU Awk | jEdit
LINUX JOURNAL
PROGRAMMING
Make Utility Primer
What's New in GNU Awk 4.0
WaveMaker for Rapid Application Development
Modern Development with Perl and Moose
DEVELOP GUIs with Qt4 Designer and Eclipse
MULTIPLATFORM DEVELOPMENT Using GNU Libraries and Tools
USE GAUGER for Performance Regression Testing
PLUS: GETTING STARTED WITH JEDIT

DOC SEARLS

What's Your Data Worth?

Your personal data has more use value than sale value. So what's the real market for it?

We all know that our data trails are being hoovered up by Web sites and third parties, mostly as grist for advertising mills that put cross hairs for "personalized" messages on our virtual backs. Since the mills do pay for a lot of that data, there is a market for it—just not for you and me. It's a B2B thing, Business to Business. We're in the C category: Consumers. But the fact that our data is being paid for, and that we are the first-source producers of that data, raises a question: can't we get in on this action?

In his RealTea blog (**http://www.realtea.net**), Gam Dias notes that this question has been asked for at least a decade, and he provides a chronology, which I'll compress here:

- In 2002, Chris Downs, a designer and co-founder of Live|Work, auctioned 800 pages of personal information on eBay. *Businessweek* covered it in "Wanna See My Personal Data? Pay Up" (**http://www.businessweek.com/technology/content/nov2002/tc20021121_8723.htm**). (Chris' data sold for £150 to another designer rather than an advertiser.)

- In 2003, John Deighton, a professor at Harvard Business School, published "Market Solutions to Privacy Problems?" (**http://www.hbs.edu/research/facpubs/workingpapers/abstracts/0203/03-024.html**). An HBS interview followed (**http://hbswk.hbs.edu/item/3636.html**). One pull-quote: "The solution is to create institutions that allow consumers to build and claim the value of their marketplace identities, and that give producers the incentive to respect them."

- In 2006, Dennis D. McDonald published "Should We Be Able to Buy and Sell Our Personal Financial and Medical Data?" (**http://www.ddmcd.com/personal_data_ownership.html**). "The idea is that you own your personal data and you alone have the right to make it public and to earn money from business transactions based on that data", he wrote. Therefore, he continued, "You should even be able to auction off to the highest bidder your most intimate and personal details, if you so desire." Also in 2006, Kablenet published "Sell Your Personal Data and Receive Tax Cuts" in *The Register* (**http://www.theregister.co.uk/2006/10/04/data_sales_for_tax_cuts/print.html**).

- In 2007, somebody called "highlytargeted" auctioned off "non-personally identifiable information to help you better target ads to me". According to Gam, "the package included the past 30 days' Internet search queries, past 90 days' Web surfing history, past 30 days' on-line and off-line purchase activity, Age, Gender, Ethnicity, Marital status and Geo location and the right to target one e-mail ad per day to me for 30 days." Also in 2007, Iain Henderson, now of The Customer's Voice, published "Can I Own My Data?" (**http://rightsideup.blogs.com/my_weblog/2007/10/can-i-own-my-da.html**) on the Right Side Up blog. Wrote Iain, "...the point at which I will 'own' my personal data is the point at which I can actively manage it. If I have the choice over whether to sell it to someone, and can cover that sale with a standard commercial contract, then I clearly have title. But—and this is crucial—this doesn't mean that I 'own' all the personal data that relates to me. Lots of it will still

But the fact that our data is being paid for, and that we are the first-source producers of that data, raises a question: can't we get in on this action?

be lying around in various supplier operational systems that I won't have access to (and probably don't want to—much of it is not worth me bothering about)."

- In 2011, Julia Angwin and Emily Steel published "Web's Hot New Commodity: Privacy" (**http://online.wsj.com/article/SB10001424052748703529004576160764037920274.html**) in *The Wall Street Journal*, as part of that paper's "What They Know" series, which began on July 31, 2010—a landmark event I heralded in "The Data Bubble" (**http://blogs.law.harvard.edu/doc/2010/07/31/the-data-bubble**) and "The Data Bubble II" (**http://blogs.law.harvard.edu/doc/2010/10/31/the-data-bubble-ii**). Joel Stein also published "Data Mining: How Companies Now Know Everything About You" (**http://www.time.com/time/magazine/article/0,9171,2058205,00.html**), in *Time*.

The most influential work on the subject in 2011 was "Personal Data: The Emergence of a New Asset Class" (**http://www.time.com/time/magazine/article/0,9171,2058205,00.html**), a (.pdf) paper published by the World Economic Forum. While the paper focused broadly on economic opportunities, the word "asset" in its title suggested fungibility, which loaned weight to dozens of other pieces, all making roughly the same case: that personal data is a sellable asset, and, therefore, the sources of that data should be able to get paid for it.

For example, in "A Stock Exchange for Your Personal Data" (**http://www.technologyreview.com/computing/40330/?p1=MstRcnt**), on May 1 of this year, Jessica Leber of MIT's *Technology Review* visited a research paper titled "A Market for Unbiased Private Data: Paying Individuals According to Their Privacy Attitudes" (**http://www.hpl.hp.com/research/scl/papers/datamarket/datamarket.pdf**), written by Christina Aperjis and Bernardo A. Huberman, of HP Labs' Social Computing Group. Jessica said the paper proposed "something akin to a New York Stock Exchange for personal data. A trusted market operator could take a small cut of each transaction and help arrive at a realistic price for a sale." She went on to explain:

> On this proposed market, a person who highly values her privacy might choose an option to sell her shopping patterns for $10, but at a big risk of not

> finding a buyer. Alternately, she might sell the same data for a guaranteed payment of 50 cents. Or she might opt out and keep her privacy entirely.
>
> You won't find any kind of opportunity like this today. But with Internet companies making billions of dollars selling our information, fresh ideas and business models that promise users control over their privacy are gaining momentum. Startups like Personal and Singly are working on these challenges already. The World Economic Forum recently called an individual's data an emerging "asset class".

Naturally, HP Labs is filing for a patent on the model.

In "How A Private Data Market Could Ruin Facebook" (**http://www.hpl.hp.com/research/scl/papers/datamarket/datamarket.pdf**), also in *Technology Review*, MTK wrote, "The issue that concerns many Facebook users is this. The company is set [to] profit from selling user data, but the users whose data is being traded do not get paid at all. That seems unfair." After sourcing Jessica Leber's earlier piece, MTK added, "Setting up a market for private data

Advertiser Index

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

ADVERTISER	URL	PAGE #
1&1	http://www.1and1.com	25
Emac, Inc.	http://www.emacinc.com	39
EmperorLinux	http://www.emperorlinux.com	27
iXsystems, Inc.	http://www.ixsystems.com	7
JaxConf	http://www.jaxconf.com	73
LinuxCon North America	https://events.linuxfoundation.org/events/linuxcon	81
Lullabot	http://www.lullabot.com	101
Microway, Inc.	http://www.microway.com	96, 97
O'Reilly Oscon	http://www.oscon.com/oscon2012	2
Silicon Mechanics	http://www.siliconmechanics.com	3
Texas Linux Fest	http://texaslinuxfest.org	35
Usenix Security	http://static.usenix.org/event/sec12/	103

won't be easy", and gave several reasons, ending with this:

> Another problem is that the idea fails if a significant fraction of individuals choose to opt out altogether because the samples will then be biased towards those willing to sell their data. Huberman and Aperjis say this can be prevented by offering a high enough base price. Perhaps.

Think about the sum of personal data on all your computer drives, plus whatever you have on paper and in other media, including your own head.

> Such a market has an obvious downside for companies like Facebook which exploit individuals' private data for profit. If they have to share their profit with the owners of the data, there is less for themselves. And since Facebook will struggle to achieve the kind of profits per user it needs to justify its valuation, there is clearly trouble afoot.
>
> Of course, Facebook may decide on an obvious way out of this conundrum—to not pay individuals for their data. But that creates an interesting gap in the market for a social network that does pay a fair share to its users (perhaps using a different model [than] Huberman and Aperjis').
>
> Is it possible that such a company could take a significant fraction of the market? You betcha! Either way, Facebook loses out—it's only a question of when.

All of these arguments are made inside an assumption: that the value of personal data is best measured in money.

Sound familiar?

To me this is partying like it's 1999. That was when Eric S. Raymond published *The Magic Cauldron* (**http://www.catb.org/~esr/writings/homesteading/magic-cauldron**), in which he visited "the mixed economic context in which most open-source developers actually operate". In the chapter "The Manufacturing Delusion" (**http://www.catb.org/~esr/writings/homesteading/magic-cauldron**), he begins:

> We need to begin by noticing that computer programs, like all other kinds of tools or capital goods, have two distinct kinds of economic value. They have use value and sale value.
>
> The use value of a program is its economic value as a tool, a productivity multiplier. The sale value of a program is its value as a salable commodity. (In professional economist-speak, sale value is value as a final good, and use value is value as an intermediate good.)
>
> When most people try to reason about software-production economics, they tend to assume a "factory model"....

That's where we are with all this talk about selling personal data.

Even if there really is a market there, there isn't an industry, as there is with software. Hey, Eric might be right when he says, a few paragraphs later, "the software industry is largely a service industry operating under the persistent but unfounded delusion that it is a manufacturing industry." But that delusion is still a many-dozen $billion market.

My point is that we're forgetting the lessons that free software and open source have been teaching from the start: that we shouldn't let sale value obscure our view of use value—especially when the latter has far more actual leverage.

Think about the sum of personal data on all your computer drives, plus whatever you have on paper and in other media, including your own head. Think about what that data is worth to you—not for sale, but for use in your own life. Now think about the data trails you leave on the Web. What percentage of your life is that? And why sell it if all you get back is better guesswork from advertisers, and offers of discounts and other enticements from merchants?

Sale value is easy to imagine, and to project on everything. But it rests on a foundation of use value that is much larger and far more important. Here in the Linux world that fact is obvious. But in the world outside it's not. Does that mean we need to keep playing whack-a-mole with the manufacturing delusion? I think there's use value in it, or I wouldn't be doing it now. Still, I gotta wonder.■

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.