LINUX™
JOURNAL

Since 1994: The Original Magazine of the Linux Community

MAKE
SITES LOOK
WITH BOOTS

A GUIDE TO 3-D PRINTER HARDWARE

COOL PROJECTS

CREATE YOUR OWN CLOUD-BASED STORAGE SERVICE WITH OWNCLOUD

ANIMATION AND VIDEO TECHNIQUES FOR LINUX

INTERFACING DISPARATE SYSTEMS FOR HOME AUTOMATION

BUILD A WALL-MOUNTED DUAL-HEAD POWER-USER DESKTOP
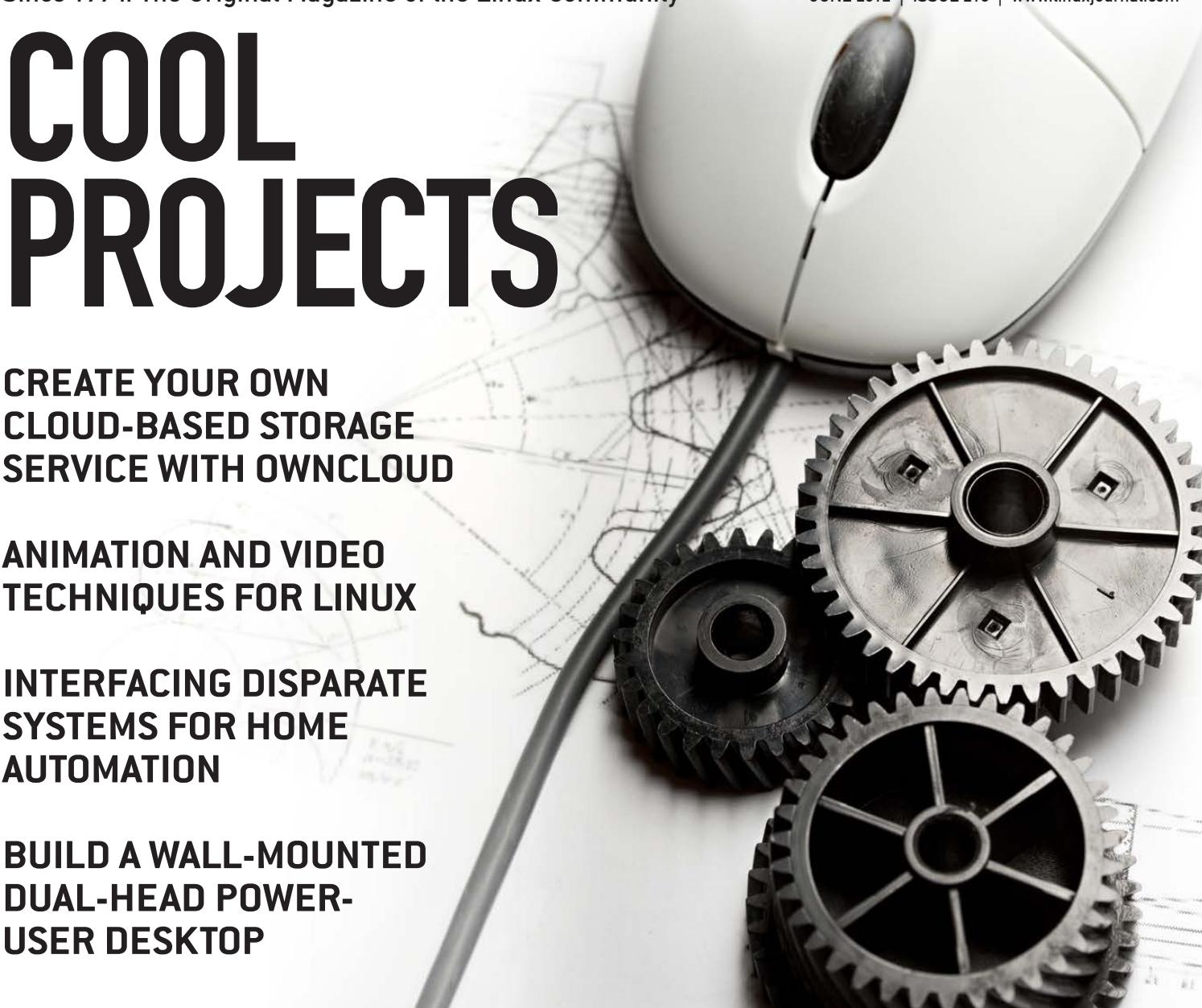
HOW-TO:
Replace Proprietary Servers with Linux

REVIEWED:
Polywell's i2303

REVOLUTIONIZE
**Your Storage Strategies with ZFS and Btrfs**
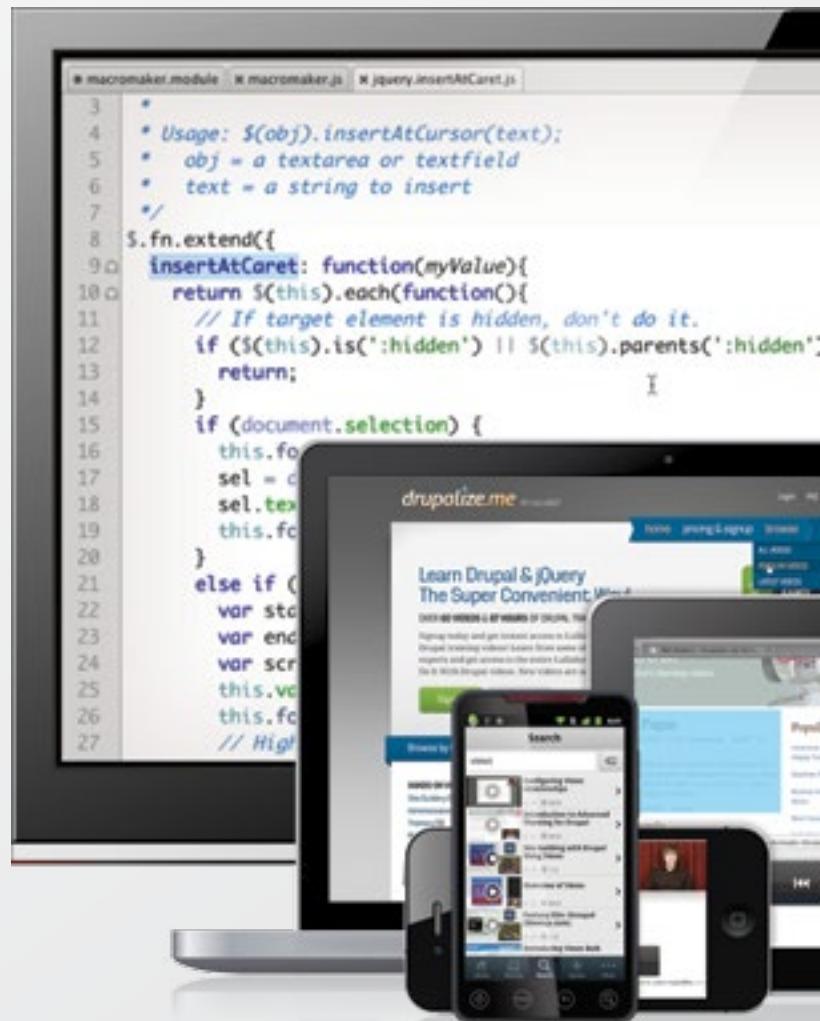
drupalize.me

The Most Convenient Way to Learn Drupal!

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

If you sign up before June 30, 2012 using the code LINUX10, you'll get $10 off any Drupalize.Me subscription.
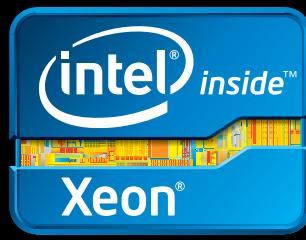
Go to http://drupalize.me and get Drupalized today!

CONTENTS
JUNE 2012
ISSUE 218

COOL PROJECTS

FEATURES

ON THE COVER

Cover Image
© Can Stock Photo Inc. / Garsya

COLUMNS

INDEPTH

REVIEW

IN EVERY ISSUE

40 3-D PRINTING

56 POLYWELL'S i2303

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

Unparalleled performance density

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-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

IXR-22x4IB

intel® inside™
Xeon®

iXsystems

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

SHAWN POWERS

Keeping It Cool...

Three days ago here in northern Michigan, we had a heavy frost overnight. Those gardeners who ambitiously planted their plants early had to cover them with tarps or tents to make sure they didn't die in the frigid night. Yesterday, the temperature was 96°F. Michigan weather is weird. This month is our Cool Projects issue, so even if the temperature continues to push 100°, I'll rely on the June issue of *Linux Journal* to keep things cool.

Reuven M. Lerner starts the issue off with a great way to make your sites look nice, even if you're not a designer. Twitter Bootstrap is a framework that has gained lots of following during the past year, and Reuven shows why. Speaking of "why", Dave Taylor continues down his dark path as he describes how to cheat at *Draw Something*. If you liked the fame and fortune Dave's *Scrabble* scripts provided, you'll likely feel right at home with his next round of rule augmentation. In all seriousness, Dave uses some pretty cool methods for teaching powerful scripting techniques. We won't judge him too harshly, because

it's all done in the name of education. Still, I'm never playing an on-line game against him!

Kyle Rankin has been chatting with me over IRC for months about his new toy. This month, you get to see the all details, as he shows off his fancy 3-D printer. Whether you try the model Kyle bought or decide a different style is more appropriate for your needs, the concepts are fairly similar. And as much as I resisted the idea of 3-D printing, I have to admit I sort of want to try it myself. At the very least, printing knick-knacks as gifts might be a way to help fund the purchase. Oh, and Kyle? My birthday is in July. I really could use a charging stand for my phone!

Inside a hot server room, nothing is quite as cool as Linux. In my Open-Source Classroom column this month, I talk about file serving. As far as the "cool factor" goes, it may not be the most exciting thing to set up, but it's a great way to introduce Linux into a traditionally proprietary environment. Following my column, I got to review a seriously awesome desktop computer

from Polywell. The i2303 looks like a nettop but behaves like a workstation-class machine. Find out if this little powerhouse ticks all your boxes as well.

Amit Saha introduces a really cool project this month, ownCloud. If you like the convenience of cloud computing, but worry about someone else controlling your data, why not make your own cloud? Supporting everything from file storage to music management, ownCloud allows you to access all your data from a browser. Rather than the other end of that browser connecting to a giant server farm, however, it can connect to your own servers, or at least servers you control.

Sometimes systems don't come with all-inclusive components like ownCloud. James Litton explores integrating disparate systems this month, even if they aren't designed to do so. In situations like these, it's often just as important to be creative as it is to program well! Rebecca "Ruji" Chapnik follows that creativity through with her article on video art. Although many people assume a Macintosh computer is the only way to survive in the world of video and animation, Rebecca proves that theory wrong. Whether you want to do screencasts, stop motion, animation or non-linear video editing, Linux has the tools you need. Rebecca even shows how to make animated GIFs!

If your idea of a cool project is more

along the lines of wall-mounted circuit boards and multi-monitor interfaces, you're in good company. Marcin Teodorczyk shows off his wall-mounted computer and explains how he tweaked it along with his laptop to create a Frankenstein-like computing environment that does everything he commands. Using multi-headed Linux and Synergy, he controls the various computers from a central keyboard and mouse. It's pretty cool stuff.

Some of you may like more subtle versions of "cool" for your projects. That's where something like Howard Powell's article on ZFS and Btrfs might tickle your fancy. Everyone's comfortable with RAID and LVM, but with filesystems like ZFS and Btrfs, much of the RAID overhead is gone. Although ZFS isn't new technology, it's fairly new to the Linux world. Howard talks about the concepts and nuances of these exciting filesystems.

Whether you're just starting to feel the heat of summer or if you've never left it, the Cool Projects issue is always a refreshing read. We hope you enjoy it; we sure enjoyed putting it together for you! ■

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

Congrats on the Digital Issues

I still don't get why people are whining about the digital vs. paper issue. I personally receive more than 90% of my information digitally, and I read a lot of books (and other stuff) on my BeBook Neo without problems—or without any annoying problems. Even on my 24" desktop, reading the PDF is okay. And maybe the most important thing is because I'm from Eastern Europe, the paper subscription was not possible for me (postal issues, delays and so on). Because of the digital edition, I'm the proud owner of a subscription now.

—Florin

Awesome! We're always happy to hear success stories. Thanks for sharing.—Ed.

Why Not Slackware?

I've been a subscriber for at least about ten years, perhaps longer, since *Embedded Linux Journal* folded. This is one magazine I read thoroughly, because it agrees with what I use a lot of, Linux.

I also use the Slackware distribution thoroughly. So the fact remains, why not cover that one? I admit, it's not as easy to use as Ubuntu, but given the fact that Slackware wants the user to do everything that the user wants to do, it just works. There are even third-party repositories. So ideally, the magazine should start being more distribution-agnostic, and if a specific distro is needed, use that one (Slackware).

—Gregg C. Levine

You make a valid point, but it's rough for us to force distro-agnosticism on our authors, because they're writing from their experiences. Because Ubuntu and its variants are the most commonly used, it's what we see more often than not. We will continue to strive for non-distro-specific solutions, but we'd rather have accurate articles geared toward a specific distro than commit errors in an attempt to stay "politically correct".—Ed.

Downloading *LJ* via wget

In the April 2012 issue of *LJ*, Bob Johnson, in the Letters section, states that he's having a hard time downloading *LJ* issues via `wget`, as he would rather read the PDF file in an independent reader rather than in a browser plugin. I perfectly agree with him, and I have no problem downloading the PDF via `wget`.

Here is how I do it:

- I click on the link received via e-mail.

- In the browser that opens, I copy the entire address bar (via Ctrl-A, Ctrl-C). The URL is something like "http://download.linuxjournal.com/pdf/get-doc.php?code=...".

- I then type `wget` and paste the URL, with one very important precaution though. The URL must be enclosed in double quotes, as the parameters are separated by an ampersand (&), which has a special meaning under UNIX/Linux.

It works like a charm! So, if you have the chance to pass that information on to Bob, that would be great!
—**Bruno**

Thanks Bruno! I suspect a procmail recipe and a little sed/awk magic could make the process completely automatic.—Ed.

Too Esoteric?
As a former print subscriber, I have been getting some free views of the digital version. The problem I have has nothing to do with print vs. digital, but rather the content. As an individual managing several Linux computers, I am by definition a system administrator, but on a small scale.

When I first subscribed to *LJ*, each issue had at least one, and usually a few articles that I could get useful information from. But as time went on, things seemed to get more esoteric, more advanced than what I could actually use, so when I realized that a half-year's issues had absolutely nothing of interest, I quit *LJ*.

I still can relate to Doc Searls' pearls, but not anything else. I write this having just viewed the System Administration issue, but the same has been true for the last several issues. I think you're out of touch with my segment of Linux users.
—**Greg**

Wow Greg, we're really sorry to hear that. Our goal is to be relevant for as much of our readership as possible. We recently started The Open-Source Classroom column, hoping to cover an area we thought might be lacking. At the very least, hopefully you'll still garner some good information from our Web site. Thanks for letting us know. Without feedback, we have no hope of meeting our readers' needs.—Ed.

LJ Download with wget Possible Solution
I was reading the April 2012 issue, and regarding the letter from Bob Johnson titled "*Linux Journal* Download with wget?", I think I may have a solution: the cliget extension for Firefox (**https://addons.mozilla.org/en-us/firefox/addon/cliget**).
—**Jordi Clariana**

Thanks Jordi. I suspect we'll hear back from Bob. Hopefully, one of these solutions works for him. Thanks again for the suggestion!—Ed.

LJ Android App

After checking many readers on the market for my Android phone (Kindle, Aldiko, PDF-based and so on), I have to admit that the best interface still is yours (Texterity's). Its reflowed text is very good; the graphics look great, and the Navigator (index) is a great way to move between articles. Would it be too much to ask that you try to improve this application further? As great as the Navigator is, if I switch to another task, say to check a phone number, I have to go find the article I was reading again, and the location I was at. Make it remember my last location.

If I do not have connectivity (such as on an airplane), I cannot access the content, as there is some check for connectivity when the application starts. Of course, it was made by developers who do not know what it is not to have an always-on data link, but that is still the case. My data is cached; that's not the issue (I can't even access the library). Make it stop "calling home" when the app is launched for the first time.

Finally, moving between articles in reflowed mode is too easy. What do I mean? The slightest scroll along the x-axis sends me to the next article. The fact that I'm reading on a phone probably

exacerbates that (smaller screen), but it still takes 1/4" or less on a 4.3" screen.

That is it. The native app works great; now please fork some love to Texterity so those little annoyances can be fixed (so I can comply with the FAA and actually turn off my phone).
—**Pablo**

We'll be sure to pass your note to the Texterity folks. I think the "calling home" that happens is Texterity checking for new content, but I'll admit, I'm not sure that's all it's doing. Thanks for the feedback!—Ed.

Yet More Digital Edition Whining
You guys do seem diligent and serious about "going digital right", providing alternative formats and access techniques for maximum convenience. And, I suppose that you are getting a bit sick of digital edition whining. I trust that reader loyalty has kept this change from hurting your circulation numbers too much. There are two aspects of this that, taken together, bring up a question in my mind: you are quite willing to go to miles of extra trouble to get all alternative formats right, and the reason for going digital is economic. So why not see paper is an alternative format—one that, because of raw economic reality, commands a premium price? Some of your readers (not me!) are quite wealthy (some even have been made that way by application of skills promulgated in *LJ*). There are any number of boutique
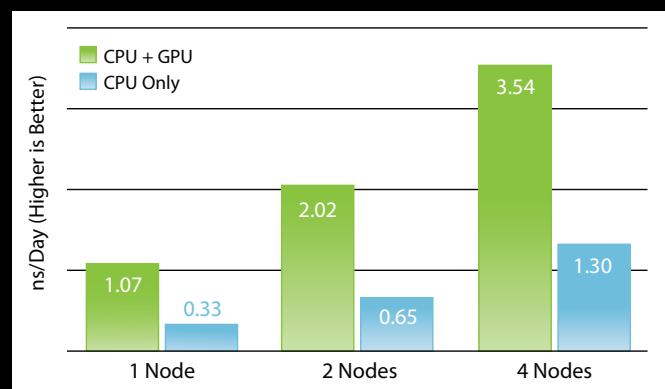
publish-on-demand technologies that could turn PDF into paper, for a (probably large) extra cost.

Whatever that cost, some population of those you are losing could choose and would choose to pay it. Even if the cost was so prohibitive that practically no one would choose it, at least the choice would be preserved, and people could feel good about saving themselves money by moving on to digital—rather than the money being saved all going to your corporation. Get your bean counters out of their cages and cost out a "boutique paper" alternative that pays its toll and offer it to your readership! You are doing so much else "right" with going digital, I hate to see you miss this trick.
—**Dave Carew**

Thanks Dave, we really have been working hard on the digital versions. It's hard to remember the days when we only had one layout to work on! As far as the print-on-demand aspect goes, we haven't had much discussion internally about it. I always assumed folks would just send their PDF files to the boutique printers if they wanted paper. It occurs to me, however, that I might be putting my foot squarely in my mouth. Perhaps a commercially produced PDF like Linux Journal *would be rejected as a copyright violation or something. We'll have to look into it.—Ed.*

Digital Subscription Side Effect
I would think that the geeks all would have loved the digital subscription. I guess not. I enjoyed reading *LJ* before bedtime. Now I do it on a tablet. A funny thing happened—a side effect. I started questioning myself for keeping old magazines around when I can always buy a DVD of the archives if I really need to. I had only a few *LJ*s, because I bought them from the bookstore and have kept a few hard copies. Now I subscribe to the *LJ* digital version. This spring cleaning has prompted me to throw out a hundred pounds of archived magazines with topics on radios and electronics and all those IT magazines that presented peripheral articles with no meat. I even had my old NT magazines that I gladly discarded from the 1990s. Thanks *LJ* for a great product. Keep developing, keep improving and keep your eye on the future. See you at the next SCALE.
—**Roman Kamienski**

That's awesome, Roman. I still get a few paper magazines at work, and I find myself getting annoyed when I misplace them. Linux Journal *is always in my pocket now, and on my computer only a click away. I think digital magazines are making me even more forgetful!—Ed.*

E-Reader Downloads
I'm sure you're tired of the messages regarding the whole e-reader topic. At first I was dismayed because I love the portability of print, but now I am squarely in the e-reader camp and all of its benefits. Kudos on the digital version. My one gripe, however, is the mechanism by which I can download

the EPUB files for my NOOK. The default browser in the NOOK doesn't seem to want to download the files. Is there a way this can be fixed? I keep forgetting to download the issues and sideload them via the laptop. Thanks, and keep up the great work!
—**Dave Heebner**

I don't have a NOOK myself with which to experiment, but perhaps if our group effort to download automatically with wget pans out, an additional step to e-mail the downloaded file as a regular attachment would work.

I'm beginning to picture in my mind a script that scans an IMAP server on the first of every month, and based on configuration options, will save and re-mail the digital versions to the reader as plain attachments.—Ed.

How to Introduce My Son's School to Linux?

I've often heard Richard Stallman argue about why schools should use free software instead of proprietary software. My kids are very young, and they haven't been introduced to computers yet, but I would like their school to start considering GNU/Linux as an alternative to Windows, but I don't know how to do that. I would really appreciate any information to achieve this goal. Maybe you could write an article about this or point me to a previous article that you already have?
—**Carlos Le Mare**

Your timing is perfect! My Open-Source Classroom column this month touches on one way to start introducing Linux in a server room. If you haven't won the hearts of the IT staff, however, it might be rough. If you read the past few months of my column regarding LTSP, you might think about a small classroom-sized install. You'd still need permission from the Principal and IT staff, but if you find the right teacher, you could start a pretty powerful grass-roots campaign. Good luck!—Ed.

Child-Friendly Linux?

My six-year-old has a computer with Ubuntu installed (thanks to dad, of course), and now I am wondering how I can secure the computer. She goes on YouTube and mostly stays on secure channels, but sometimes some not-very-friendly channels come up. Could you describe some ways to make the Web experience for kids better? I have tried the 0.0.0.0 in the hosts file for YouTube, but that stops my kid from even going there, and as I mentioned before, there are some channels that are meant for kids.
—**Trond Huso**

Since YouTube is completely controlled by Google, filtering specific aspects of the site is challenging. You might check out YouTube's education-specific site though: **http://www.youtube.com/ education***. If that doesn't help, there are sites like* **http://www.schooltube.com***, but they don't have the same amount of content.—Ed.*

Is Synapse Launcher Dead?

I started using Ubuntu and Synapse a little while ago. I learned the Synapse program, and I was very happy with it, and I wanted to give some ideas to the Synapse team. I sent my idea to the launchpad page, but no one replied. I found one of the main developers on his personal blog, and he answered me, "I'm really sorry, but we are both busy with our jobs, so at the moment we are no longer active on Synapse." You can see it here: **https://answers.launchpad.net/synapse-project/+question/192032**.

I'm very sad that the project is practically dead—no bug fixes and no updates. Can you let others know? Maybe some other developers would like to continue the project. I know Synapse has many users, and if people know this, they might want to join the project. Thank you.
—Emre

Sadly, this happens to many, many projects. Often wonderful applications are developed by overworked folks in their spare time. Once that spare time goes away, so do updates. If there is a large user base, you might consider posting to the e-mail list or support forums and hope someone else will take over. I wish I had better news.—Ed.

Correction

In my article "System Administration of the Watson Supercomputer" in the April 2012 issue, I made an error regarding the corpus size. I mistakenly had it as 400TB, but the corpus size is actually 59GB.
—Aleksey Tsalolikhin

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

Save Money

Want a Linux server, powered by the latest Intel Xeon E5-2600 series processors, for $12.95? Try a virtual private server.

Gone are the days when everyone needed their own Xeon processors. Intel's new E5-2600 family of Xeon processors boost performance up to 80% (over previous generation Xeon processors) -- making virtualization more attractive than ever before.

Now is the time to benefit from Intel's performance breakthrough and save money at the same time. That is the brilliance of virtualization, and that is why intelligent system administrators are choosing virtualization today.

Whatever the workload, you will appreciate faster execution with Intel's new E5-2600 series of Xeon processors. And best of all, because of virtualization, the full 80% performance upgrade is available at a fraction of Intel's price.

Slash your operating costs and send your Linux applications soaring with Acenet's new *performance series* virtual machines, powered by Intel's E5-2600 processor lineup.

Choose from three virtual machines, each outfitted with a different caliber of Xeon processors, to get the right amount of processing power for your task.

All three virtual machines are equiped with high-performance 15K SAS hard drives in a hardware RAID-10 array, protected by a flash backup unit.

Try one risk-free for 30 days.

Xen E5-2640 Virtual Machine

4 vCPU (2.5 GHz @ 7.2 GT/s)
    2.5 GHz Max Turbo Frequency
2 GB ECC RAM (1333 MHz)
18 GB 15K SAS HDD
200 GB Free Bandwidth
$12.95 Monthly

Xen E5-2670 Virtual Machine

4 vCPU (2.6 GHz @ 8.0 GT/s)
    3.3 GHz Max Turbo Frequency
4 GB ECC RAM (1600 MHz)
36 GB 15K SAS HDD
400 GB Free Bandwidth
$29.95 Monthly

Xen E5-2690 Virtual Machine

4 vCPU (2.9 GHz @ 8.0 GT/s)
    3.8 GHz Max Turbo Frequency
8 GB ECC RAM (1600 MHz)
73 GB 15K SAS HDD
800 GB Free Bandwidth
$69.95 Monthly

VPS Control Panel

Plus each virtual machine includes access to the VPS Control Panel, where you can start, stop and reboot your virtual machine, reset its root password, change its hostname and even reload the operating system.

CLICK HERE to go to Acenet's website for complete details...

diff -u
WHAT'S NEW IN KERNEL DEVELOPMENT

The **KVM API** has been accruing cruft for many years, a result of the natural process of being an actively developed project and having a lot of crazy deep requirements for virtualizing systems. It's hard to just up and redo an API that's already well established though, so **Avi Kivity** suggested either a thought experiment of how it might be done or, at most, a co-API that would sit side by side with the original one for several years.

Interesting enough, one of the things people favored in the original crufty interface was the ease of producing cruft. Specifically, Avi wanted to migrate away from **ioctls** as the primary interface and replace them with system calls. This would be cleaner than the crawling chaos of ioctls, but folks like **Alexander Graf** and others argued that KVM still was under hot development and definitely was going to need an unknown vector of extensions, and ioctls made that kind of development very easy.

So, this is more of a negative news story: hot flash—KVM interface to remain unchanged! But still, it's interesting because Avi seems to be right. The KVM API has a lot of unnecessary craziness, such as the way the virtualized CPU state is handled. The current cluster-bomb of ioctls in that area could be replaced by just two system calls that would operate on a single orderly data structure. And, there are other similar areas he points out. But apparently, the pace of development will have to slow down, and the interfaces stabilize, before those cleanups really will be practical.

Rakib Mullick is working on a new approach to load balancing between multiple CPUs on an **SMP** system. The **Barbershop Load Distribution** (BLD) algorithm keeps track of any change to the load on each CPU and always assigns new processes, and processes coming in from swap-space, to the least-loaded CPU.

Because the preferred CPU is known instantly at the time a process needs to be assigned, the BLD has a speed of O(1), which means it takes the same amount of time to identify

the appropriate CPU, regardless of whether there are 2 or 2,000,000 CPUs from which to choose.

As Rakib explains it, this is load balancing without load balancing, because the actual effort to balance the load never takes place. The appropriate CPU simply is picked from an ordered list of all CPUs that already has been sorted by total load.

His code is still under development and hasn't been given a thorough going-over by many developers. It could turn out to be an obvious improvement that is accepted quickly, or it could fall victim to the fact that process migration tends to be a deep, dark art with many incantations and strange requirements. Time will tell. But, it seems like Rakib's got a fun piece of code that inevitably will gain some interest among CPU people.
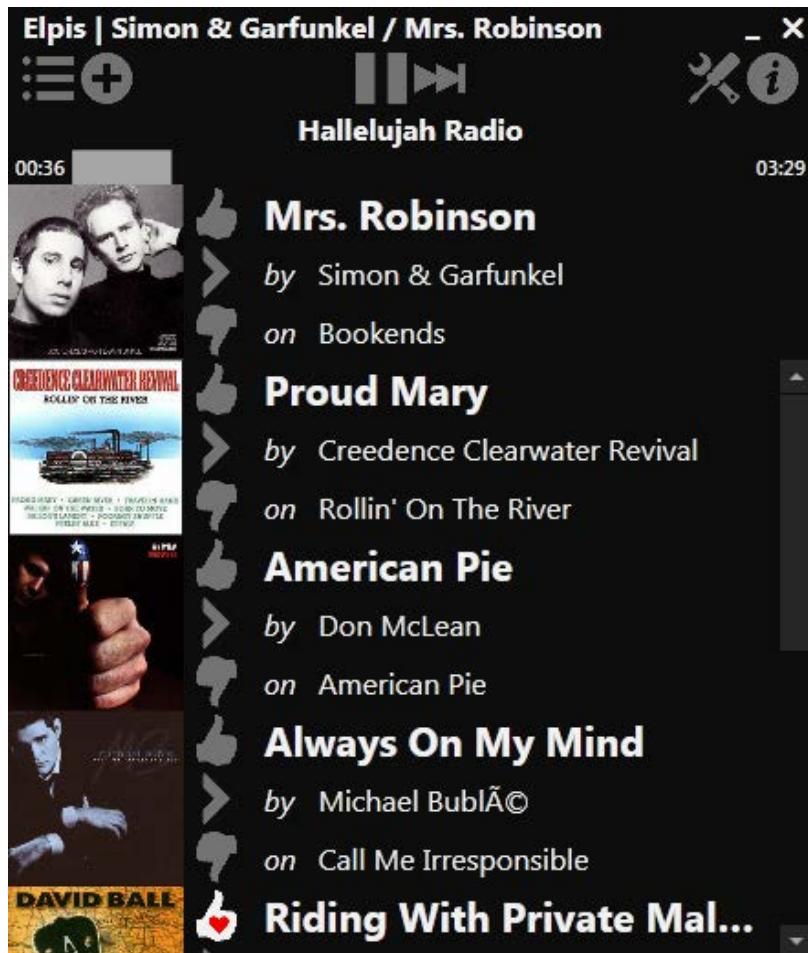
Periodically, someone expresses frustration at how difficult it is to test scheduler improvements. This time it was **Pantelis Antoniou**, who has been working on **big.LITTLE** architectures. The idea behind big.LITTLE is to have a device with two CPUs: one smaller and weaker that handles ordinary operations, and the other bigger and more powerful that gets processes

requiring more kick. Any process on the smaller CPU that starts to get too active will be migrated over to the bigger CPU in a seamless fashion. Devices like this have been coming out lately because they tend to have longer battery life than traditional systems, but they're radically different from the SMP-oriented scheduling that Linux grew up with.

Pantelis has suggested implementing some kind of user session recording and playback, so that a large corpus of typical user activity could be built up and then used to test improvements to Linux's scheduler code. As it turns out, some rudimentary session recording tools already exist, but they'd need a lot of work in order to be useful for scheduler testing. And, one potentially big obstacle is the fact that any type of recording undoubtedly would slow down the user interface, changing user behavior and tainting the sample.

But, it still seems like a cool idea. Scheduler advancement is a nightmare of uncertainty in which debates often take the form of the following: "We should do this!" "Why???" "Because I say so!" "Well, I say something else!" "Oh yeah?", and so on.—ZACK BROWN

Non-Linux FOSS

Usually it's Linux folks who are left out in the cold with regard to cool software. In the case of Pandora Radio, however, it's tough to beat the Linux native Pithos application. Thankfully, Windows users have an open-source option for native Pandora Radio playback as well. Elpis is a .NET Pandora application available for Microsoft Windows.

We still think Pithos looks nicer and has a smoother interface, but if you're stuck on Windows, Elpis is fully open source and under active development. A screenshot is worth a thousand words, but if you want to check it out for yourself, visit **http://www.adamhaile.net/ projects/elpis**.—SHAWN POWERS

Plex Media Server + Roku = Awesome

Plex always has been the Mac-friendly offshoot of XBMC. I've never considered using an Apple product for my home media center, so I've never really put much thought into it. Things have changed recently, however, and now the folks behind Plex have given the Linux community an awesome media server.

Installing the media server is fairly straightforward. Instructions are available at **http://www.plexapp.com**. The server application runs on a headless Linux server and is configured via a Web interface. After you've pointed Plex Media Server at your video collection, the real magic begins.

Fire up your Roku, any model, and search for the Plex channel in the Roku Channel Store. With some simple configuration, your Roku will be able to browse your entire media collection and stream HD video to your television. The responsiveness is incredible, and the video quality is astounding. I was expecting pixelated video with stuttering playback over wireless, but everything was smooth. Your local media behaves just like Netflix! For more information, check out the Plex Web site: **http://www.plexapp.com**.

—SHAWN POWERS

Open Source for the Space Age

NASA has started a rather ambitious project: to provide open-source everything. The main site is located at **http://open.nasa.gov**. From here, there is access to data, code and applications, among other things. This is a great launching point for anyone interested in space science and NASA work. In this article, I look at what kind of code is being made available that you might want to explore.

The available software covers several genres. Some are low-level, systems-layer software. You can go ahead and do some really long-distance transfers with the Interplanetary Overlay Network (ION). This is an implementation of the Delay-Tolerant Networking architecture (DTN) as described in RFC 4838. This software is physically hosted at SourceForge, and you can use this code to communicate with your next interplanetary probe.

A bit more down to earth is a middleware package that actually is hosted by the Apache Foundation. You can download and use the Object-Oriented Data Technology (OODT) middleware. OODT is component-based, so you can pick and choose which parts you want to use. There are components to handle transparent access to distributed resources, data discovery and query optimization, and distributed processing. There are also components to handle work-flow and resource management. Groups that are using it include the Children's Hospital of Los Angeles and NASA's Planetary Data System. If you're managing data systems, this might be worth taking a look at.

Getting back to actual science processing, you might want to download the Data Productivity Toolkit (DPT). This package is a collection of command-line tools, written in Python, that lets you work on text data files. These utilities follow the UNIX design method of having small utilities that do one task well, and then chaining them together to do more complicated processing. There are tools for massaging and manipulating your data, tools for doing statistics on that data and even tools for visualizing the data and the results. Many of the tools even provide an API to basic Python and numpy/scipy/matplotlib routines.

And, while I'm talking about Python and science, you also can look at SunPy. SunPy aims to provide a library of routines that are useful in studying solar physics. With it, you can query the Virtual Solar Observatory (VSO) and grab data that you can process. Many routines are available that allow you to plot this data using various color maps and processing filters. There is a Sun object that contains physical constants useful in solar physics, along with the sun's position

and numerous other solar attributes.

A lot of the computational work done at NASA involves clusters of machines and massively parallel code. This means the NASA folks have needed to put together lots of tools to manage these machines. They also have been nice enough to release a lot of this code for public consumption. The first of these is multil (Multi-Threaded Multi-Node Utilities). In the standard GNU file tools, cp and md5sum operate as a single-threaded process on a single machine. The multil tools provide drop-in replacements called mcp and msum. These utilities use multithreading to make sure each node is kept as busy as possible. Read and write parallelism allows for individual operations of a single copy to be interleaved through asynchronous I/O. Split file processing allows for different threads to operate on different portions of a file in parallel.

NASA also provides a utility to give SSH access to your cluster. There is a middleware utility called mesh (Middleware Using Existing SSH Hosts) that provides single sign-on capability. Mesh sits on top of SSH, and instead of using the local authorized_keys file, loads a file for a dedicated server at runtime. Mesh also has its own shell (called mash) that restricts what applications are available to the user. Using this system, you can add and remove SSH hosts that are available to be used dynamically. Also, because the authentication is

handled by a library that is preloaded when SSH first starts up, the restrictions are sure to be enforced on the user.

Now that you have a connection mechanism, you may need to handle load balancing across all of these machines. Again, NASA comes to your aid. It has a software package called ballast (Balancing Load Across Systems) that might help. This package handles load balancing for SSH connections specifically. Each available host runs a ballast client, and there are one or more ballast servers. The servers maintain system load information gathered from the clients and use it to make decisions about where to send SSH connection requests. Because all of this is handled over SSH, the policy deciding which host to connect to also can take into account the user name. This way, you can have policies that are specific to each user. This lets you better tune the best options for each user, rather than trying to find a common policy that everyone is forced to use.

Going back to doing science, another important task is visualization, and NASA has released several tools to help. The first one I look at here is World Wind. This is an Earth visualization system. You can use it to get a 3-D look at the Earth and to see data projected onto the globe. It is a Java application, so it works on any desktop that has a Java virtual machine, as well as in most browsers. It is a full development kit, and it has several

example applications that you can use as jumping-off points for your own code.

Taking visualization further from the surface of Earth, there is ViSBARD (Visual System for Browsing, Analysis and Retrieval of Data). This application allows you to pull data from multiple satellites and display them concurrently. It also allows for 3-D viewing of all of this data. This type of vector field information is very difficult to analyze in 2-D plots, hence the need for this kind of tool. The latest version also allows you to visualize MHD (Magneto-Hydro-Dynamic) models. This way, you can compare results from model calculations to actual satellite measurements.

More extensive image processing can be done with the Vision Workbench. This is an application and a full library of imaging and computer vision algorithms. It isn't meant to be a complete, cutting-edge library though. Rather, it provides solid implementations of standard algorithms you can use as starting points in developing your own algorithms.

When you're ready to go and launch your own satellite, you can download and use the Core Flight Executive (cFE). This software is used as the basis for flight data systems and instrumentation. It is written in C and based on OSAL (Operating System Abstraction Layer). It has an executive, along with time and event services. You can track your satellite with the ODTBX (Orbit Determination Toolbox). The ODTBX
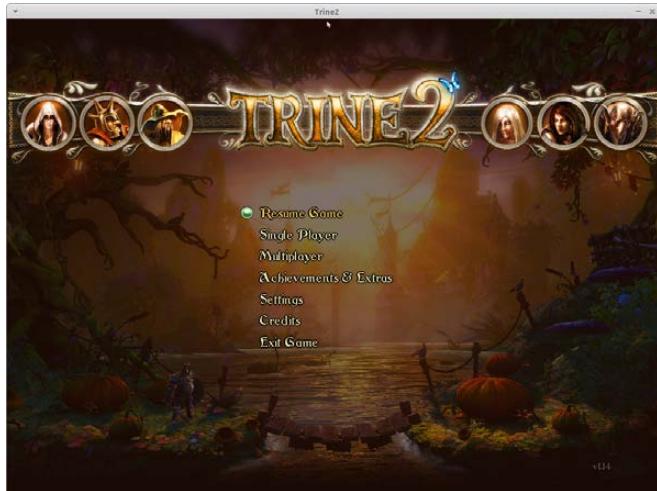
package handles orbit determination analysis and early mission analysis. It's available as both MATLAB code and Java.

The last piece of code I cover here is S4PM (Simple, Scalable, Script-based Science Processor for Measurements). This actually is used at the Goddard Earth Sciences Data and Information Services Center to do data processing. It is built up out of a processing engine, a toolkit and a graphical monitor. S4PM allows a single person to manage hundreds of jobs simultaneously. It also is designed to be relatively easy to set up new processing strings.

The open-source project at NASA doesn't cover only code. NASA has been releasing data as well. The Kepler Project is looking for exo-planets. As I mentioned previously, you can download data from the Solar Dynamics Observatory. You can work on climate data by checking out information from the Tropical Rainfall Measuring Mission. You can look up tons of data from the various moon missions, from Apollo on up. There also is data from the various planetary missions. Climate data and measurements of Earth are available too.

I've touched on only a few of the items NASA has provided for the public. Hopefully, you have seen enough to go and check out the rest in more detail. There is a lot of science that regular citizens can do, and NASA is doing its part to try to put the tools into your hands.—**JOEY BERNARD**

Trine 2 from Frozenbyte

One of the great things about independent game companies is that they realize Linux gamers exist—and we're willing to spend money. Frozenbyte is the indie game developer that brought us *Shadowgrounds*, *Shadowgrounds Survivor* and *Trine*. Frozenbyte was kind enough to send me a review copy of its newest release, *Trine 2*.

As an obvious sequel to the original, *Trine 2* is a platform-based side-scroller. It has stunning graphics, rich sounds and a pleasant storyline. The game revolves around solving physics puzzles in a fantasy-based world. It's no secret I'm a rather poor gamer, but *Trine 2* is enjoyable even for a part-time player like myself. It also supports multiplayer co-op mode for up to three players. Multiplayer mode works both on the local network and over the Internet.

I'll admit, it took a little while to get used to jumping with the W key (*Trine 2* uses the WASD keys for movement), but that wasn't a big deal. I was able to jump right in and start playing. I truly appreciated the ability to skip past cut scenes and get right to the action. If you're in the mood for a fun game with great graphics, I highly recommend *Trine 2*.

The Collector's Edition comes with a digital artbook and soundtrack. You can buy it through the Humble Store now at **http://www.trine2.com** or from the developer's Web site, **http://www.frozenbyte.com**.

—SHAWN POWERS

New White Paper and Resource Library

Visit this newest section of LinuxJournal.com at **http://www.linuxjournal.com/whitepapers** to find carefully selected resources from *Linux Journal* partners that will help you with topics from version control to automation, multimedia and more. Is there a topic you'd like to see covered? Please let us know by sending e-mail to webmistress@linuxjournal.com, and we'll do our best to accommodate.
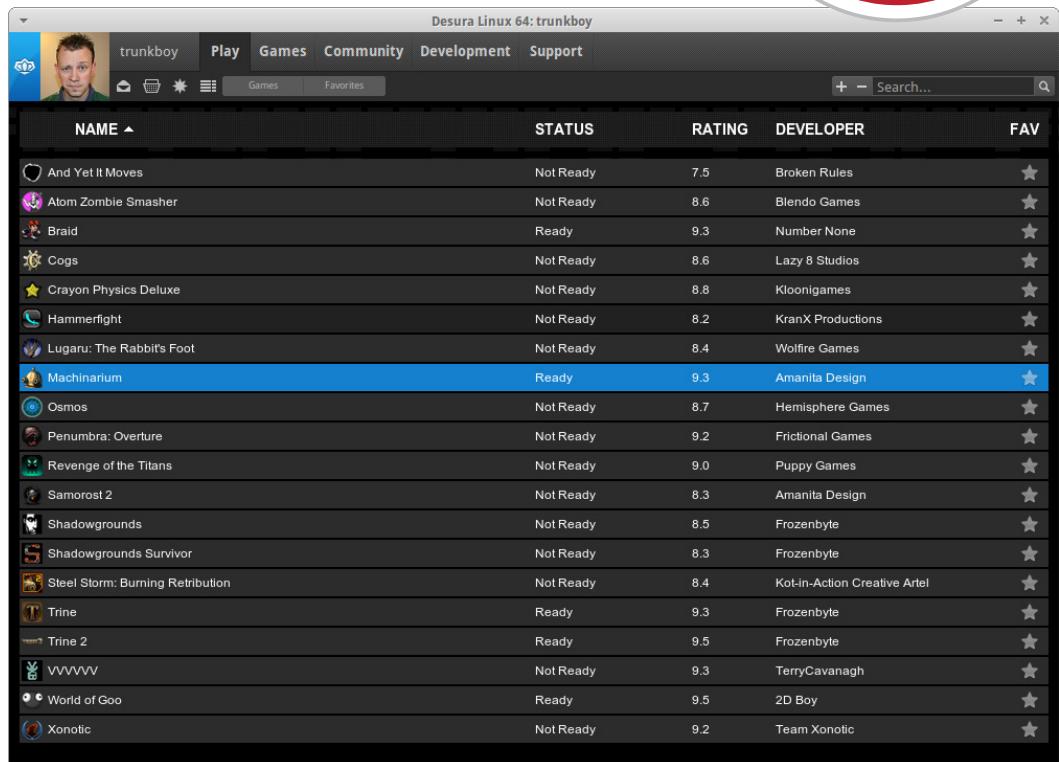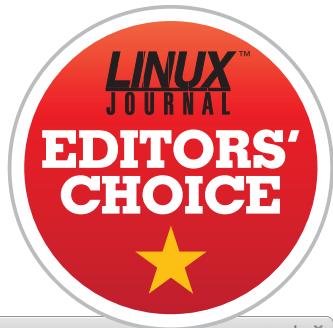
—KATHERINE DRUCKMAN

Getting Steamy with Desura

I remember the first time I tried to install *Quake* on Linux. I was so excited to have a native "real" game to play, that I couldn't grab my installation CD fast enough. Unfortunately, I didn't really take good care of my media, and the CD was too scratched to read.

I suspect something similar happened to the inventor of Steam for Windows. Having a permanent on-line archive of your video game library is awesome. Unfortunately, Linux users not only don't have Steam, but it seems like every game we do have installs in its own unique way. Some have binary installers; some are in the package management system; some must be compiled.

Thanks to the fine folks at Desura, Linux users finally get to play with the cool kids! Although obviously a clone of the Steam idea, Desura does an incredible job of simplifying management and installation. It has a built-in game store offering free and for-pay games, and it even supports "codes" for activating games purchased elsewhere. (My favorite "elsewhere" is the awesome Humble Bundle site that periodically sells Linux-compatible games.)

If you're a gamer and a Linux user, you owe it to yourself to give Desura a try. You can download it at **http://www.desura.com**. And if you like Linux-compatible games, be sure to check out **http://www.humblebundle.com** from time to time.—**SHAWN POWERS**

REUVEN M. LERNER

Twitter Bootstrap

Even if you're not a designer, Bootstrap is a great way to make your sites look nice.

In my last article, I described the relatively new phenomenon of design frameworks. Just as Web application frameworks make it easier to create server-side applications, design frameworks make it easier to create client-side designs. They do this by providing predefined CSS classes, each of which indicates the width of the column you want to create, the type of element you're using, or the color and style you want to use.

It's true that CSS masters will find a design framework unnecessary. After all, such people already can tweak the layouts, adjust the colors or create beautiful buttons, all on their own, without having to use the assistance of predefined classes. But speaking as someone who is design-challenged, the introduction of design frameworks has made it possible, even for someone like me, to make a Web application that doesn't cause people to go screaming into the night.

The secret behind such frameworks is that by buying into their predefined CSS classes, you give up some of the freedom you had with pure CSS. You agree to use their classes and to use their HTML structure in some cases. This is generally a worthwhile trade-off, in that your code and CSS end up being much shorter and more legible. You can concentrate on your domain of expertise, namely software development, rather than tweaking the CSS to look just right. And, because these frameworks constantly are evolving to support designers and developers, each upgrade supports more browsers, more optimizations and more CSS classes that you can use to integrate into your work.

One framework in particular has taken the world of Web development by storm during the past year: Twitter Bootstrap. Bootstrap first was released in summer 2011, and it was written by Mark Otto and Jacob Thornton, both

of whom work at Twitter. Bootstrap, which was released under an open-source license on GitHub, has become an almost-overnight sensation. Indeed, it is currently the most-watched open-source project on GitHub, surpassing even Ruby on Rails.

There have been grumblings on such sites as Y Combinator's "Hacker News" about the proliferation of sites using Twitter Bootstrap. I see this as positive, raising the minimum acceptable level for Web design, while also making it straightforward and easy for designers to extend those designs.

Bootstrap is based on LESS, a language that compiles into CSS, and which is similar in many ways to SCSS. This means if you want to work with Bootstrap, you either need to compile the LESS stylesheets into CSS manually, take care of this in your deployment system or use a precompiled version of Bootstrap. Although using a precompiled version of Bootstrap is clearly the easiest way to go, it also gives you the least flexibility, because you won't be able to modify the LESS source to your own designs.

In this article, I discuss Bootstrap, including how you can integrate it into your own Web applications.

Installing Bootstrap

There are several ways to install Twitter Bootstrap. The easiest and most straightforward is to clone the Git repository from GitHub:

```
git clone https://github.com/twitter/bootstrap.git
```

This creates a new subdirectory named bootstrap, containing a number of files and directories. The docs directory contains the current documentation, the same as you can see on the Bootstrap site itself on GitHub. And, because it is written using Bootstrap, it provides a nice set of examples for using Bootstrap on your own sites.

I discuss the img directory below, so for now, you largely can ignore it. The js directory contains jQuery plugins that integrate with Bootstrap. The less directory contains, as you might expect, a number of files in LESS format, which then can be compiled into CSS files. It is these resulting CSS files that you will use on your site, not the LESS source files.

Now, how do you compile the LESS files into CSS? One way is to install and run the LESS compiler yourself. LESS is implemented in JavaScript, which means that it either can run in the browser or—in the case I'm describing—on the server, in a browserless implementation of JavaScript, such as Rhino or node.js. (It's true that earlier versions of LESS were implemented in Ruby, but that has not been true for some time, so be sure to use the JavaScript version.) I decided to upgrade node.js on my server by

downloading the source and installing it from scratch with this:

```
./configure && make && make check && make install
```

Once this was done, I had not only the "node" binary, but also the "npm" binary, which allows me to install packages for node.js. You can install LESS with the following:

```
npm install less
```

Once that's done, you should be able to go into the less directory and issue the command:

```
lessc bootstrap.less > bootstrap.css
```

This compiles bootstrap.less into CSS, which you capture into the file bootstrap.css.

Some Web application frameworks can perform this compilation for you behind the scenes, on a regular basis or as you deploy. It's worth searching through the documentation for your favorite application framework. Bootstrap has become so popular that nearly every major one supports it somehow.

Now, if this seems like a lot of work just to get started with LESS, you're right. And in general, you probably won't want to start off adjusting the source LESS files, but rather just playing with the resulting CSS classes and other goodies Bootstrap provides. To do that,

you simply can take a snapshot of the compiled CSS files created by LESS. One quick-and-dirty way to do this is to grab the CSS file that came with the Bootstrap documentation; although you won't be able to change things in the LESS source, it's good enough for beginning purposes. Just look in docs/assets/css/bootstrap.css, and copy that into your Web project.

A middle ground, which allows you to customize things easily but doesn't require that you install and use the LESS compiler each time, is to use the Bootstrap configure-and-download functionality, from the Bootstrap home page. Enter the colors and basic design paradigms you want to use, and you'll get a custom CSS file delivered to your browser.

Finally, it's possible to run LESS in the browser, because (as I indicated above) it's written in JavaScript. However, I've generally found this to be a less acceptable way to go about things, if only because it's so much slower, and of course, I'll always want to use it on the server, so I generally get things going there instead.

Using Bootstrap
In every case except for the in-browser LESS compiler, you always will end up with a CSS file, typically named bootstrap.css. This file should be placed, not surprisingly, alongside your other CSS files, and then incorporated into your HTML file in

the standard way:

```
<link href="/bootstrap.css" media="all"
 ➥rel="stylesheet" type="text/css"/>
```

If you fire up your HTML file with just that, you'll see...well, almost nothing. A little bit has changed, namely the default fonts and margins (which probably have disappeared). But for the most part, you really haven't changed anything, and there aren't any obvious benefits. That's because Bootstrap is an à la carte system, in which you can take any or all of it, depending on your interests and needs.

That said, one thing that you'll almost certainly want to do is add a div tag with the "container" class immediately within your body tag, as follows:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>ATF</title>
    <link href="bootstrap.css" media="all" rel="stylesheet"
     ➥type="text/css" />
  </head>
  <body>
   <div class="container">
   <h1>Title</h1>
   <p>Body</p>
   <p>Body 2
     <ul>
       <li>Thing One</li>
       <li>Thing Two</li>
       <li>Thing Three</li>
```

```
     </ul>
   </p>
   </div>
  </body>
</html>
```

If your system is like mine, you'll see that the contents have shifted over to the right, with a large left-hand margin. Actually, the size of the margin depends on the width of your browser. By using a div within the body, the width of that div can stay relatively stable, even as the browser width changes around it.

Now, things truly become interesting when you start to divide your file into separate rows. You see, Twitter Bootstrap provides you with a grid—meaning that you divide your text into rows, and that each row consists of content, each of which consumes a certain number of columns. Bootstrap uses a 12-column grid, meaning that within any row, you can mix and match text using any combination of 12 columns. You indicate how many columns wide something should be by assigning a class of "spanN" to your HTML element, where N is a number between 1 and 12, inclusive.

For example, you can have a wide paragraph as follows:

```
<div class="row">
    <p class="span12">Wide paragraph. Very wide paragraph.
    Super-duper wide paragraph. Fill the screen's width paragraph.
    Use lots of text to fill the screen's width paragraph.</p>
</div>
```

Or, you can have four narrow columns:

```
<div class="row">
    <p class="span3">Narrow paragraph. Narrow paragraph. Narrow
    paragraph. Narrow paragraph.</p>
    <p class="span3">Narrow paragraph. Narrow paragraph. Narrow
    paragraph. Narrow paragraph.</p>
    <p class="span3">Narrow paragraph. Narrow paragraph. Narrow
    paragraph. Narrow paragraph.</p>
    <p class="span3">Narrow paragraph. Narrow paragraph. Narrow
    paragraph. Narrow paragraph.</p>
</div>
```

Or, even a left-hand sidebar, with main text on the right:

```
<div class="row">
  <p class="span2">Sidebar. Sidebar. Sidebar. Sidebar.
  Sidebar. Sidebar. Sidebar. Sidebar. Sidebar. Sidebar.
  Sidebar. Sidebar. Sidebar. Sidebar. Sidebar. Sidebar.
  Sidebar. Sidebar.</p>

  <p class="span10">Main text. Main text. Main text. Main
  text. Main text. Main text. Main text. Main text. Main
  text. Main text. Main text. Main text. Main text. Main
  text. Main text. Main text. Main text. Main text.</p>
</div>
```

Now, it's true that these class names lack any semantic value, and they are in some ways a new-fangled technique for doing tables—which, as all of us know, are a terrible non-CSS way to do layout. But, the advantages are huge. I know that these classes have made it far easier for me to experiment with layouts, moving text to different places on the page, and understanding what will happen to the rest of the items on my page when I do so.

Now, Bootstrap provides a number of classes that are not meant to be used on their own, but rather in conjunction with other classes. For example, you always can have a title that uses the h1 tag, but perhaps you really want the h1 to stick out on your home page. To do that, just add the "hero-unit" class to your h1's list of classes. The fact that each tag may contain any number of classes makes it trivially easy to add and remove design styles from Bootstrap and to experiment with what different ones will look like.

Now, it's true that h1, h2 and their friends get styled nicely by default when you use Bootstrap, setting not only the font size and boldface but also the line height. But, you can make them even snazzier than the defaults, by (you guessed it) using some of Bootstrap's classes. If your h1 is the page header, you can enclose it in a div whose class is "page-header".

Non-headlined text can enjoy some treats too. If you want some text, such as a glossary definition, to appear when the user's mouse cursor hovers over it, just wrap the text in an "abbr" tag:

```
<abbr title="GNU'S Not Unix!">GNU</abbr>
```

Text with such an abbr tag has a light underline beneath it, which allows users

to identify such text more easily.

Tables get some fancy styling as well. By using the table-striped class, you automatically can have alternate rows contain a light background color to distinguish them.

Forms

Bootstrap also has strong support for HTML forms. Like all other block-level elements, the "form" tag can take a "spanN" class, indicating its width. But the form itself will look much spiffier than would be the case without Bootstrap. Text fields are sized appropriately and get a nice shadow when they receive the input focus. Submit buttons have nicely rounded corners, and the mouse cursor changes to a pointer when hovering over them.

But wait, it gets better. Let's say you want your submit button to be a bit larger than usual. Well, just add the "btn-large" class to the "input" tag, and you'll have a larger button. You also can use btn-small or btn-mini for buttons of other sizes.

You also can colorize your buttons by setting additional classes. The "btn-primary" class will use Bootstrap's primary color (blue, by default). But, instead you can use "btn-danger" (for red), "btn-warning" (for yellow) and a host of other colors. In this way, Bootstrap is using CSS classes semantically. I find it very useful to be able to think in terms of what the

buttons are for, rather than what color they contain. The levels and colors are consistent across Bootstrap as well. If you put text within a span or div with both the "label" and "label-warning" classes, or the "badge" and "badge-warning" classes, the text will be highlighted with the warning color. You also can have button groups, labeling of check boxes and radio buttons (so people can click on the text, not just the widget), and many other features.

Navbars and Icons

If you want your site to have a simple navigation bar, you can avoid creating the HTML and CSS necessary for such a thing to work, and just use Bootstrap's "navbar" class. Whether you want a simple navbar, one with pop-up, JavaScript-controlled buttons or even a form inside your navbar, Bootstrap has all of those options available. You even can set up your navbar such that it's "responsive"—meaning that it knows how to expand and collapse based on the user's browser window width.

The "img" directory contains two images: one for all of the white-on-black icons that Bootstrap provides and another for all of the black-on-white versions of those same icons. The icons are all on one image, because they take advantage of CSS sprites, a technique that makes it possible for your browser to download a single

image and then selectively display parts of that image independently. Thus, if you have a download link on your site and you want to have a download icon, you can add:

```
<p><i class="icon-download"></i>Download</p>
```

Now, here's the thing you might notice about the icons: they're using the <i> tag, which normally is for italics. But, the text that accompanies the icon doesn't go inside the tag. Rather, it goes next to the tag. The only reason for <i> is that it allows CSS to pull up a sprite as the background image. The icons are a great start for many of the things you'll want on a site.

Conclusion

I've been doing Web development for a while and thought I had seen rapid adoption of technologies before. But, Twitter Bootstrap has beaten them all from what I can tell. It's popular, of very high quality, adheres to standards, works across platforms and languages, and is easy to use.

The only negative I can think of is fairly minor—namely that it uses LESS rather than SCSS, which means that integration is a tiny bit more complex with Ruby on Rails. But, even that integration is a no-brainer. Ruby gems exist that solve this problem in a number of different ways.

I've adopted Bootstrap in my own work and expect to adopt it on future projects as well. Bootstrap doesn't solve all the problems with CSS, but it solves so many of them and makes them so accessible, that it allows even someone like me to create an attractive site. And, after years of having to give excuses when my simple, prototype sites didn't look very attractive, I now have a way to ensure that they'll at least be minimally aesthetically pleasing.■

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 Twitter Bootstrap on GitHub is **http://twitter.github.com/bootstrap**. The documentation as well as numerous links to examples are there.

More information about LESS is at **http://lesscss.org**. Full documentation as well as program binaries are available there.

If you're interested in using Bootstrap with Rails, I suggest watching the "Railscast" screencast on the subject, which both introduces Bootstrap and demonstrates how to integrate it with a Rails application: **http://railscasts.com/episodes/328-twitter-bootstrap-basics**.

Linux Risin'

Get Your Mojo On

2012
SouthEast LinuxFest

Charlotte, NC
June 8-10th
southeastlinuxfest.org

Rocking Charlotte:

- **Build An Open Source Cloud Day**
- **LPI Exam Cram**
- **MySQL Training Day**
- **Java Hack-A-Thon**
- **Open Database Camp**
- **Drupal Camp/ Drupal in a Day**
- **BSDA Certification**
- **LPI Certification**
- **PuppetLabs Training**
- **SaltStack Training**

Come Play with Rock Stars

Cheating at *Draw Something*

DAVE TAYLOR

Now that we have a script that can help with *Scrabble* and the like, it turns out it's helpful for any game that involves words. Let's see how I tap into it to get an edge in the popular new iOS game *Draw Something*.

More words. This time I noticed that many of my friends and family are engrossed in a popular new iPad and iPhone game called *Draw Something*. It's a sort of digital version of *Pictionary*, but what makes it interesting to me is that you're presented with a sketch and have to guess what it is, but you're given a set of letters and spaces that clue you in to how long the word is. See Figure 1 for an example.

You can see in Figure 1 that you're presented with a set of letter tiles and a blank space for the word. In this case, you can see that the word is five letters long and that the set of letters from which you can choose is A D E E K N N S V W X X. By the duplication of letters, you can conclude that if the word to guess includes more than one occurrence of a letter, each letter has to appear.

This is, of course, a simple variation on the word selection programs we've been

Figure 1. What the heck is this sketch from *Draw Something*?

writing during the past few months as part of our *Words With Friends/Scrabble*

program, so let's see what's involved in making this cheat work. Let's assume that the words are in our existing dictionary.

findwords.sh

In my previous few articles, we've developed a handy script called `findwords.sh` that you give a set of letters and it tells you what words could be made out of those tiles—for example:

```
$ findword.sh sejdowa
++ word dowse can be constructed from the letters sejdowa
++ word jawed can be constructed from the letters sejdowa
```

Yes, those are particularly bad letters for any *Scrabble*-like game when there are only two words possible that are at least five letters long (by default, the script omitted all words four-or-less-letters long because usually there are a ton of them).

It seems like we could apply the same script to *Draw Something* without modification, however, simply by listing all the letters possible—more than seven, needless to say. Here's what happens when I do that, with a wee bit of tweaking the output:

```
$ findword.sh adeeknnsvwxx
++ word ANNEX can be constructed, length = 5
++ word ASKEW can be constructed, length = 5
++ word DENSE can be constructed, length = 5
++ word EAVES can be constructed, length = 5
++ word EVADE can be constructed, length = 5
++ word KNAVE can be constructed, length = 5
```

```
++ word KNEAD can be constructed, length = 5
++ word NAKED can be constructed, length = 5
++ word NEEDS can be constructed, length = 5
++ word SEDAN can be constructed, length = 5
++ word SENNA can be constructed, length = 5
++ word SEVEN can be constructed, length = 5
++ word SKEWED can be constructed, length = 6
++ word SNAKE can be constructed, length = 5
++ word SNEAK can be constructed, length = 5
++ word SWANK can be constructed, length = 5
++ word VEXED can be constructed, length = 5
++ word WAKEN can be constructed, length = 5
++ word WAXEN can be constructed, length = 5
++ word WEAKEN can be constructed, length = 6
++ word WEAVE can be constructed, length = 5
++ word WEEDS can be constructed, length = 5
```

The only tweak I'd make is that we should specify the length of the word and have it show us only matches that could fit; showing a six-letter word like WEAKEN isn't very useful.

To fix that, let's add an optional second parameter to the script that specifies target word length. Being lazy, I'll make that parameter #2 and use the extraordinarily sloppy parameter test of:

```
if [ ! -z "$2" ] ; then
  targetlength=$2
fi
```

Yeah, yeah, `getopt` would be better, but as I've discussed before, sometimes scripting is more about getting the job done than it is about being optimal in

your coding strategy.

The main loop continues to increase in complication too, now adding the `targetlength` parameter:

```
for word in $(cat $possibilities)

do

length=$(echo $word | wc --c)

  length="$(( $length - 1 ))"

  idx=1

  while [ $idx -le $length ] ; do

    letter=$(echo $word | cut -c$idx)

    occurrences $letter $word

    wordfreq=$freq  # number of times letter occurs #1

    occurrences $letter $1 # and letter occurrences #2

    uword=$(echo $word | tr '[[:lower:]]' '[[:upper:]]')

    if [ $wordfreq -gt $freq ] ; then

      break # get out of the "nearest" loop

    else

      if [ $idx -eq $length ] ; then

        if [ $targetlength -ne 0 ] ; then

          if [ $length -eq $targetlength ] ; then

            echo "word $word can be constructed"

          fi

        else

          echo "word $uword can be constructed, length = $length"

        fi

      fi

    fi

    idx=$(( $idx + 1 )) # increment loop counter

  done

done
```

If you're comparing this with the previous versions of this loop, you'll notice a few changes, including the `uword` assignment that converts

words from lowercase to uppercase. Most important, if `targetlength` is nonzero, the script will list only words that match the target length.

Now the results are a bit trimmed down, but let's go a bit further and try to get all the results on a single line by simply having the word itself output, adding a before-and-after wrapper line, and pushing the entire output through the ever-handy `fmt` command:

```
$ findword.sh adeeknnsvwxx 5 | fmt

Possibilities: ANNEX, ASKEW, DENSE, EAVES, EVADE, KNAVE, KNEAD, NAKED,

NEEDS, SEDAN, SENNA, SEVEN, SNAKE, SNEAK, SWANK, VEXED, WAKEN, WAXEN,

WEAVE, WEEDS, out of the letter set adeeknnsvwxx.
```

That's a good place to stop. Oh, and did you figure out the solution to the *Draw Something* puzzle yet? I hope so!

The only problem is that this isn't a great cheat in this particular instance, because one of the things that makes *Draw Something* popular is that it uses a lot of contemporary words like Lady Gaga, NASCAR and JayZ—words that aren't going to be in our dictionary. Maybe that's another reason it's popular—unlike *Words With Friends*, it's hard to cheat at *Draw Something*. Ah well, it's still an interesting extension to our earlier word finder script! ∎

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.

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

Getting Started with 3-D Printing: the Hardware

KYLE RANKIN

Want a 3-D printer but are overwhelmed with all the options? Find out what printer best suits your personality and pocketbook.

I've been interested in 3-D printers ever since I saw one at a Maker Faire a few years ago, but it was only a year ago when I started seriously thinking about having one of my own. At that point, I started to realize just how many different options existed and ultimately started researching the RepRap family of 3-D printers (more on the different printer families below). After about a year of research, I finally settled on a printer that fit my needs and my budget.

During my research, I found that 3-D printing was an even more vast world than I imagined before. Not only are there lots of choices with the hardware, even when it comes to the software, you have a lot of options. In this article, I give a general overview of 3-D printing hardware, and then in my next article, I will discuss some of the current most popular software to control your printer.

So if you have thought about getting into 3-D printing and were wondering how it works with Linux, these columns should give you a good overview from the perspective of someone who's relatively new to 3-D printing himself.

What Is 3-D Printing?

When most geeks talk about 3-D printing, they are talking about some method to create three-dimensional objects much like a regular printer. Although most hobbyist 3-D printers work with plastic, there also are efforts to print in all sorts of other materials from metals to ceramics to organic materials. Even if you narrow things down to talking only about 3-D printing in plastic, there still are all sorts of methods 3-D printers can use. That said, basically all of the home 3-D printers operate via plastic extrusion. If

you have used a hot-glue gun before, you know that you load the back of the gun with a large glue stick, and once the tip of the gun heats up, your trigger forces the glue stick toward the hot end of the gun where it extrudes in a much narrower blob of glue. Imagine taking a three-dimensional object and slicing it into layers as thick as the hot glue from your hot-glue gun, and then imagine carefully squeezing the hot glue out onto a surface layer by layer until you got your object.

Plastic extrusion works in a similar way to a hot-glue gun except your plastic comes in the form of a long filament of ABS or PLA plastic 1.75mm or 3mm in diameter. The 3-D printer's extruder forces the filament into the hot end of the printer, which heats up enough to melt the plastic (160–190°C is average for PLA, and 200–250°C for ABS). The melted plastic is then pushed through a much narrower nozzle (.5mm to .35mm normally) onto a print bed.

If all a 3-D printer did was melt plastic like a hot-glue gun, it would be great for creating blobs of melted plastic but not much else. What makes the 3-D printer useful is that this extruder is mounted on a stable frame with precise X, Y and Z axis motors controlled by custom electronics. When you send a 3-D diagram to the printer, it is sliced into individual layers, and each layer is represented by a series of X, Y and Z movements, along with instructions to extrude or retract the plastic filament at appropriate points. The result is that your object is printed layer on top of layer.

Types of 3-D Printers

If you are new to 3-D printing, the number of options available can be overwhelming at first. Not only do many of the printers look relatively different from each other, there also is a wide range of prices for 3-D printer kits, from around $500 to more than $2,000 (which is much better than $15,000+ for commercial models). Plus, if you are resourceful, you even can bypass the kits and source all your own parts—it's more effort on your part, and there's a greater chance something might not work, but you can cut down the cost rather dramatically in some cases.

What I've found in my research is that 3-D printers are a lot like Linux distributions. If you were to ask everyone who had 3-D printers which one to get, you'd get about as many replies as if you asked all the Linux users what distribution to use. Some 3-D printers appeal to newbies, and others appeal to experienced users. Some 3-D printers focus on how open their hardware and software is, and others take a more commercial approach. Sound familiar?

I'm not going to enumerate every
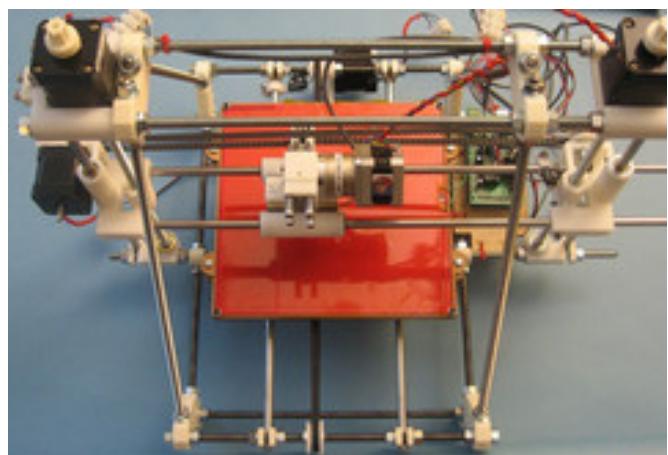
single 3-D printer out there, so I'm sorry if I leave out your favorite one, but I've found when you look at what 3-D printers the majority of people use, they tend to fall in two big categories: laser-cut wooden-box 3-D printers and RepRap-based printers. Like with Linux distributions, the 3-D printers I'm discussing generally follow open-source principles not just in their software but also in their hardware. Also note that although in some cases you can spend extra to get a pre-assembled kit, most of the kits I mention come unassembled and will require many hours to assemble and calibrate.

Laser-Cut Wooden-Box 3-D Printers

The first category of printer is most identifiable by the fact that the whole printer is enclosed in a wooden box that's usually created by a laser cutter. Printers that fit this category include the MakerBot family of printers, the Ultimaker and the Mosaic printer from MakerGear. Although all of these printers are different, they generally appeal more to people who are new to 3-D printing and want a more polished appearance to their printer. This more-polished appearance means more parts, and unlike the RepRap family of printers, most people who go this route buy a kit with a complete set of parts, and as a result, the price can be higher, starting at $900 for an unassembled Mosaic kit to $2,000 for an all-bells-and-whistles

Ultimaker (photo from http://blog.ultimaker.com)

MakerGear (photo from http://www.makergear.com)

MakerBot Replicator (photo from
http://store.makerbot.com/replicator.html)
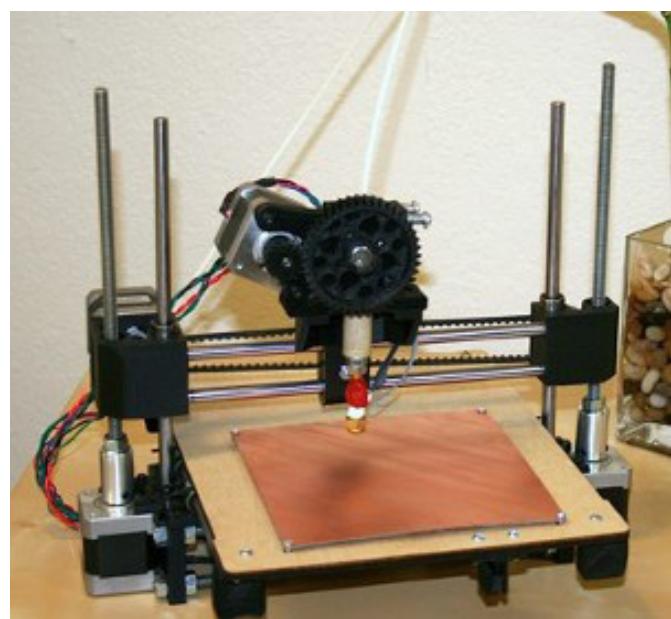
dual-extruder MakerBot Replicator.
If you have the money, you can get
a nice-looking 3-D printer in this
category that functions well, but just
don't assume that the extra money
necessarily buys you better specs.
Think of it like buying a commercial
Linux distribution versus downloading
a community-supported one.

The RepRap Family
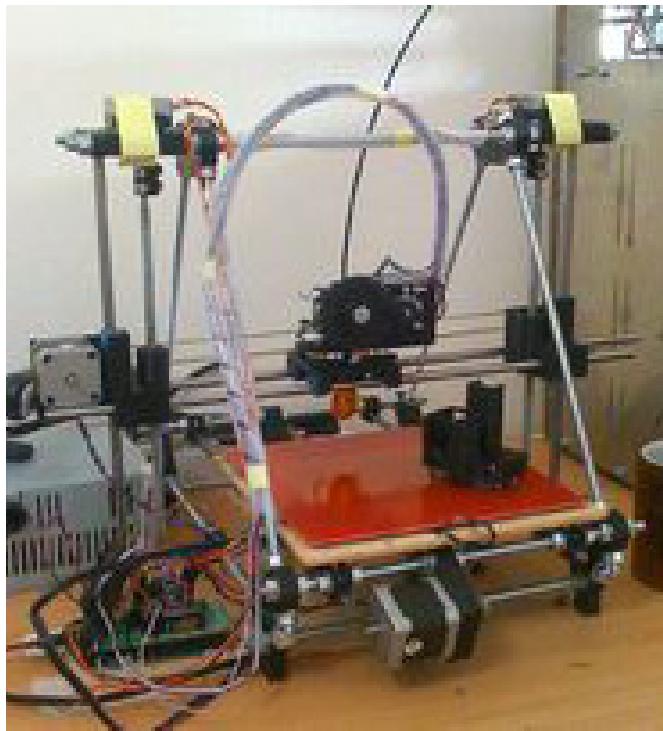
If the wooden-box 3-D printers are
like commercial Linux distributions,
you can think of the RepRap family
like Debian. RepRap is a community-
driven project that aims to build a
3-D printer that can create many
of its own parts and self-replicate.
Because of the community-focused
design process, anyone can propose
improvements or new RepRap

derivatives, and successful designs get
rewarded with popularity. There are
a number of generations of RepRap
designs, but the Prusa Mendel seems
to be the design most people in the
RepRap community recommend to
newbies today. The community also
has built incremental improvements
and additions to the design—many
of which you can download from
3-D printer design sharing sites like
http://www.thingiverse.com and
print out yourself, so you can continue
to improve your printer as you use it.

As you might expect, there are a
number of different ways to acquire
a RepRap, starting with sourcing your
own parts on-line from published bills of
materials on the RepRap Wiki to buying
completed kits from other members of
the RepRap community on eBay. Some
members of the RepRap community

Printrbot (photo from http://printrbot.com)

Prusa (photo from http://www.makergear.com/
products/3d-printers)

have even gone on to start their
own businesses selling RepRap parts,
including sites like **http://lulzbot.com**,
http://makergear.com and
http://printrbot.com, the latter being
a new Kickstarter-funded company that
sells a low-cost RepRap derivative that
has a more-simplified design.

Although the RepRap is as capable
(some would argue more capable)
when compared to the wooden-box
printers, people tend either to love or
hate the simplified metal-rod design.
The simplified design results in a lower
cost in general though, with self-
sourced RepRaps as cheap as $400 or
$500, a complete Printrbot kit starting
at $550, and complete Prusa Mendel
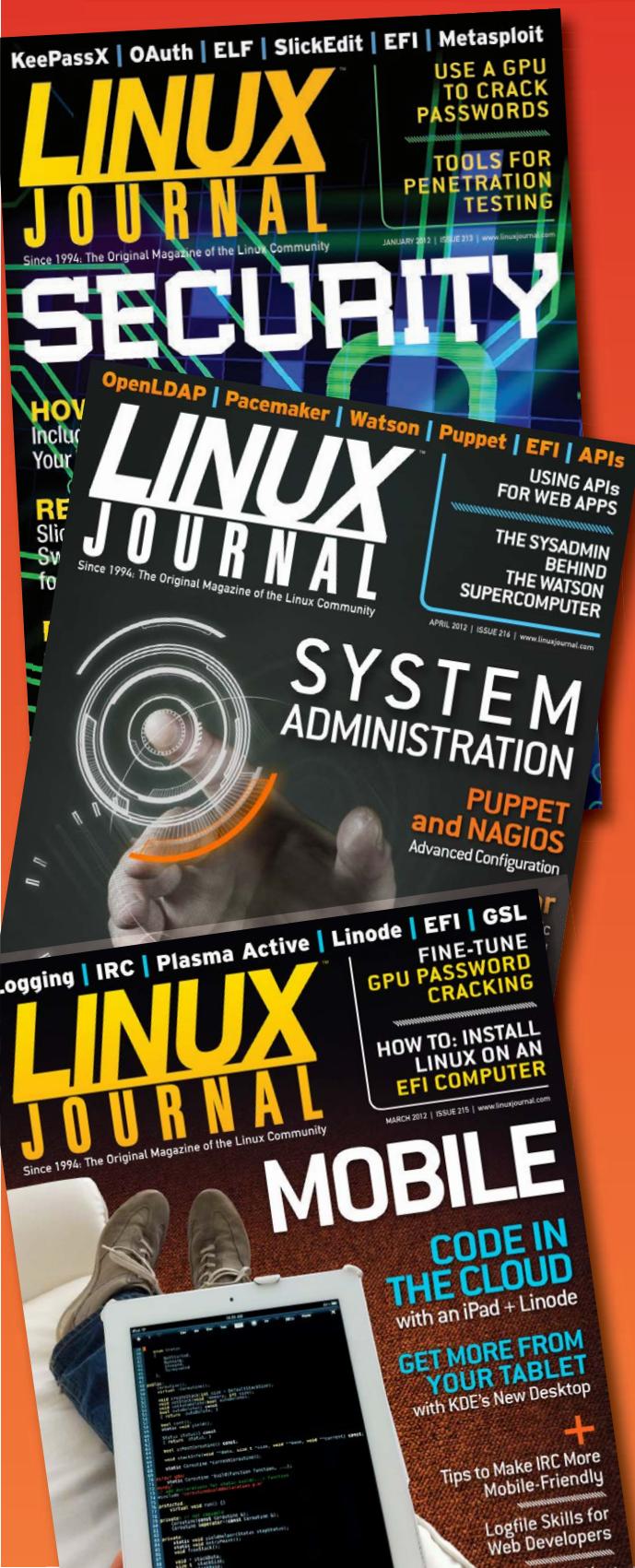
kits usually starting around $800 or so.

So what printer should you
choose? That's like asking what Linux
distribution you should use. As with
Linux distributions, it really comes
down to people's personalities and
what they want to accomplish. For
instance, the RepRap community is
full of people who love to build and
calibrate 3-D printers and tinker with
them to get the most out of them, so
if you are the kind of person who likes
to fine-tune and tinker with your Linux
distribution, that class of printers may
be right up your alley. On the other
hand, the laser-cut wooden-box 3-D
printers seem to appeal more to folks
who don't find tinkering with the
3-D printer itself as appealing and are
more interested in printing objects.
If you are the kind of person who
leans more toward the commercial
Linux distributions that focus more
on working out of the box, these
printers might appeal to you more. In
my case, I bought a basic Printrbot kit
because I tended to lean more toward
the RepRap side of things as far as
tinkering went, but in particular, the
simplified design and the lower overall
price to get started on the Printrbot
appealed to me.■

**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.**

Files, Served Fresh Daily

SHAWN POWERS

Linux can serve files to any computer, and most never know it's Linux!

Much like a Taster's Choice commercial from the 1980s, Linux advocates often need to be sneaky in order demonstrate the virtue of open-source software. It seems that fear, uncertainty and doubt (FUD) tend to make people dismiss Linux before it ever gets a chance to prove its worth. The easiest way to infiltrate an organization with Linux is to attack the server room.

Although hordes of penguins armed with machine guns sounds like a blockbuster way to attack a server room, in reality, the best way to get Linux in the door is by using it to replace a few services previously served by proprietary operating systems. The simplest service to replace is usually the Web server. File serving, although a bit more complex, also is effective. When an entity is growing past its licensing, free file services really begin to look attractive.

NFS, Simple and Insecure

NFS often gets a bad rap for being insecure. That's really only because it's *so insecure*! Seriously though, even though NFSv4 has added user authentication, to use NFS in the traditional way, it's important to realize its shortcomings. Many folks (myself included) still use NFS, but it's easy to expose your servers to unneeded threat if you don't understand how it works.

On the server side, you can specify the IP address of the clients allowed to connect. Unfortunately, it's not uncommon to see someone use wild cards to allow any and all computers to connect. That's a bad idea, because NFS (v2 and v3) restricts file access using only standard user permissions. If I connect to an NFS server with my laptop and my UID is 501, I'll have access to the files owned by UID 501 on the server— without authentication. There are options to restrict the root user, but that doesn't really restrict anyone. If I want to access files owned by UID 6224, all I have to do is create a user with that UID, and I

have access. Squashing root access on the server is no security at all, just an annoyance to potential hackers.

Because restricting the client IP addresses that are allowed to connect limits the clients to trusted computers, it would seem that NFS is secure as long as the server is configured correctly. Sadly, that's not really true either. As there is no way to verify what computer is using a particular IP, an attacker simply can shut off an allowed client machine, assign the IP address of that client and connect to the NFS server quite happily. For that reason, I restrict my NFS usage to server-to-server sharing on servers in secured locations. Even then, it's important to think through any potential point of entry.

The reason I bring up NFS sharing is because it's perfect for LTSP. Because servers in the server room are secure and physically adjacent, NFS is a simple way to share the /home directory between your LTSP servers. Configuring NFS is simple. Make sure the NFS server packages and its dependencies are installed on your server, and the NFS client packages are installed on your clients. (In this case, by "client" I mean the server acting as an NFS client to the file server. Sorry if that sounds confusing.)

On the server, edit your /etc/exports file. The following example has three separate shares. See if you can spot the "bad" one:

```
/home     192.168.1.35(rw)
/iso      192.168.1.*(ro)
/home     192.168.1.*(ro)
```

To activate the shares you've created in your /etc/exports file, either reboot the server, or type `/usr/sbin/exportfs -a` to reload the configuration.

In the above example, the first line allows the client with the IP address of 192.168.1.35 to have full read/write access to the /home directory on the server. This is about as secure as you can make NFS on the server. You also can specify hosts with DNS entries, but then you risk compromise by DNS spoofing.

The second line shares the /iso directory in read-only mode to anyone on the 192.168.1.0/24 network. The wild card here means what you'd expect, and anyone in that range can connect. Making this read-only does protect the content of the folder from being molested, but it doesn't stop anyone from reading and copying the contents. In this case, that's the desired result.

The third line looks to be safe, because it's shared only in read-only mode. As any client on the network can connect, however, that means every folder in the /home directory, so that means usually all your users have their personal files exposed! Granted, no one can change the files because they are read-only, but with that single line, you've effectively given free access to everyone's files—definitely not a good idea.

On the client computer, NFS shares are mounted just like any other partition or filesystem. To make a mount permanent and automatic, simply edit /etc/fstab and add a line like:

```
#<server:/share> <mountpoint> <filesys_type> <options> <dump> <pass>

192.168.1.5:/home   /home    auto    defaults  0   0
```

To mount the share, either reboot the client machine or run `mount -a` as root. It's important to note that like any filesystem mounting, the /home directory must be empty in order to mount the NFS share in that location.

The above example tells the client to try connecting to the server 192.168.1.5 (it also can be a DNS name). Once connected, it tries to mount the /home share on the local /home directory. The filesystem type is set to `auto`, and the options in this case are set to `defaults`. The last two zeros are the `dump` and `pass` options. The options along with dump and pass settings are tweakable, but beyond the scope of this article.

When NFS is used in a closely controlled server-to-server scenario like this, it makes sharing home directories very convenient. Once you realize the shortcomings of the "security" NFS offers, it is possible to include it in very specific scenarios in your infrastructure.

Samba, or CIFS

Samba is one of those programs that can wash your car, feed the dog, fly a space shuttle and also serve files. It can act as a Primary Domain Controller (not an Active Directory server yet, unfortunately), along with sharing printers and files. For the purposes of this article, let's take a cursory look at file serving—specifically, sharing the /home directory for your users. By installing Samba on the same file server you installed NFS, those same files can be shared to Windows clients. Oh, and Samba *does* support user authentication for security!

Configuring Samba could easily turn into a five-part series, but if your goal is simply to serve files in your /home directory, I recommend starting with the default configuration. If you're running a Debian variant, I recommend installing the libpam-smbpass package along with the samba server packages. It is a neat little package that keeps the separate UNIX and Samba passwords in sync. It's certainly possible to configure other authentication schemes, but in its simplest form, keeping user names and passwords in sync with the underlying system is ideal.

If you open the /etc/smb.conf file, you'll see some settings in the `[global]` section that can be tweaked to best fit your network. Namely, you can set the NETBIOS name and a few other tweakable configurations to better define your server. Even those changes, however, are not required. Simply scrolling down and uncommenting the
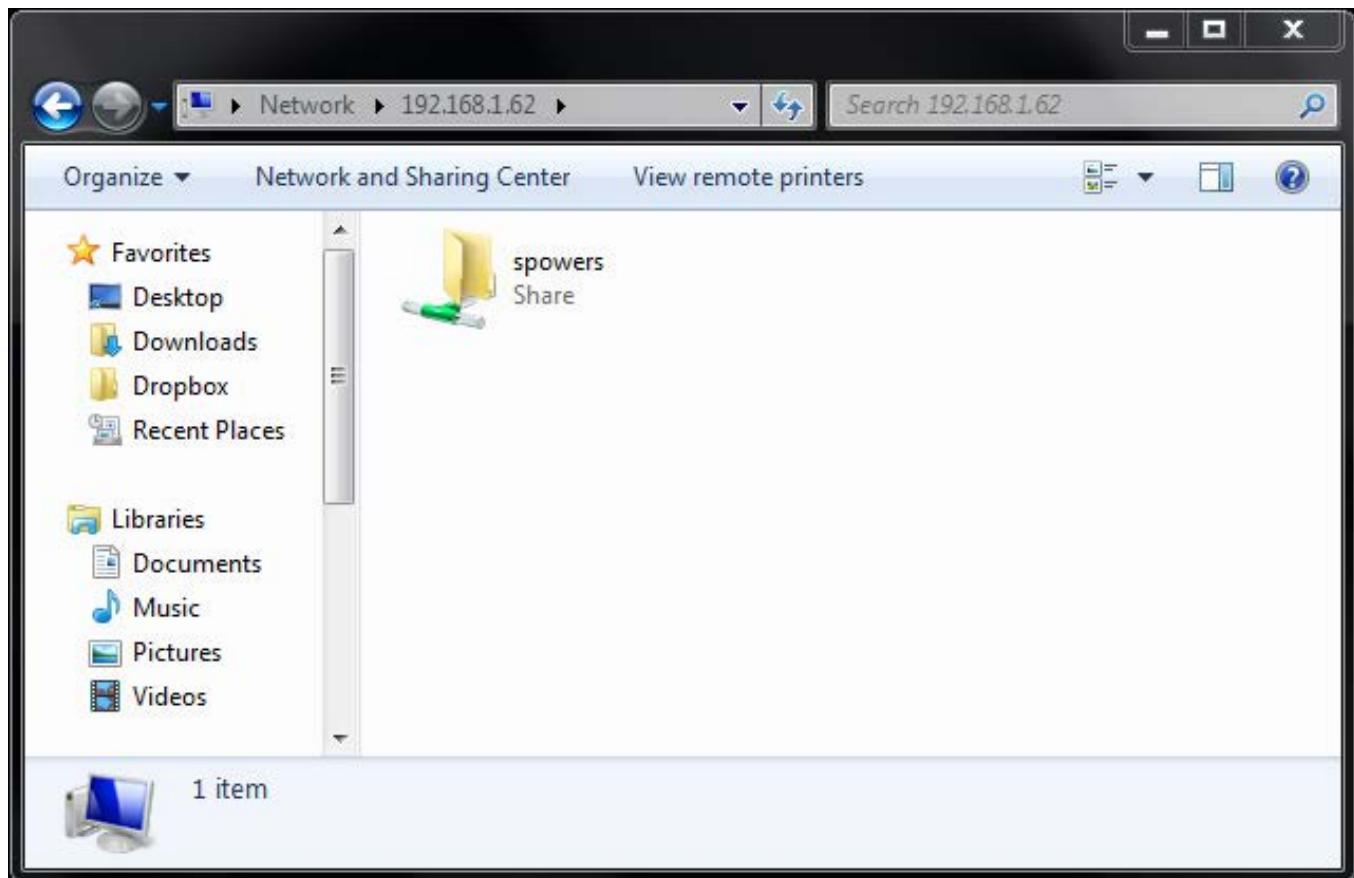
Figure 1. As far as Windows is concerned, it's connected to another Windows machine.

following three lines will get you a functional Samba server:

```
[homes]
    comment = Home Directories
    browseable = no
```

After changing those lines, you should be able to go to a Windows machine and browse your server as if it were a Microsoft server (Figure 1). As I mentioned earlier, Samba is a vastly powerful system. Sharing your /home directories to users is barely scratching the surface. Still, even with this little bit of setup, it's quite useful.

Fruity Computers Too!

With modern OS X machines, connecting to Samba shares is simple. In fact, OS X does its own magic to preserve the resource fork of its files by adding a ._filename file to Samba shares. If you're not familiar with Apple's data fork and resource fork file structure, don't worry. The only frustrating part is all those dot-underscore files everywhere!

If you prefer your file sharing to be native, there's an app for that. As it's not only Windows servers that you may hope to replace with Linux, it's nice to have an Apple native file server even if OS X can connect to Samba. Netatalk

is the server program that provides file and print services for Apple products. Apple actually uses LDAP for its Open Directory server, but the configuration is proprietary and uses closed-source management tools. So for the short term, the only thing you can do seamlessly is share files and printers.

The package that comes with most Debian variants comes preconfigured for serving home directories. A simple `apt-get install netatalk` will install and configure the server without any other effort required on your part. Users then can connect to your server as if it were an Apple-branded file server and never know the difference. For production use, you likely will want to edit the configuration files to set the name of your server, but it really does usually work right out of the box. OS X users often claim their system "just works", and you can claim the same about your open-source server program for their platform!

But I Use Linux for a Client!

You've probably noticed that my solution for Linux clients appears to be the most limited. Rest assured, if you're running Linux on a client that isn't appropriate for NFS, you can connect via Samba without any problems. It might be possible to connect to a Netatalk server from Linux, but I've never seen a reason to try. If you don't want to use Samba, there are other less-common ways to connect.

SSHFS

LTSP thin clients mount a user's home directory using this method. Years ago, if the thin client had to access the /home folder, it was mounted via NFS. For reasons I outlined above, it's clear that isn't a really good idea. As its name suggests, SSHFS mounts a remote filesystem over SSH. This makes mounting very flexible, but it's important to keep in mind that shares mounted over a slow WAN connection often are difficult to use, especially on a regular basis.

In order to use SSHFS in a Debian- or Ubuntu-based system, you need to install the sshfs package. Just like NFS, the folder on which you choose to mount needs to be empty or the mounting process will fail. (Note: the Ubuntu Web site claims users must be in the "fuse" group in order to use sshfs filesystems, but I have not found that to be the case.) To mount a remote filesystem:

```
sshfs user@remote.system.com:/remote/folder /home/user/local_folder
```

Once authenticated, the folder /home/user/local_folder (or whatever folder you chose) will be mounted remotely to /remote/folder on the server. Access speed is dependent on network speed, but the files should work the same as any other mounted filesystem. It's really cool!

In order to unmount, use the

`fusermount` command:

```
fusermount -u /home/user/local_folder
```

It is possible to add sshfs filesystems to the /etc/fstab file in much the same way that you add an NFS share. Because sshfs uses FUSE, a userspace file-mounting system, it's generally used for individual users as opposed to system mounts. As with most things Linux, this is flexible.

Old-School FTP

FTP isn't really ideal for file transfers, mainly because it transfers user names and passwords in plain text. Linux still supports FTP file servers, and most systems come with FTP logins jailed in a chroot environment for each user. Although this protocol is insecure, unfortunately, you need to use it sometimes for programs that support only traditional FTP. If I have to set up an FTP server, I prefer vsftp, but I won't go through the setup process here because FTP is just not a great protocol to use.

The "proper" thing to do is use SFTP, which for Linux users is as simple as having an SSH server installed. SFTP is handled by the OpenSSH dæmon on Linux servers, and setting it up on most servers is as simple as doing nothing. If there are security concerns about exposing your server to the Internet, it is possible to configure

SSH for chroot jails per user as well. Thankfully, with SSH everything is encrypted, so sniffing the network isn't a concern. Unfortunately, no amount of securing can protect your system from compromised passwords, so make sure your users don't have a habit of using passwords like 1234 for your system.

Conclusion

The server room is by far the easiest place to replace expensive proprietary operating systems with Linux. In fact, the days of sneaking Linux into the server room are largely behind us. Most bosses, even the technologically illiterate ones, expect Linux to be in the server room. They might not know what Linux does, but they know they should have it! If your institution is outgrowing its licensing, or if you'd just like to replace "proprietary" with "open", file servers are a great place to start. If done carefully, your users will never know. Be sure to mention to your boss how much you've saved in licensing and support, because upper management always likes to brag about how much the department has saved!■

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.
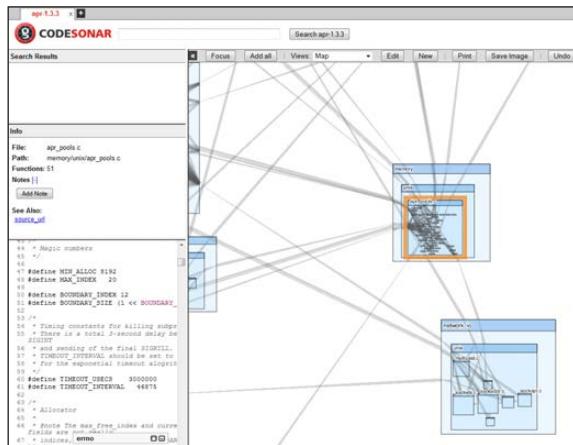
Digium Inc.'s G100 & G200 VoIP Gateways

Digium, Inc.'s newly released standalone G100 & G200 VoIP Gateways are an integral part of the company's mission to simplify further the deployment of Asterisk-based, converged-media networks. The gateways, which are built on a combination of the Asterisk open-source communications engine and an advanced embedded platform, complement the heretofore standard solution of telephony ports as interface cards on standard PC hardware. The power-saving embedded design features a highly efficient digital signal processor (DSP) to handle all media-related operations. The G100 & G200 are built to support both TDM-to-SIP and SIP-to-TDM applications. Digium says that the combination of the company's intuitive user interface, the flexibility of Asterisk and the purpose-built media processing capabilities of the DSP results in a gateway platform that outperforms the "dated designs" found in the market today.
http://www.digium.com

GrammaTech Inc.'s CodeSonar
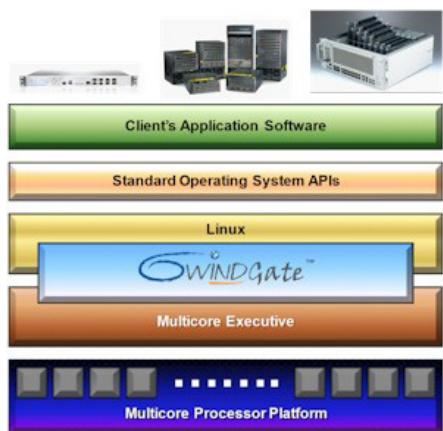
While text editors typically show a scant few lines of code, the upgraded CodeSonar and its new architecture visualization system from GrammaTech, Inc., scales to millions of lines of code. The new system tackles the growing complexity of large software systems, allowing developers to test and analyze the interrelationships between elements of source code via visual inspection and analysis of their software. Real-time, fluid transitions allow for the visualization of massive data sets at a wide range of levels of abstraction. GrammaTech says that CodeSonar goes beyond what is offered in other visualization tools, which are effective at displaying small-scale parts of the program structure, such as fragments of class hierarchies and subsets of the call graph. CodeSonar, says the company, is unique in its ability to show whole-program structures as well as deal with complexity found in certain smaller projects. CodeSonar visualization runs through a standard Web client, such as Mozilla Firefox, Google Chrome or Microsoft Internet Explorer.
http://www.grammatech.com

6Wind's 6WINDGate Cloud Edition

The new 6WINDGate Cloud Edition from 6WIND is targeted at suppliers of both physical and virtual network appliances for data centers. These suppliers, says 6WIND, are under intense pressure not only to deliver new cloud-optimized solutions with the best possible cost/performance but also to accelerate their development schedules and be early to market in new product categories. Building on the success of its Mobile Edition software, which already is deployed in two-thirds of LTE networks worldwide, 6WIND has developed the Cloud Edition to apply its unique technology to address critical performance and scalability challenges faced by cloud infrastructure providers. The 6WINDGate Cloud Edition product includes an integrated suite of optimized networking protocols matched to the specific requirements of network appliances. These include protocol termination, QoS, routing, switching, security, tunneling and VPN functions. Support for industry-standard hypervisors also is included.
http://www.6wind.com
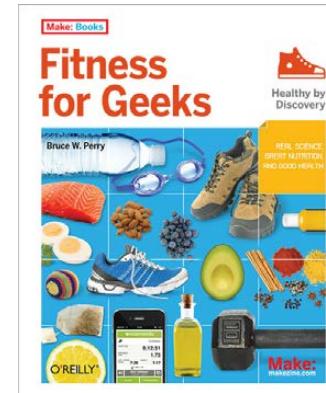
Bruce W. Perry's *Fitness for Geeks* (O'Reilly Media)

"There's more to being a geek than most people think when they hear the word", says O'Reilly Media—which is why the publishing company sees no sign of irony in the title of its new book *Fitness for Geeks: Real Science, Great Nutrition, and Good Health* by author Bruce W. Perry. The book is a treatise on the science behind one's health and exercise routine, which O'Reilly says will appeal to a broad audience of scientists, programmers and people with inquisitive minds who want to experiment with health the way they tinker with technology. By applying the same technical approach

they use to debug software or hack hardware, readers will learn both what to do and why it works. Whether they have an established exercise routine and want to dig deeper to realize the physiology behind it, or if they're just looking for a way to get away from the computer to exercise in a manner that speaks to them, this book helps readers discover a new method for building and maintaining fitness and a healthy lifestyle.
http://www.oreilly.com

Michael J. Hammel's *The Artist's Guide to GIMP*, 2nd ed. (No Starch Press)

Michael J. Hammel is one of the ur-names associated with the GIMP graphic design and photo manipulation application. Back in the 1990s, the producers of *Linux Journal* published the first edition of Hammel's book *The Artist's Guide to GIMP*, which No Starch Press recently released in a fully updated 2nd edition. The new edition, which carries the subtitle "Creative Techniques for Photographers, Artists, and Designers", illustrates how to put the GIMP's powerful tools to work to produce professional-looking advertisements, apply impressive photographic effects, and design cool logos and text effects. Hammel demystifies the notoriously challenging elements of the GIMP not with a boring list of menu paths and options but with step-by-step tutorials that help to accomplish important tasks. After a crash course in GIMP's core tools like brushes, patterns, selections, layers, modes and masks, the reader will learn photographic and Web-design techniques, as well as type and advertising effects. Readers of all levels will be inspired by the art that they can create.

http://www.nostarch.com

BeyondTrust's PowerBroker Servers Enterprise

The new PowerBroker Servers Enterprise solution from BeyondTrust was created to confront the challenges related to managing Linux and UNIX user accounts in heterogeneous networks. BeyondTrust describes PowerBroker as "a cost-effective approach to allow organizations to seamlessly manage privileged access on Linux and UNIX systems across dynamic physical, virtual and cloud environments through Microsoft Active Directory". The result is a reduction in costs, complexity and susceptibility to security breaches, especially inside jobs. PowerBroker provides an intuitive GUI to implement privileged access policy and the ability to log, monitor and report on all user access and activities. This approach empowers IT organizations with the ability to delegate root tasks and authorization on Linux, UNIX and Mac OS X platforms without ever disclosing the highly sensitive root password. Furthermore, businesses of every size and industry are better able to satisfy critical internal and external compliance requirements and regulations, including SOX, HIPAA, GLBA, PCI DSS and FISMA.

http://wwww.beyondtrust.com

Napatech Time Synchronization Endpoint

As the world moves to 10GbE and 40GbE, and the number of packets that need to be analyzed in data centers and telecom reach tens of millions per second, the need for time stamping with nanosecond-level precision is growing. Enter Napatech's new Time Synchronization Endpoint (NTTSE), a new time synchronization endpoint unit supporting the IEEE1588 PTP time synchronization protocol. NTTSE can be installed as an IEEE1588 PTP slave, allowing Napatech customers to synchronize all appliances in their data center using Napatech intelligent network adapters to an IEEE1588 PTP master. NTTSE is suitable for applications in which high-precision time stamping of captured packets is required, such as latency monitoring for financial networks, telecom protocol analysis or Ethernet network performance monitoring.

http://www.napatech.com

S.E.C.U.R.E. Virtual Computing Platform

Three disparate companies—Fritz Technologies, LynuxWorks and TransLattice—joined forces to create the new S.E.C.U.R.E. Virtual Computing Platform to meet the exacting requirements of sensitive cloud deployments in government and commercial data centers. The companies posit that S.E.C.U.R.E., short for Secure, Enterprise, Cross-Domain, Unified, Resilient Environment, "creates the ideal environment for situations requiring secure hosting of applications, geographic redundancy of applications and data, and secure cross-domain transfer of information." At S.E.C.U.R.E.'s core is Fritz Technologies' secure server-based separation kernel (SepKer), which was created working with the LynxSecure 4.0 separation kernel from LynuxWorks. SepKer virtualization is complemented by TransLattice Application Platform (TAP), a "lattice" application and data distribution solution that adds resiliency and efficiency. TAP fully distributes both the application and its data across multiple VMs and system platforms within a secure domain, ensuring nonstop operations, even in the case of the loss of one or more components of the computing lattice.

http://www.translattice.com, http://www.lynuxworks.com and http://www.fritztechnologies.com

HARDWARE

Polywell i2303— the Powerhouse in Nettop Clothing

Polywell's i2303 packs a ton of awesome into a teeny-weeny little box. SHAWN POWERS

There's a lot to love about the nettop concept. The small form factor, the quiet operation, the power efficiency—they all make a nettop seem like the perfect casual workstation. Unfortunately, along with all those good features comes the unwelcome reality of slow performance and underwhelming responsiveness. I recently had the opportunity to review the Polywell i2303, which takes two concepts and mashes them together to form a product much better than the sum of its parts. Much like the combination of peanut butter and chocolate, this workstation from Polywell is better than a nettop and better than a powerhouse workstation. It's both.

The i2303 is marketed for use in digital signage, HTPC and as a Linux workstation. For this review, I focused on the latter. Because the market is full of great HTPC units, I wanted to see if this nondescript box could really cut the mustard as a workstation. The answer is an overwhelming *yes*.

Construction

The body of the i2303 is about the size of a standard nettop device, or maybe just a bit bigger. The chassis is 1.55″ wide, 6.6″ deep and 9.1″ tall. The included stand makes it a bit bigger, but that's the point of the stand— to help it stand up. The entire unit, including the stand, is metal. It's quite solid and feels strong enough to use as a jack stand for your car. (I'm sure that would void any warranty, so I don't suggest testing my theory.)

The design is simplicity at its best.

The front sports a power button, activity LEDs, microphone and headphone jacks, a USB 3.0 port and an SD card slot (Figure 1). Both sides are vented, but the metal chassis is thick enough that the vents don't weaken the structure at all.

Figure 1. Silver or black—no more, no less (photo from http://polywell.com).

Flipping the unit over reveals a generous assortment of ports (Figure 2): a rear analog speaker jack, an external 802.11n antenna (removable with standard connector for larger antenna), an optical audio port, HDMI, four USB 2.0 ports, an eSATA port, Gigabit Ethernet, DVI and power connector. The

Figure 2. Thank you for not skimping on USB ports, Polywell.

power is a 19V DC connector, which is common in computers this size. The silence of this noiseless design is a fair trade for the inconvenience of an external transformer. The included power cord is very long with the transformer in the middle of its length, so it doesn't take up more than a single plug like wall warts often do.

Notable on this unit is a lack of a standard VGA port. Although a DVI-VGA converter dongle certainly will accommodate a VGA monitor, it does hint to the unit's higher-end internals. The massive copper heatsink is visible from the back as well. That too hints to the unit being more than meets the eye, but from an aesthetic standpoint, I think the copper looks cool showing through.

The Guts

A peek inside (Figure 3) shows pretty much what you'd expect. The design is more akin to a high-end laptop than a desktop, but with the nettop form factor, that makes sense. Most notable is the large heatsink, efficiently designed to get heat away from the CPU and GPU with the aid of a blower fan. I've been hinting this whole time about the unit's specs, but here is a list of the specifications this review unit boasts:

- Intel Core i5-2467M mobile processor.

- NVIDIA GeForce 520 graphics card with 1GB graphics memory.

Figure 3. The compact design necessitates the well-placed heatsink (photo from http://polywell.com).

- DVI dual-link graphics port supports 2560x1600 high-resolution display.

- HDMI output supports dual-display configuration.

- 8GB DDR3 memory.

- 20GB SLC mSATA SSD (super speed) for OS and swap files.

- 500G 2.5" SATA 7200RPM hard drive for data storage.

■ Gigabit Ethernet, 802.11n Wi-Fi + Bluetooth.

■ Ubuntu Linux.

■ DC19V low-voltage power adapter.

Reading the specifications is more like reading the specs for a high-end desktop machine than a compact unit like this. Since I opened the box before looking at the packing slip (my wife says this is a typical guy move), I did a double-take when first looking at the hardware list.

The CPU is a mobile chip, but it's an i5 Sandy Bridge processor. Because the i2303 looks like a unit that should be sporting an Atom processor, the i5 option is impressive. Although the larger CPU is certainly part of the reason this unit has a cooling fan, even under heavy load, the unit is very quiet.

Graphics with computers of this size are always an interesting topic. Although most nettop units support acceleration for 1080p video playback, they usually fall short for other GPU-intensive tasks. I'm only a casual gamer, but with the i2303's NVIDIA GeForce 520 chip, I was able to play 3-D shooters at a fairly high resolution without notable frustration. Although this certainly isn't a gaming rig, the GPU was capable enough for more than just *Solitaire*. In fact, this is

With all the impressive features the i2303 boasts, my absolute favorite feature is the hard drive system.

the unit I used for the *Trine 2* review in the UpFront section of this issue, and it played quite nicely.

I didn't test the memory speed, since for the most part memory is as memory does, but the *amount* of memory included is impressive. Again, the tiny plain-Jane case is misleading, and 8GB of DDR3 memory was a welcome surprise. This boon of memory may be part of the reason large games worked so well, as the computer had plenty of room for memory-intensive applications.

The Hybrid Beast Within

With all the impressive features the i2303 boasts, my absolute favorite feature is the hard drive system. In Figure 3, you can see there is a standard 500GB SATA drive, which although speedy itself at 7200RPM, is there only for bulk storage. The system has an integrated 20GB SLC-based SSD designed for storing the OS and swap files. (See the sidebar for an explanation on SLC vs. MLC.) Because the OS is hosted on a blazing-fast SSD, it means the boot time is measured in seconds. It also means applications launch lightning-fast, but more on that later.

I've heard mixed reviews from folks using the new all-in-one hybrid hard drives that have both SSD and HDD components to increase speed. The concept is brilliant, but because Linux so easily uses multiple devices for different partitions, Polywell's multidrive setup works perfectly. For my test, I put everything but /home on the SSD, and with a few tweaks, it worked well.

Getting Down and Dirty

Once the computer boots up, which with my Xubuntu 12.04 system takes less than 15 seconds, it feels like a computer running from RAM. The menus are snappy, the applications start up immediately, and Internet connectivity, whether via LAN or wireless, works flawlessly.

As far as functionality goes, the computer behaves much like you'd expect once you know its specifications. All hardware was detected properly; everything seems to function correctly, and there were no odd configuration steps required to get full functionality. Although this is largely credited to Linux developers, it's fair to point out the i2303 uses standard-enough hardware that it works with Linux out of the box. It also ships with Linux out of the box,

SLC vs. MLC

SSD drives are becoming quite standard, but few people understand or care about SLC technology vs. MLC. The science behind the two could be an article in itself, but the quick-and-dirty explanation is that SLC can hold a single bit per cell, and MLC can hold two.

The advantage for SLC SSDs is that because each cell holds only a single bit, it is fast and less error-prone. The downside is that the storage density is half that of MLC, so the cost per bit is double. With MLC drives, the advantage is, of course, the storage capacity, but because they require a more complex error-correction mechanism, they tend to be a bit slower.

Keep in mind these are generalizations. Please don't send in letters about brand XYZ's MLC SSD being 50 times faster than brand ABC's SLC SSD. The point is to realize that an SLC SSD was the right choice for the i2303's hybrid design.

which always is appreciated by folks not wishing to pay for a Microsoft Windows license they'll never use.

Getting Tweaky with It

It seems every time I have distinct partitions set up on a computer, months later, I find that I estimated my needs incorrectly. I knew I wanted to install the operating system and swap files on the 20GB SSD, but I did have concerns about running out of space.

I spent a lot of time thinking about how best to partition the two drives, but in the end, I put the root partition on the SSD and the /home partition on the 500GB SATA hard drive. As you can

imagine, my concern was that the /var directory would fill the limited space quickly. I strongly considered putting only the /boot and /usr partitions on the SSD, but I really wanted to see just how fast the computer could be.

Because I installed a handful of games, several of which were quite large, one of the first things I had to do was make a symbolic link to the /opt directory and place it on the hard drive. If I start to have problems with the /var directory, I'll likely do the same. I decided that because Linux is so flexible with symbolic links, I'd be better off putting everything but /home on the SSD and carve off problem areas later. I love Linux.

As the bulk of my games are on the /opt partition, they do load from the hard drive instead of the SSD. It means a couple more seconds to adjust my headphones while I wait for games to load, but it doesn't affect the load time of other applications, nor does it slow down boot time. That said, the 7200RPM SATA drive is no slouch, so even from the hard drive, things load acceptably fast.

The Good

There's a lot to love about the Polywell i2303. Although it certainly would make a wonderful computer for digital signage or HTPC, I think both tasks would be a waste of the unit's abilities. If I had to pick my favorite things, I guess they'd be these:
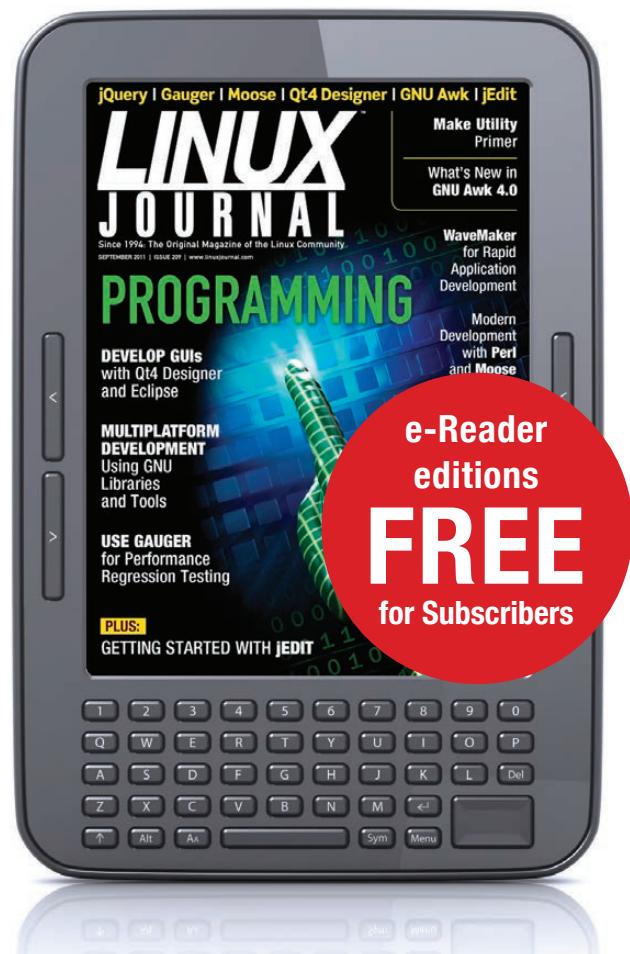
- Dual SSD/HDD combination.

- Above-average graphic performance for such a compact unit.

- Rock-solid construction.

- Impressive feature list.

The last item on my list is easy to overlook. This little unit has a few

This computer is a sturdy metal box and has no frills. I personally think that is a plus, but if you want glowing LED lighting and an aerodynamic design, this computer isn't for you.

features that surprised me. The USB 3.0 port, the dual analog speaker jacks, Bluetooth and detachable antenna are just a few.

The Bad

There were a few minor quirks to the unit. The antenna, while removable, does jut out a bit in the back. I worry if placed in a busy environment it might get broken off. I also noticed whenever I loaded a Web page with Flash on it, the system fan kicked on full blast for a moment. I suspect that is as much the fault of NVIDIA and Adobe as anyone, but it still caught my attention on a regular basis.

The Ugly

Some might be turned off by the absolute stark design of the case. This computer is a sturdy metal box and has no frills. I personally think that is a plus, but if you want glowing LED lighting and an aerodynamic design, this computer isn't for you.

At first glance, the $799 price tag seems steep, but once you realize

how much computer you're getting, I think it's quite reasonable. If you're just hoping to get a cheap nettop computer, this probably isn't the unit for you. If you're looking for an energy-efficient workstation-class device, $799 isn't a bad price.

Conclusion

I won't lie; I love this little computer. If I could use an automotive metaphor, it's a total sleeper. What seems like a piddly little nettop is actually a very functional workstation. I'm a writer, a casual gamer and I occasionally compile software. Even though my demands aren't terribly high, I still prefer a computer that is fast and responsive. This computer ticked every box for me. For more information, check out **http://www.polywell.com/us/oem/i2303.asp**.∎

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.

A Look at ownCloud

Your data under your control—a step toward freedom in the cloud.

AMIT SAHA

ownCloud (currently ownCloud 3, **http://owncloud.org/discover**) helps you store your files on a computer under your control and access them yourself or share with anyone else. Besides storing files, you also can use ownCloud's calendar to keep track of important events and meetings, maintain and sync your contacts using CalDAV and stream your music via its built-in Ampache server. To start using ownCloud, all you need is a basic Web server (Apache, in this article) installation with PHP5. Let's get started.

Installation and Configuration

You want your ownCloud to be truly your own, so you will want ownCloud running straight out of your home computer. Let's see how.

At the time of this writing, you easily can install ownCloud in Ubuntu via a simple `sudo apt-get owncloud`. On Fedora 16, the installation is a little involved, as an ownCloud package is not yet available. You can grab this installation script from **https://gist.github.com/2341191**, which I created so you wouldn't need to type each of the steps, and you can see what is going on in the script by following this blog post entry: **http://echorand.me/2012/02/26/owncloud-boxgrinder-pagekite-owncloud-appliance**. (It also installs MySQL server and configures it; comment the relevant lines if you want to use SQLite with ownCloud.) A key point to note in the script is giving appropriate permissions to the owncloud directory so that the Web server can read/write from/to it.

To check if everything has been set up properly, open your browser and type in the following URL: http://localhost/owncloud. If you see the screen shown in Figure 1, your ownCloud installation has been completed.

As you can see on the screen, you first need to create an admin account for
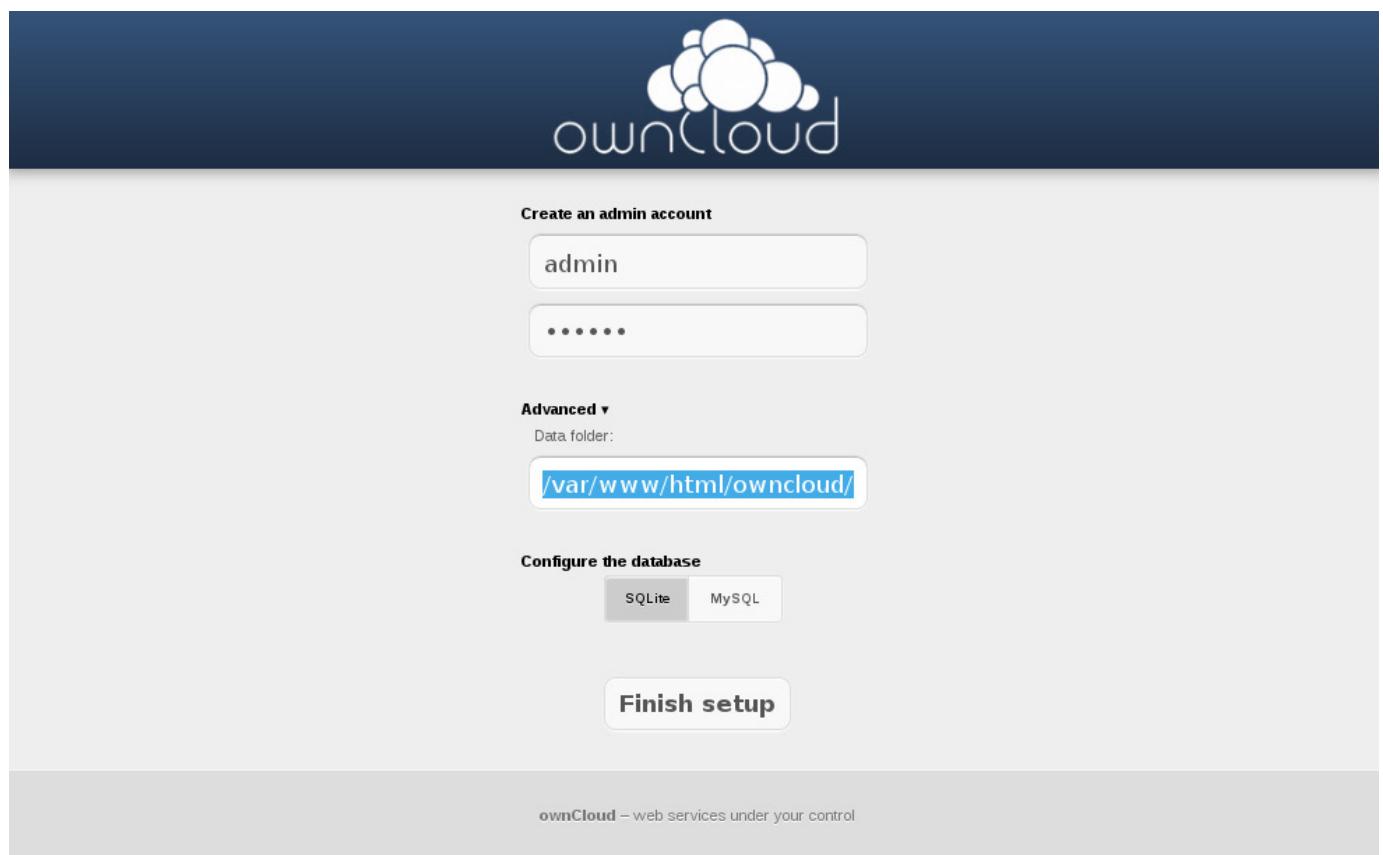
Figure 1. ownCloud: the First Sighting

ownCloud, and specify the directory where files will be stored and the database to use. Let's use the default directory, which is
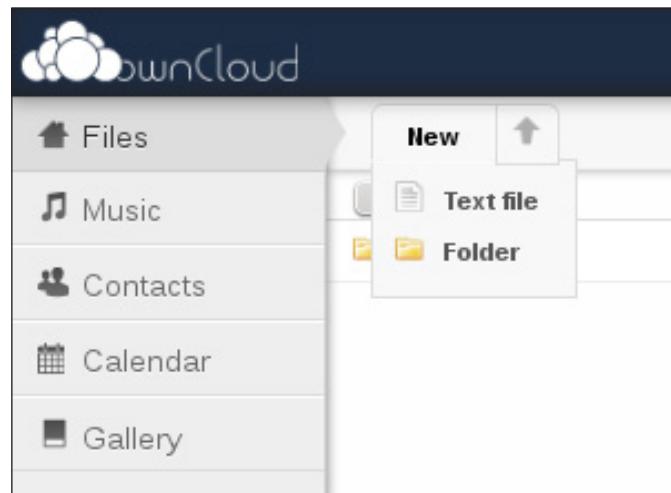
Figure 2. User Home

/var/www/html/owncloud/files, and /var/lib/owncloud/data, in the case of Ubuntu. Let's also use the SQLite database instead of MySQL, but you can choose either one. Once you click the Finish Setup button, the ownCloud home page appears where you will see a sidebar with the following contents: Files, Music, Contacts, Calendar and Gallery.

You can create a new file or a folder on this page from the New tab (Figure 2). By default, there is a maximum size limit (2MB) of files you can upload (which is really a PHP-imposed limit and not an ownCloud one). You can change it to something more desirable, say 50MB, by changing the

Figure 3. Files and Directories
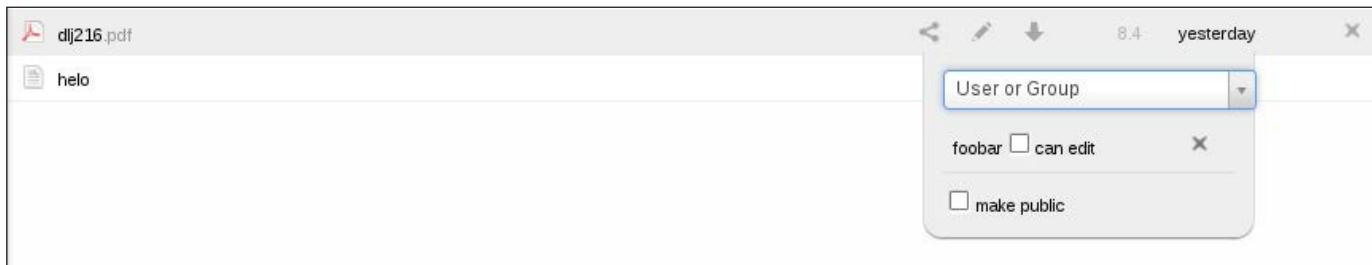
Figure 4. Sharing a File with Other Users Publicly

variables: `upload_max_filesize=50M` and `post_max_size=100M` in the file php.ini (the location of which will depend on your distribution and may be found using `$locate php.ini`; on Fedora 16, it is /etc/php.ini). Restart your Web server, and your file upload limits will be changed.

Now, create a few files and directories. Mine at this point in time look like those shown in Figure 3.

If you have other installation requirements, such as running it off a virtual appliance, refer to **http://owncloud.org/ support/setup-and-installation** for more information.

Exploring ownCloud

One thing you could use ownCloud for is to share your files with others—other users on your ownCloud installation or via a publicly accessible URL. Figure 4 shows how. When you share a file with another user, it shows up in his or her Shared directory. (The Shared directory is special in the sense that you

cannot upload/create files/directories in it. It will show only the ones that have been shared with you by another user.) It seems that you either can have a publicly accessible URL or share it with another user, but not both (that's sensible, if you ask me).

There also is a built-in pdfjs-based PDF viewer, which works quite well. For example, Figure 5 shows a previous issue of *Linux Journal* viewed from my ownCloud installation. Mozilla Firefox had some issues rendering the PDF file, whereas Google
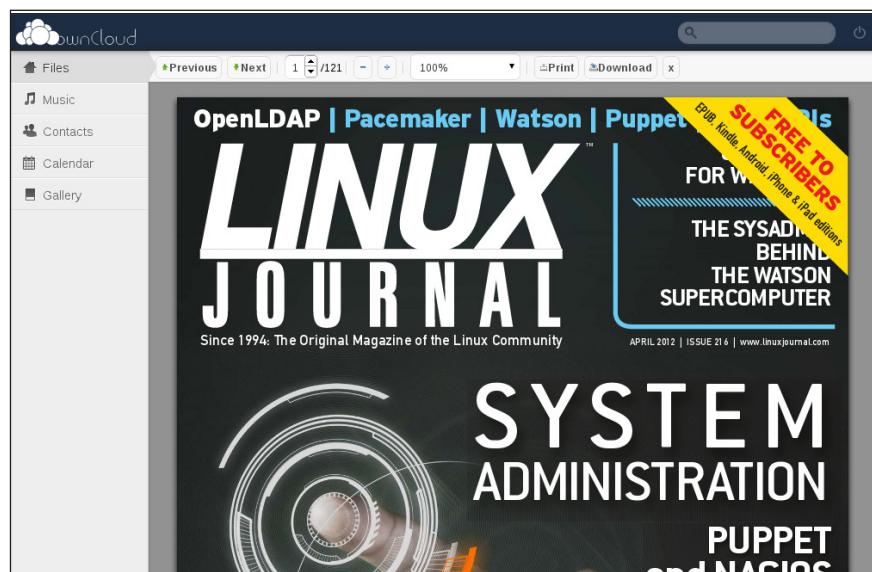
Figure 5. PDF Viewer

Chrome worked perfectly (bug report: **http://bugs.owncloud.org/ thebuggenie/owncloud/issues/oc-442**).

The Music section scans your directories/files and displays them (Figure 6). You can create a playlist from the files and play it in your browser, and use an Ampache client to play them either on your local machine, a remote computer or your Android phone (the Ampache.net client worked perfectly for me). Quickplay (**http://quickplay.ampache.org**) is an

easy-to-use program for playing music from an Ampache server (Figure 7).

The Ampache address is http://localhost/ owncloud/apps/media/, and the user name is your ownCloud user name, and the password/API key is your ownCloud password.

The Contacts and Calendar sections can be used to create, view and maintain contacts and calendars (Figures 8 and 9). As you can see, I have friends in high places and need to be reminded of a shutdown. You can have multiple
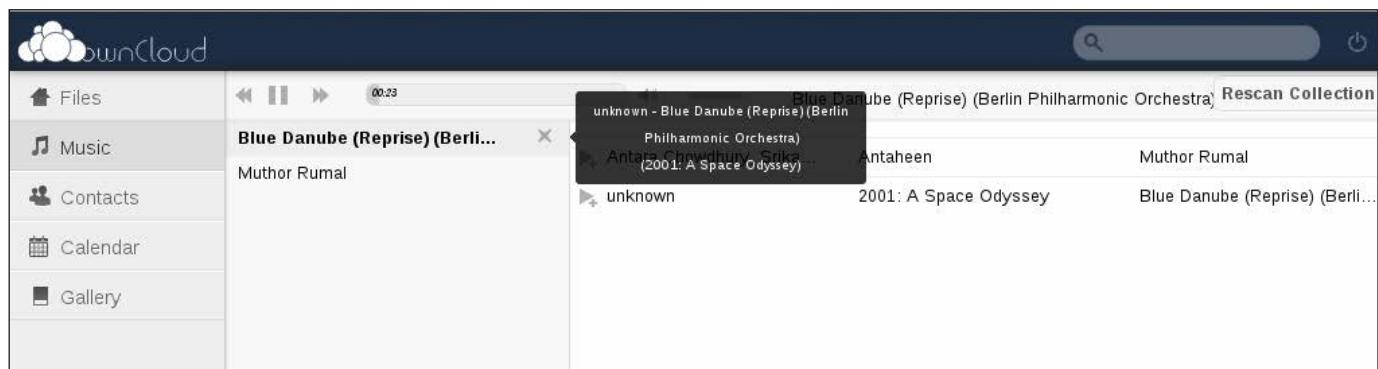
Figure 6. Music

Figure 7. Quickplay
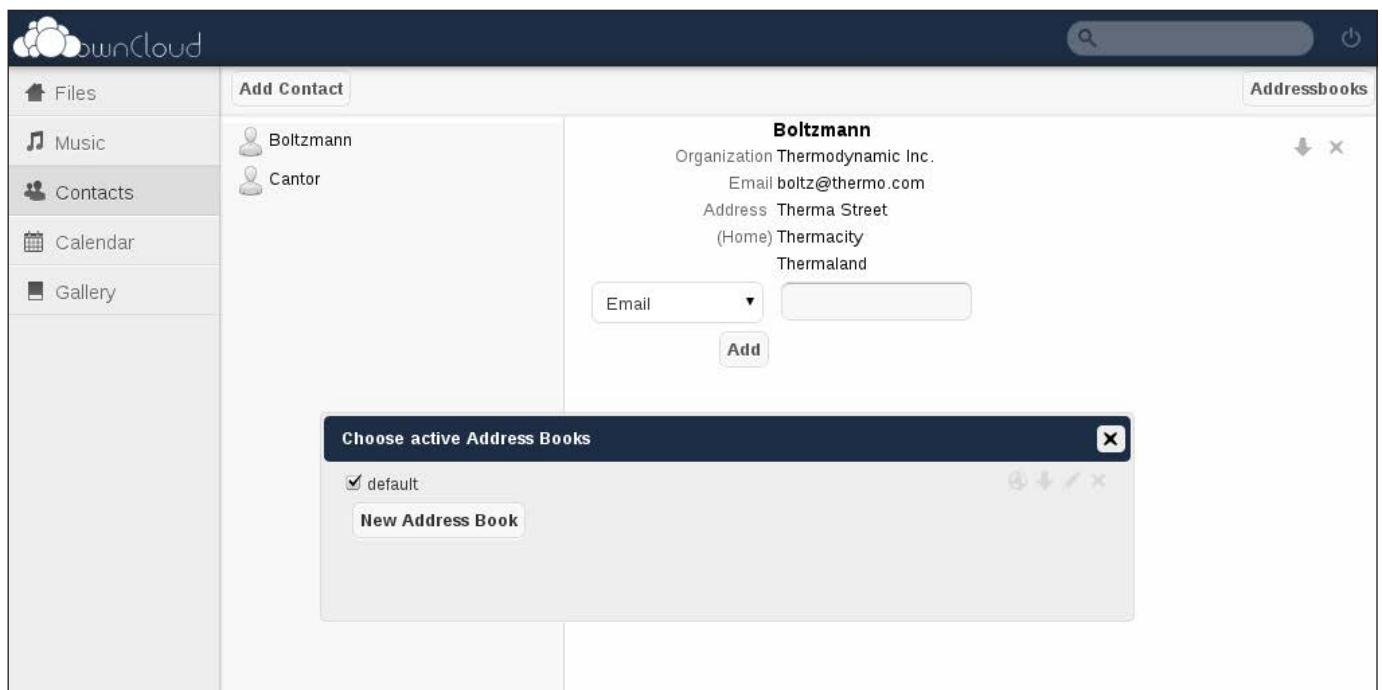
Figure 8. Contacts
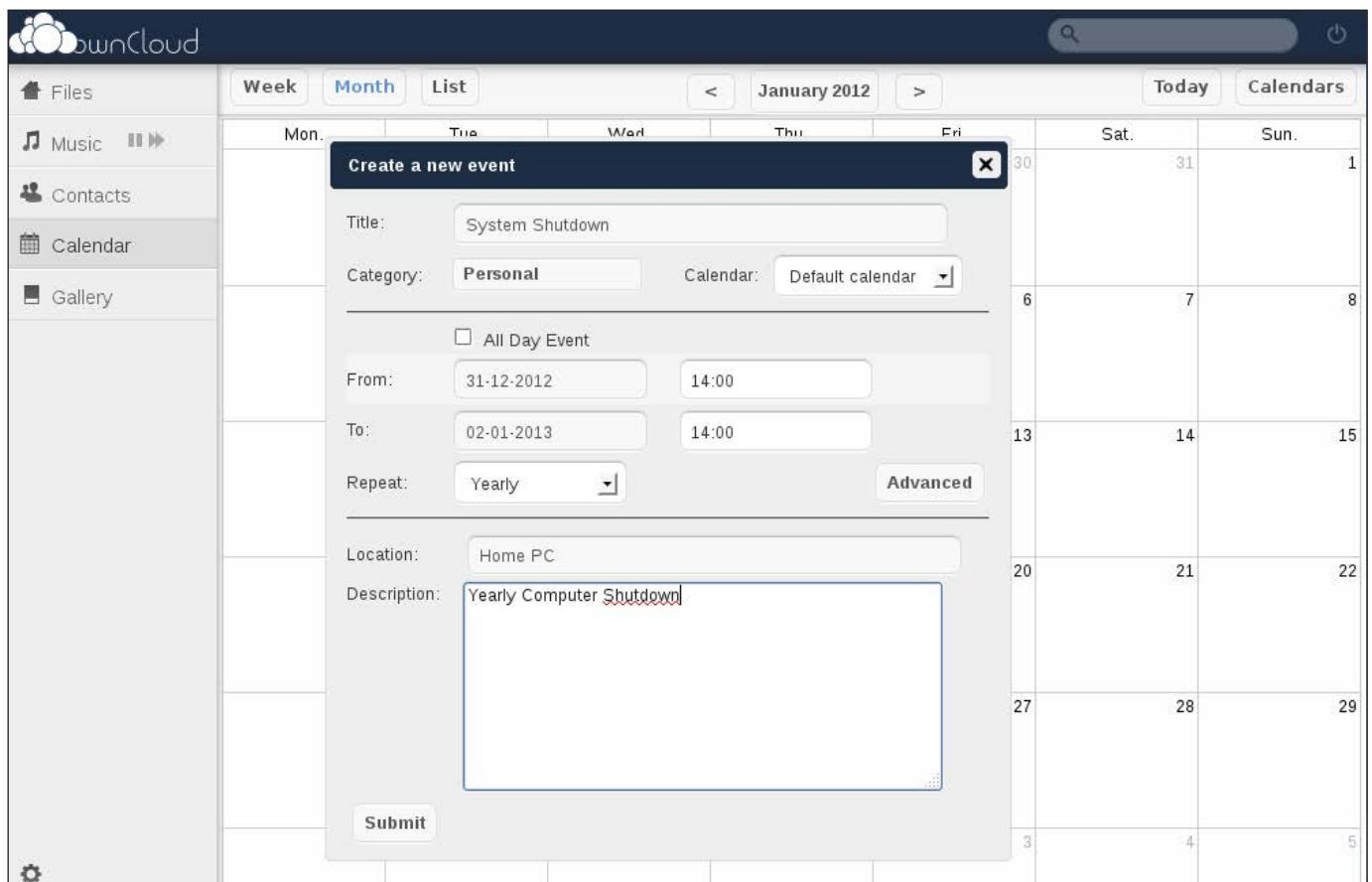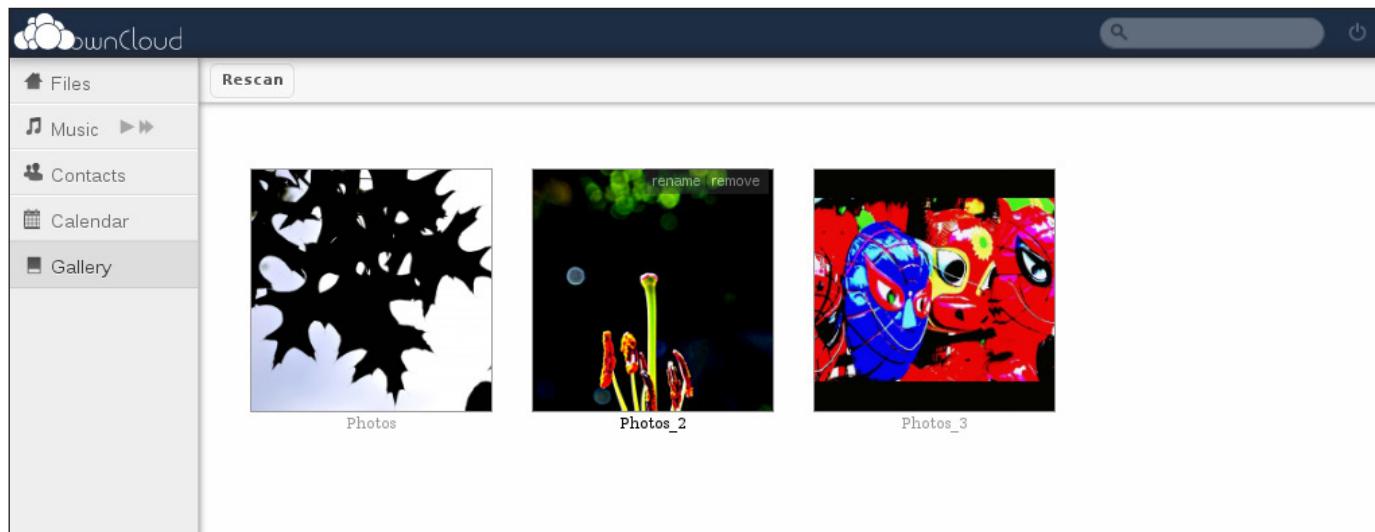
Figure 9. Calendar

Figure 10. Gallery
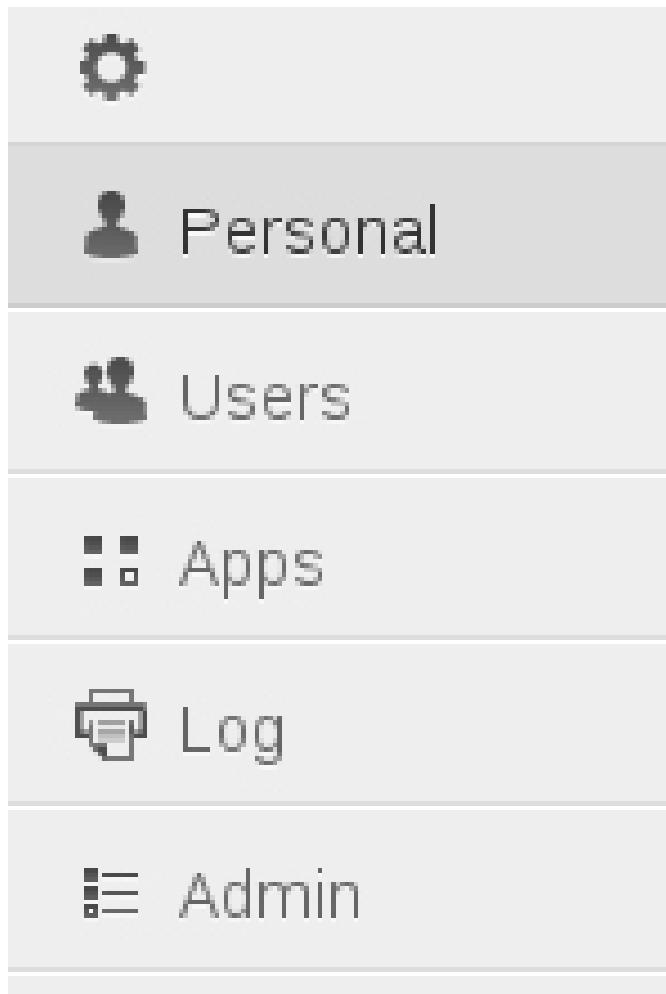
address books and multiple calendars, download them and share them in standard formats and protocols.
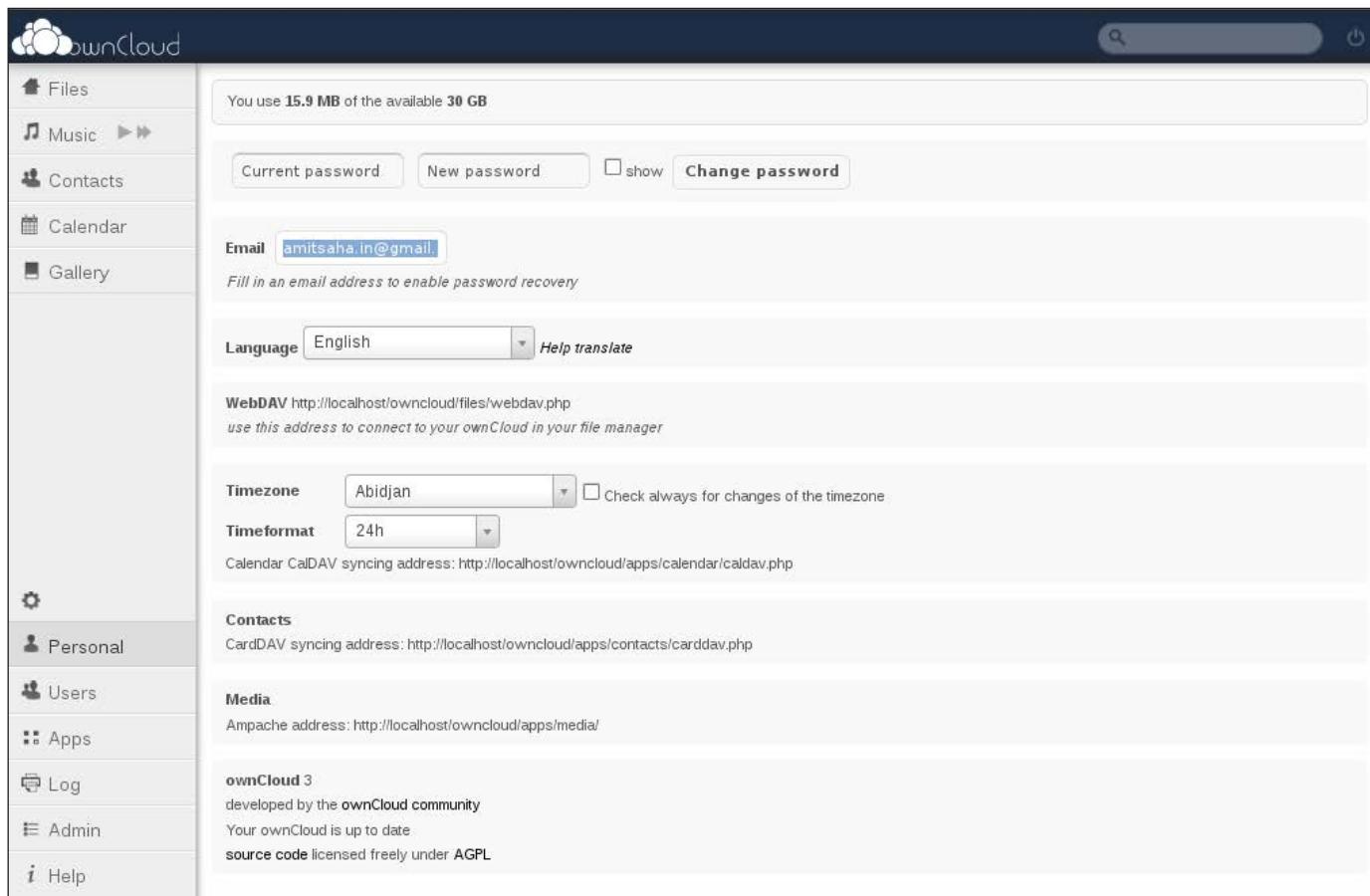
If you have uploaded any image files in any of your directories, the Gallery section will show them. It's not my intention to show off my photography skills, but Figure 10 shows some of the photographs I've uploaded into my directories.

That pretty much sums up the basic usage scenarios for ownCloud. Now, let's go a little deeper.

ownCloud Settings and Administration

The settings section toward the bottom left of your ownCloud home page has four major sections: Personal, Users, Apps, Log and Admin (Figure 11). Note that you will see all these sections only when you are the admin user. Other users see only the

Figure 11. Settings

Figure 12. Settings: Personal

Personal section.

In the Personal section, you can set the e-mail address for password recovery, your language, timezone and also view the WebDAV, CardDAV, CalDAV and Ampache addresses (Figure 12).

You can use WebDAV to view your files without logging in to the ownCloud installation by using the URL shown to you using browser-based authentication. You also could hook up your operating system's file manager to create an entry pointing to your ownCloud files (**http://owncloud.org/support/webdav**).

If you are looking for a command-line way to access/retrieve your files, Cadaver (**http://www.webdav.org/cadaver**) is a command-line WebDAV client and can be used to store and retrieve your files from ownCloud. It can be installed using either apt-get or yum. Once you have it installed:

```
$cadaver http://localhost/owncloud/files/webdav.php
```

```
Authentication required for ownCloud on server `localhost':
Username: admin
Password:
dav:/owncloud/files/webdav.php/>
```

The interface to Cadaver is similar to that provided by an FTP client, with commands like put for uploading files and get for downloading files.
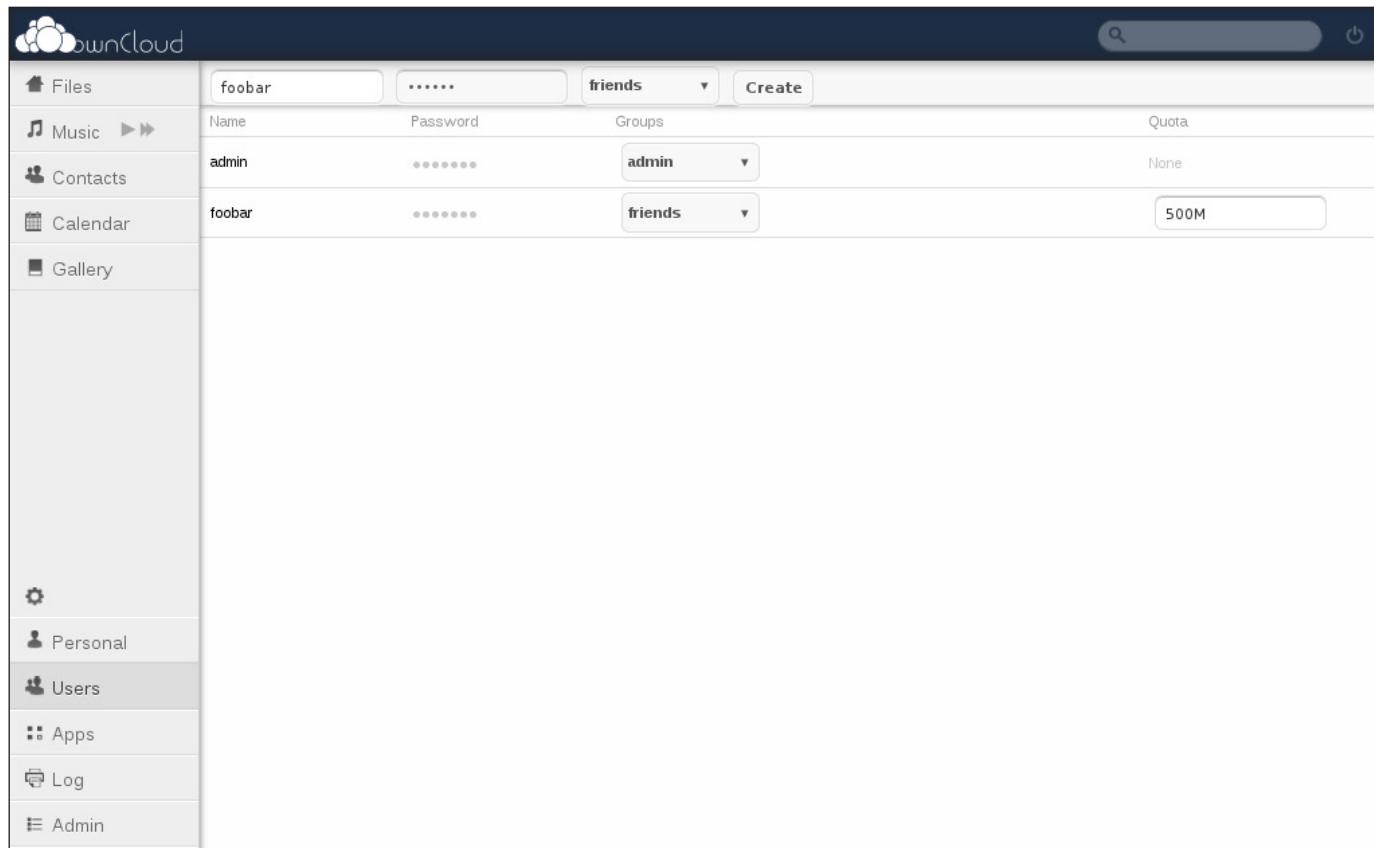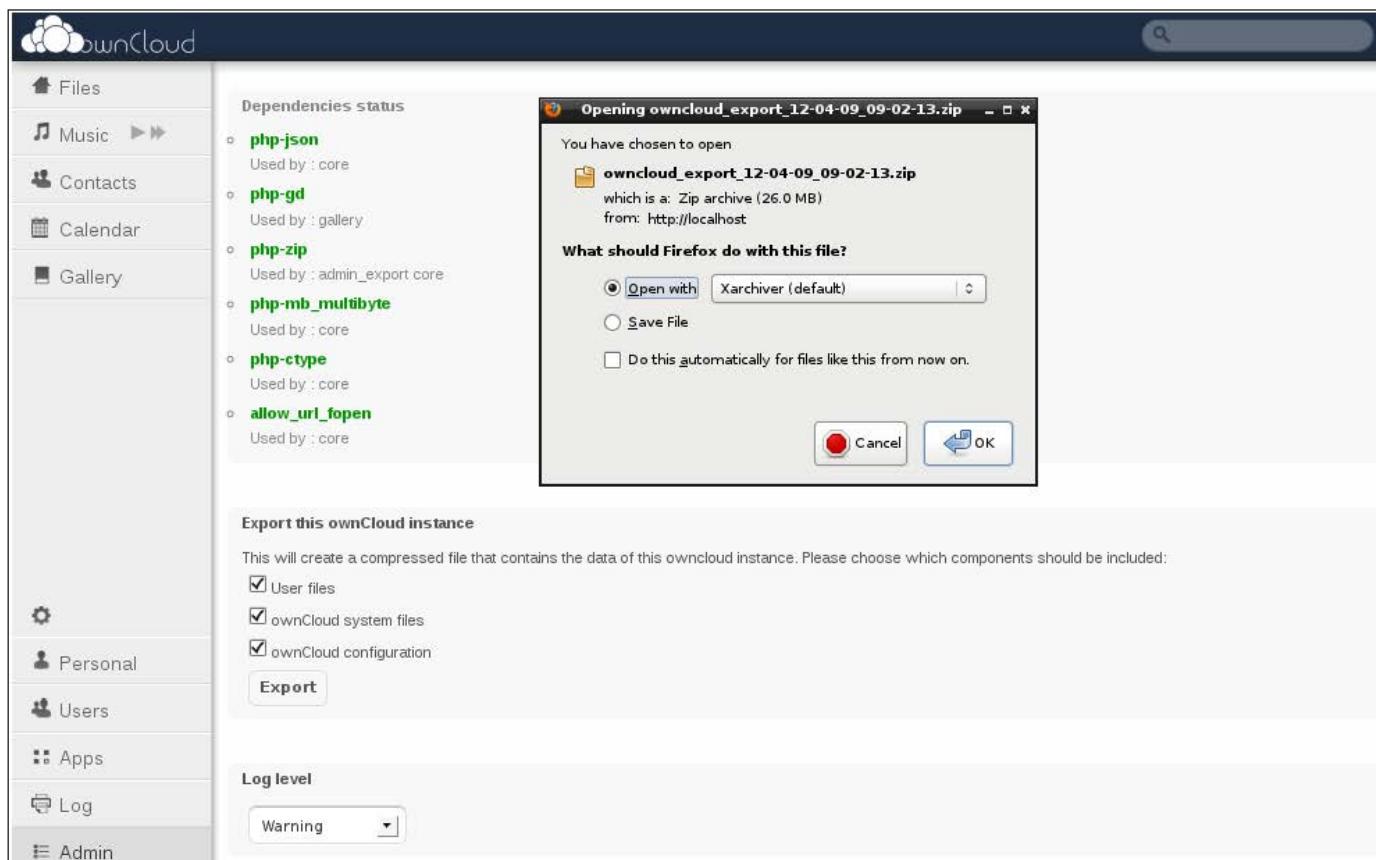
Figure 13. Settings: User

The interface to Cadaver is similar to that provided by an FTP client, with commands like put for uploading files and get for downloading files. (I faced problems with getting Cadaver to work on my Fedora server installation, although on my Debian and Ubuntu installations it worked great.)

In the Users section, you can add new users and groups and assign them disk usage quotas (Figure 13). This is useful when your ownCloud installation will be used by someone other than yourself.

Use the Log section to see information if something isn't happening the way you would expect. The Admin section doesn't have much in it. The only thing you can do here is export your ownCloud installation: user files, system files and

Figure 14. Settings: Export

configuration as a .zip file (Figure 14). This zip file backs up the ownCloud directory, and you then can use it to replicate or replace your ownCloud installation elsewhere.

ownCloud apps are a way to extend ownCloud's functionality. In fact, the Calendar, Contacts, Gallery and Media server are all implemented as ownCloud apps. In the Apps section under Settings, you can see the currently installed apps and others available for install. More applications are available at **http://apps.owncloud.com**. However, a number of them seem quite rough and unstable at this time.

Accessing Your ownCloud and a Step toward Freedom in the Cloud

On the face of it, ownCloud is just an open-source solution allowing you to store your own files where you please. With a little thought, it becomes apparent that it helps you take one step ahead in the fight not to give up your data to services where it may not be very safe. However, to become really useful, you should be able to access your data from places outside your home/office/college network. This is where services like PageKite (**http://pagekite.net**) make the goal of having your own storage service

running off your desktop computer or laptop a practical reality.

If you have used the installation script for Fedora, you already will have PageKite installed. Otherwise, follow the one-line installation process: `curl -s https://pagekite.net/pk/ |sudo bash`.

Once that is done, set up PageKite: `pagekit.py --s signup`, and follow the steps. Once the signup is done, set up your kite using `$pagekit.py 80 <yourname>.pagekite.me` (assuming you are running Apache on port 80). Your ownCloud service now will be accessible at http://<yourname>.pagekite.me.

Ending Notes

ownCloud is a good experience and provides basic functionalities flawlessly. However, there still are some rough edges (**http://bugs.owncloud.org/thebuggenie/owncloud/issues/open**), which gradually should be fixed as ownCloud starts welcoming enterprise customers (**http://owncloud.com**). Besides, if there isn't something you want, you can venture into writing your own app: **http://owncloud.org/dev/writing-apps**. An official desktop app for ownCloud is not fully there yet, except for the Mirall Project for OpenSUSE (**http://en.opensuse.org/Mirall**), and a basic Android client is available from **http://owncloud.org/support/android**. It's functional, but nothing fancy.

Have fun building your own cloud storage from your desktop for yourself,

ownCloud 4

As we go to press, ownCloud 4 was released, boasting a number of new features, such as the versioning app that adds basic version-control-like features for your files, the encryption app to store your files in encrypted form and the Drag and Drop Upload facility. Installation is basically the same as with ownCloud 3, and you can use a slight modification of the shell script provided with this article to try it out (on Fedora). The modified shell script is available at **https://gist.github.com/2770050**. When you start your ownCloud instance and run into error messages, such as "Cannot modify header information", simply go to the relevant PHP file mentioned in the error message and look for an extra new line at the end and remove it for an error-free installation. In my case (on Fedora 16), it was the file /var/www/html/owncloud/apps/files_odfviewer/appinfo/app.php (I haven't yet investigated the exact cause of the error).

your friends and your colleagues.

Acknowledgements

Credit is due to the ownCloud folks for an interesting project, to Bjarni of Pagekite.net and the awesome Shutter tool (**http://shutter-project.org**) for helping me create the screenshots for this article.■

Amit Saha is currently a PhD research student in the area of Evolutionary Algorithms and Optimization. Like his random echoes show (http://echorand.me), he has been writing on myriad Linux and open–source technologies for the past few years. He welcomes comments on this article and beyond at amitsaha.in@gmail.com.

Resources

ownCloud Project: **http://owncloud.org**

Get Started: **http://owncloud.org/get-started**

ownCloud Appliances: **http://owncloud.org/support/setup-and-installation/appliance**

ownCloud Forum: **http://forum.owncloud.org**

ownCloud Developer Center: **http://owncloud.org/dev**

Interesting Forum Discussion—ownCloud on Pogoplut: **http://forum.owncloud.org/viewtopic.php?f=3&t=2101**

PageKite: **http://pagekite.net**

INTERFACING
DISPARATE
SYSTEMS

DON'T LET THE LACK OF A KNOWN INTEGRATION METHOD KEEP YOU FROM ACHIEVING YOUR GOALS. IF YOU HAVE TO, BUILD IT YOURSELF!

JAMES LITTON

When hearing the word interface, most people probably think of a Graphical User Interface or a physical hardware interface (serial, USB). If you dabble in scripting or are a serious developer, you, no doubt, are familiar with the concept of software interfaces as well. Occasionally, the need arises to integrate disparate systems where an interface doesn't already exist, but with a little ingenuity, an interface can be created to bridge the disparity and help you meet your specific needs.

I have an extensive home automation implementation I developed over the years. As I knocked out the "easy" integrations, I eventually came to a point of wanting to integrate systems that are not home automation-friendly. An example of this is my alarm system. Excellent alarm panels exist on the market that make integration a cinch, but I already had a fully functional alarm system and was determined to integrate it into my home automation setup rather than replace it.

My first inclination was to hack a

me first lay out my integration goals. Although it would be nice to capture sensor data from the alarm system, in my case, it was totally unnecessary as the only data that might be helpful was motion sensor data or specific zone faults. Because I already have numerous motion sensors installed that are native to my home automation build, and because fault data wasn't a factor in my immediate integration requirements, I concluded that I needed to know only if my alarm was "armed" or

EXCELLENT ALARM PANELS EXIST ON THE MARKET THAT MAKE INTEGRATION A CINCH, BUT I ALREADY HAD A FULLY FUNCTIONAL ALARM SYSTEM AND WAS DETERMINED TO INTEGRATE IT INTO MY HOME AUTOMATION SETUP RATHER THAN REPLACE IT.

keypad or build my own hardware interface that would allow me to capture status information. Both of those approaches are viable, but as I thought about other options, I realized I could integrate my proprietary alarm system into my home automation system without even cracking open the alarm panel.

Before I reveal the details of how I achieved the outcome I wanted, let

"unarmed". Knowing the state of the alarm system helps me make my home automation system smarter. An example of this added intelligence might be to change the thermostat setting and turn off all lights if the alarm state changes to armed. Another example might be to turn on all of the lights in the house when the garage door opens after dark and the alarm is armed.

As I thought through the scenarios a bit further, I quickly realized I needed a bit more data. Depending on how an alarm system is installed and the practices of its users, there may or may not be multiple armed states that need to be considered. In my case, I have two separate armed states. One state is "armed away" (nobody home) and the other is "armed stay" (people are in the house). It wouldn't make sense to turn off all of the lights in the house, for example, if the system was set to armed stay, but that would make perfect sense if it were set to armed away. As I continued to think through my needs, I concluded that knowing whether the system was armed away, armed stay or unarmed was all I needed to add significantly greater intelligence to my home automation scenes.

Once I had a firm grasp of my needs, I realized my alarm-monitoring company already was providing me with some of the data I was looking for in the form of e-mail messages. Every time the alarm was armed or disarmed, I would get an e-mail message indicating the state change. I had been using this feature for a while, as it was helpful to know when my kids arrived home or when they left for school in the morning. Because I had been using this notification mechanism for some time, I also knew it to be extremely timely and reliable.

Because I was getting most of the data I needed, I started thinking about ways I might be able to leverage my e-mail system as the basis for an interface to my proprietary alarm panel. In days gone by I had used procmail to process incoming e-mail, so I knew it would be fairly easy to inject a script into the inbound mail-processing process to scan content and take action.

Before I started down the path of writing a script and figuring out how to make my e-mail system run inbound mail through it, I needed to deal with the shortcoming I had with status notifications. You may have noticed that I said my alarm monitoring company was sending me two status notifications: one for armed and one for unarmed. I was fairly certain that an additional relay could be configured so the folks at the company could notify me with the two variations of "armed" that I needed to proceed, so I called them to discuss the matter, and sure enough, they were able to make the change I requested. In fairly short order, I was receiving the three notifications that I wanted.

With the notifications in place, I could start the task of creating a script to scan incoming mail.

To keep things as simple as possible, I decided to write the script in Bash.

To follow this example, the first thing you need to do is capture all of the data being piped into the script and save it for processing:

```
#!/bin/bash
while read a
do
  echo "$a" >>~/tmp/results.tmp
done
```

This block of code redirects inbound e-mail messages to a temporary file that you now can perform search operations against. Because e-mail messages are simple text files, there are ample methods you can leverage to search for the strings that tell you the state of the alarm system. In this case, you could receive three possible messages which are "Your alarm has been Armed Stay", "Your alarm has been Armed Away" or "Your alarm has been Disarmed".

Now that you know exactly what you are looking for, use grep to perform the search operations. When you combine your grep searches with if statements, you can execute a specific command when one of your searches evaluates to true.

The heart of my home automation system is a software package that has an extensive REST API that I can leverage to change device states, set variables, control access groups and control device links. This makes it extremely easy to set a variable for the alarm state that I then can use to trigger various actions and control scenes in my home. To interact with the REST API, let's use curl.

In my case, my home automation software expects data to be sent as a PUT instead of curl's default of GET. To accomplish this, let's use the -X parameter to tell curl to use PUT. Identify the data you want to send to the server with the -d parameter followed by the data that you need to send to the server. Finally, you need to tell curl what URL to connect to:

```
url="http://ha.example.com/vars/alarmstate"
if grep -q 'Armed Stay' ~/tmp/results.tmp; then
  curl -X PUT -d value="ARMED Stay" $url
elif grep -q 'Armed Away' ~/tmp/results.tmp; then
  curl -X PUT -d value="ARMED Away" $url
```

IN ORDER FOR THIS INTERFACE TO WORK CORRECTLY AND CONSISTENTLY, IT IS IMPERATIVE THAT YOU CLEAN UP AFTER EACH EXECUTION OF THE SCRIPT.

```
elif grep -q 'Disarmed' ~/tmp/results.tmp; then
  curl -X PUT -d value=DISARMED $url
fi
```

When you put all of this together, the result is a block of code similar to this that will scan your file for the three string possibilities that will tell you the current alarm state. Because the evaluation is set up as an else/if, the if evaluation block will terminate when one of the expressions evaluates to true.

In order for this interface to work correctly and consistently, it is imperative that you clean up after each

Now that you have a script, it needs to be injected into the e-mail processing process in such a way that inbound mail is forced through the script. How this is accomplished will vary from system to system, so I won't go into great detail here, but I have implemented this very easily on both an open-source edition implementation of Zimbra and the e-mail platform provided by a very large and well-known US hosting provider. My current implementation resides with a hosting provider where I have an e-mail account called

AFTER HAVING MY INTERFACE SCRIPT IN PLACE FOR A NUMBER OF MONTHS, I HAD THE OPPORTUNITY TO USE THIS TROUBLESHOOTING METHOD.

execution of the script. You may have noticed in the first block of code that data is appended to the temporary file using the >> I/O redirection operator. You append data, because the data streams into the file one line at a time. If you failed to use the append operator, the resulting file would contain only the last line of data from your message. Fortunately, cleaning up after ourselves is as easy as deleting our temporary file:

```
rm ~/tmp/results.tmp
```

ha@example.com. I configured this account to "forward" all inbound mail to my script and then throw the message away.

With everything now in place, I configured my monitoring service to send alarm state change messages to my ha@example.com address and to my personal e-mail address. Having the messages go to both locations is helpful if you need to troubleshoot. Testing whether messages are coming from the monitoring company is as easy as checking my personal e-mail

to see if the state change message is present. If the message is present in my personal mail, but state change data isn't flowing to my home automation system, I then know that I need to troubleshoot my connectivity or my interface script itself.

After having my interface script in place for a number of months, I had the opportunity to use this troubleshooting method. I noticed that I was getting e-mail notifications in my e-mail, but state changes were not flowing to my home automation system. In order to verify that data was flowing to my script and that the data was being saved to my temporary file, I commented out the line that deletes the temporary file, then I forced an alarm state change. Sure enough, the file was created, but no change to the variable in my home automation system. In order to rule out a connectivity or a firewall issue, I then ran the curl command from the command line manually to see if my REST API call could reach my home automation system and change the variable. That worked fine. This conclusively proved that there was an issue with my interface script. At this point, I inspected the contents of the temporary file more closely, and I saw a new line in the headers that said "Content-Transfer-Encoding: base64". Apparently my monitoring company had made some changes to its e-mail

system that I needed to account for. To do this, I would need to add a new block of code to see if the content of the newly arrived e-mail message was base64-encoded. If you find the message is encoded, use Perl and the the decode_base64 function in the MIME::Base64 module to decode the message:

```
if grep -q 'Content-Transfer-Encoding: base64' ~/tmp/results.tmp;
then perl -MMIME::Base64 -ne 'print decode_base64($_)'
<~/tmp/results.tmp>~/tmp/results2.tmp
  rm ~/tmp/results.tmp
  mv ~/tmp/results2.tmp ~/tmp/results.tmp
fi
```

Adding this block of code just before the block of code that performs the grep evaluations fixed the problem, and I was off to the races once again.

I have had this homebrew script interface in place for more than a year now. Other than the encoding issue, the interface's performance has been absolutely rock-solid. This allowed me to achieve the integration I wanted, even if that integration is a bit of a loose integration. Not only did this allow me to use the system that I already had, but also the implementation was completely free, and after all, who doesn't like *free*?∎

James Litton has held executive-level positions at numerous organizations and is a partner and CEO of Identity Automation.

VIDEO ART

Experimental animation and video techniques in Linux.

REBECCA "RUJI" CHAPNIK

Animation and video editing in Linux can be treacherous territory. Anyone who has tried working in these media probably has experienced the frustration of rendering a huge file for an hour only to see the program crash before the export is finished. A bevy of tools and applications for manipulating video exist for Linux, and some are more mature than others.

The most mainstream of GUI applications have been covered quite a bit in other Linux-related articles on the Web and in print, including in previous issues of *Linux Journal*. Some of these names may ring familiar to you: Kino, PiTiVi, Openshot, Cinelerra, Kdenlive and Open Movie Editor.

Although I refer to these nonlinear editors (NLEs) from time to time here, the main purpose of this article is to introduce some video effects and techniques you may not have thought of before. If you are producing a film or animation in a conventional format, such as a DVD or a Web video, you most likely will want to employ a suitable NLE at some point in your process. Many of the ideas I present in this article are experimental.

Video Editing

LiVES LiVES is primarily a VJ (video jockey) tool for performing live audio-visual effects, but it also can encode and export video via its MPlayer back end. The interface has two modes: clip editor and multitrack editor. The clip editor view is more suitable for live VJ sets, while you'll probably lean toward the multitrack view if using LiVES as your NLE.

LiVES is highly extensible. In addition to the built-in effects, you can apply custom RFX (rendered/real-time effects) plugins. Several of these scripts are available for download from the LiVES Web site. You also can share LiVES' real-time effects with other applications using the frei0r effects API.

The number of options and the advanced effects in LiVES are comparable to those of Cinelerra, but I strongly recommend LiVES over the latter. Cinelerra is indeed a powerful video editor, but the interface is antiquated and difficult to use. Although LiVES can seem foreign to new users, it is not hard to become acquainted with it.

ZS4 ZS4, formerly known as Zweistein, is a unique—and quite strange—video editor and compositor. The developers of ZS4, who go by the name "t@b", are a duo of musicians who use their own software to create music videos. They are hard to pinpoint on the Web, as they use several sites for different purposes.

I admit that I was confused by the existence of both zs4.net and zs4.org,
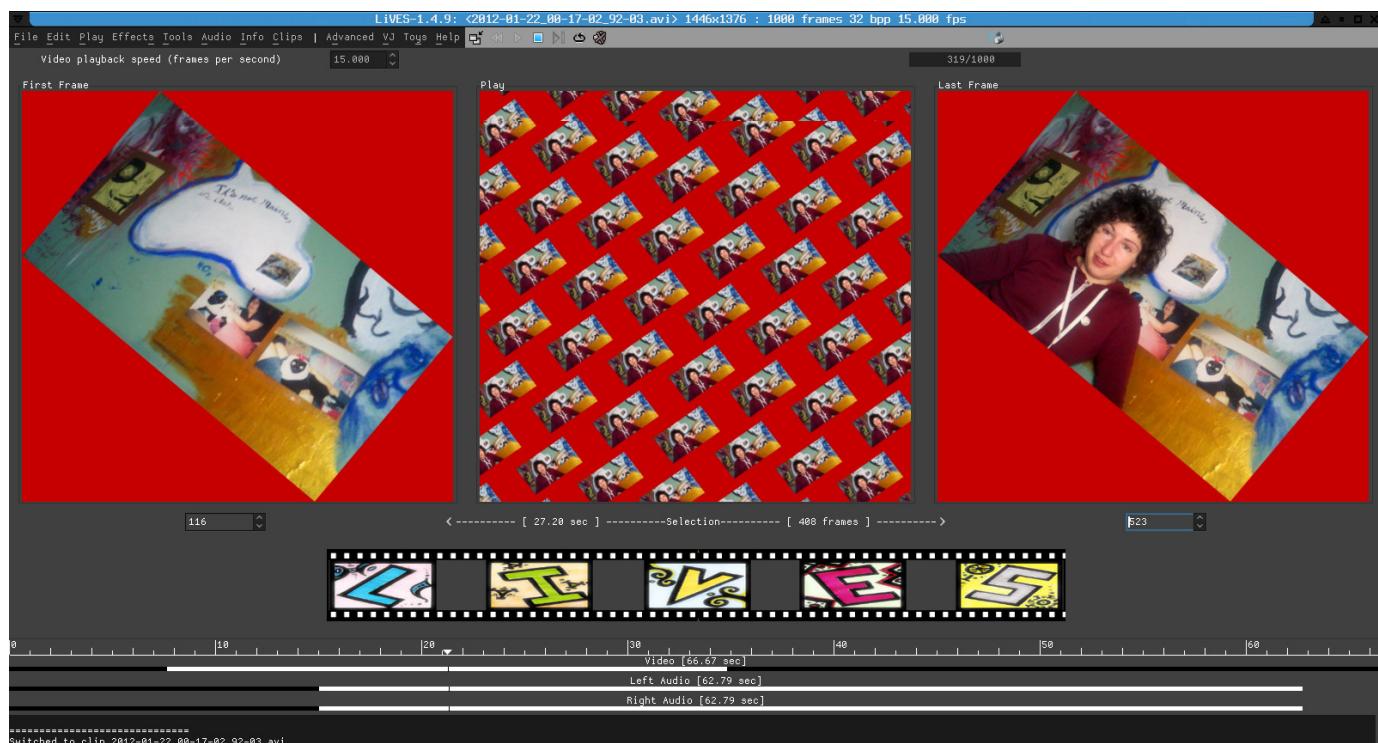
Figure 1. LiVES in the Clip Editor View
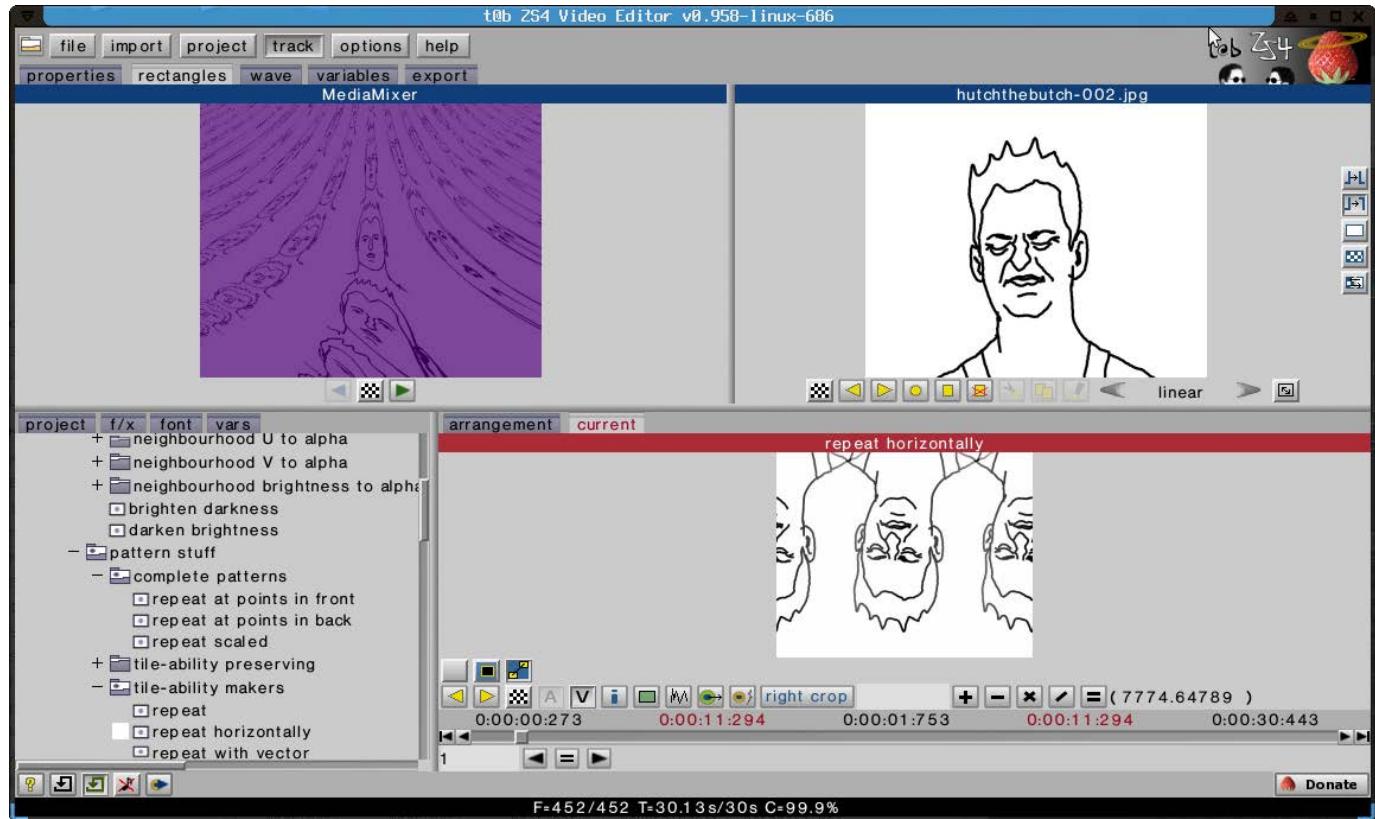
Figure 2. Tiling Effects in ZS4

as well as the Viagra advertisement lines that appeared in Google search results at the zs4.net domain. The two sites both contain download links for ZS4 as well as some other software.
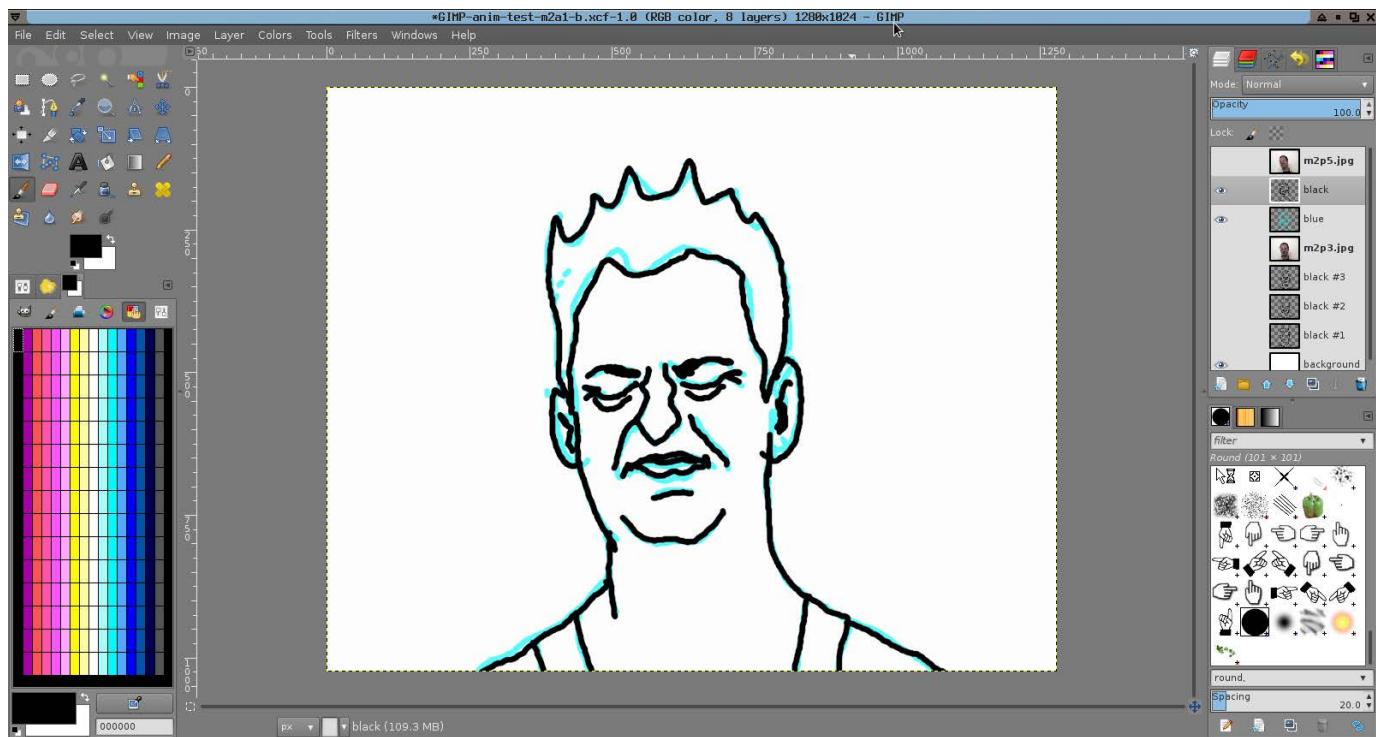
If you plan to use ZS4, I recommend downloading the t@b Media Converter and/or installing Avidemux, as ZS4 is picky about importing video files. Most videos are not compatible out of the box, so it may be necessary to convert them to a format ZS4 can work with.

Working with ZS4 can be frustrating at first because the interface is far from intuitive. Menus are not where you would expect them to be, and you might find yourself aimlessly clicking your cursor in different places to accomplish a simple task, such as dragging a media file into the timeline. The media viewing windows are vaguely labeled "rectangles". To show or hide a track, you click on + or - instead of the typical open- or closed-eye icon.

It took me years to gather the patience to learn my way around this program. So yes, the GUI needs some serious work if it ever is going to reach a mass audience, but it doesn't seem like mainstream appeal is a major concern for the eccentric-minded developers.

So, why tell you about a bizarre-looking application that hasn't been updated in years when there are plenty

Figure 3. An animation in progress that I made by tracing reference photos of faces from the Psychological Image Collection at Stirling (PICS).

of other video editors for Linux? Well, for all ZS4's graphical quirks, it can accomplish some very interesting compositing effects.

Animation

GIMP The famous GNU Image Manipulation Program can create animations as well as still images. Because it is a full-featured image editing program, you can use it to create an animation entirely from scratch.

In order to import a prepared image sequence into GIMP, click File→Open as Layers... or press Ctrl-Alt-o. The Open Image dialog allows you to select multiple files, which then will appear as layers.

In the example shown in Figure 3, I imported a series of reference photos into GIMP and traced over them in cyan and then in black. I eventually deleted the reference photos and blue layers, leaving only the black-lined drawings that I planned to use for my final animation.

To finish my animation, I exported the layers as a GIF and specified animation parameters in the export dialog. Because I wanted to use the animation in a video, I had to turn the animated GIF into a video file. I ultimately chose to do this by way of screen recording, but that is not the only option.

From Stills to Movies

Let's say you have a sequence of images,

Stopmotion is simple and to the point, with a nice drag-and-drop interface.

///

or perhaps an animated GIF, that you want to make into a video file. There are several ways to go about this.

Stopmotion Stopmotion started as a student project under the Skolelinux/Debian-edu organization in 2005. Although it hasn't been updated since 2008, I find it to be a handy tool for anyone working with frame-by-frame animation. You might have trouble finding Stopmotion in your distribution's repositories if you aren't using a DEB- or RPM-based package manager, but you can, of
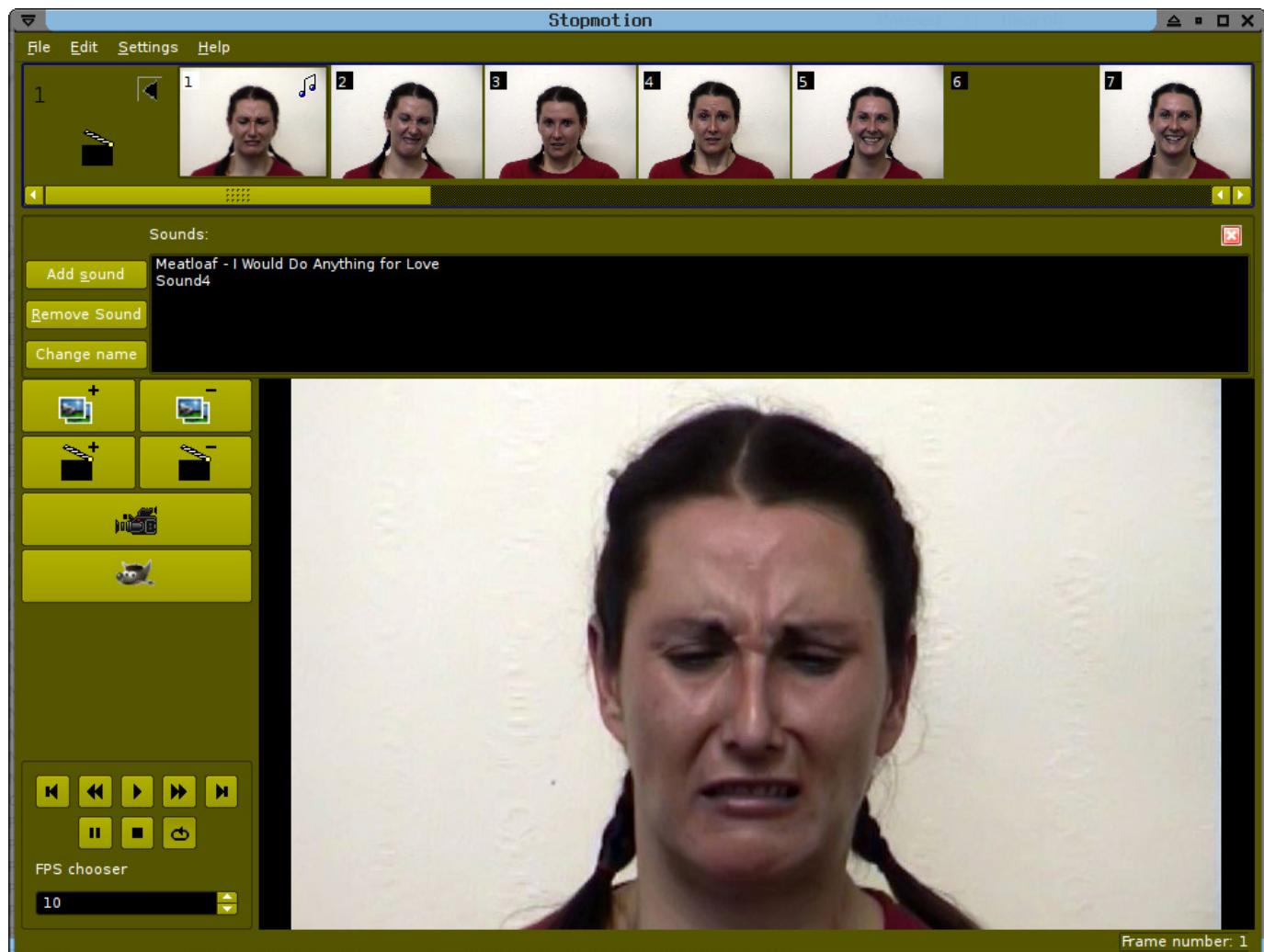
Figure 4. Animating a Sequence of Faces in Stopmotion

course, compile it from source on any distribution; that's how I set it up in Sabayon Linux.

Stopmotion is simple and to the point, with a nice drag-and-drop interface. It's not designed for heavy post-production or for drawing and adding effects to frames. Rather, the point is to give users an easy way to arrange images sequentially and export them into a video file.

The video import and export options are limited only by your imagination (and your knowledge of the command line). If you know how to use FFmpeg and/or MEncoder to convert image sequences to video, you can pass your desired command-line arguments to Stopmotion, which is essentially a GUI for those programs. Stopmotion also gives you several choices of video capture commands for grabbing video from your Webcam or another attached device.

One cool feature I didn't know about until I read the user's handbook was the option to add sound. You can set a sound clip to start at any given frame by double-clicking on it. The audio I added to my sequence didn't play in the exported AVI, but maybe you'll have better luck.

If you want to perform more-advanced editing on your individual frames, Stopmotion has a button to open a selected frame in GIMP. You also can export your data into Cinelerra for video editing.

Command Line

There are several ways to turn frames into motion via the command line.

jpegtoavi jpegtoavi is a simple C script that does exactly what its name suggests—converts a sequence of *.jpg files into an AVI movie. If your images are not in the JPEG format, you first can convert them using the `convert` command from ImageMagick:

```
convert image.png image.jpg
```

If you need to convert a batch of images in a folder, ImageMagick grants you about a gazillion different methods. One of these is to `cd` to that directory and do:

```
convert *.png image.jpg
```

The new filenames will be numbered automatically.

Once you have your folder of sequenced JPEG files, you can employ jpegtoavi. A basic usage template from the man page is:

```
jpegtoavi -f fps width height img1 [img2 ... imgN]
```

Although jpegtoavi is nice for simple tasks, minimal documentation exists. I was surprised to find that none of the main Web sites hosting downloads of the software provided any type of wiki or instructions beyond what already was in the man page and README.

You can do more-advanced encoding with FFmpeg and MEncoder, both of which are heavily documented on-line and in their man pages. These programs both rely on libavcodec and have many overlapping uses, but the command formats are different. For this article, I cover only FFmpeg.

This will convert a folder of GIF files sequenced as "image-001", "image-002" and so forth into an MP4 movie file with a framerate of 10 frames per second and a reasonably high bitrate of 1800:

```
ffmpeg -r 10 -b 1800 -i image-%03d.gif movie.mp4
```

Make sure your files are named properly, because encoding will stop early if the program encounters a gap in the number sequence.

Abstraction

For most of us, the notion of animation brings to mind deliberate, structured sequences. In this section, I introduce some less-traditional ways of creating mind-blowing, computer-generated effects without having to know languages like Processing or Pure Data (both of these are very powerful, but not everyone who wants to animate knows how to code).

In my own work with video, screen recording tools have been indispensable. Sometimes I use them to capture animations I make in Pencil, because the movie export feature is broken in the version I use. Other times, I just want to capture some cool imagery on my screen without worrying about proprietary copyrights, so I take screen recordings


Figure 5. Using recordMyDesktop to Capture Electric Sheep

Animated GIFs

If you're a citizen of the Internet, you've no doubt noticed the recent proliferation of animated GIFs on sites like Tumblr.com. Now that more people have access to high-bandwidth network connections, the GIF art form is not so limited in resolution and number of frames as it was in the 1990s when tiny GIF animations originally rose to popularity in Geocities and Angelfire home pages. Modern GIF animations often display entire scenes from movies.

So, are you ready to pimp out some mad GIF skills?

With ImageMagick, it's easy to fashion an animated GIF from a sequence of non-GIF images:

```
cd /path/to/image/folder ; convert *.jpg animation.gif
```

The mother of all command-line GIF manipulation programs though is Gifsicle. Your images must already be in the GIF format to use it.

To create a GIF that animates just once, do:

```
gifsicle image1.gif image2.gif image3.gif > animation.gif
```

In most cases, you'll want your animated GIF to loop endlessly. You also may want to specify parameters, such as framerate. Try this for a dithered animation that loops at 10 frames per second:

```
gifsicle --loopcount=0 --delay 10 --dither image1.gif
 ➥image2.gif image3.gif > animation.gif
```

You also can use Gifsicle in reverse mode—that is, to extract the individual frames from an animated GIF. Just use the `--explode` argument:

```
gifsicle --explode animation.gif
```

Now, go out (and by "out", I mean to your nearest terminal) and explore all the neat tricks you can do with Gifsicle!

Here's one more to wet your feet:

Take a ready-made animated GIF with a white background and make it transparent:

```
gifsicle --transparent '#FFFFFF' --disposal 2
 ➥animation.gif > animation-transparent.gif
```

XaoS is a real-time, interactive fractal zoomer that will capture the eye of mathematicians and VJs alike.
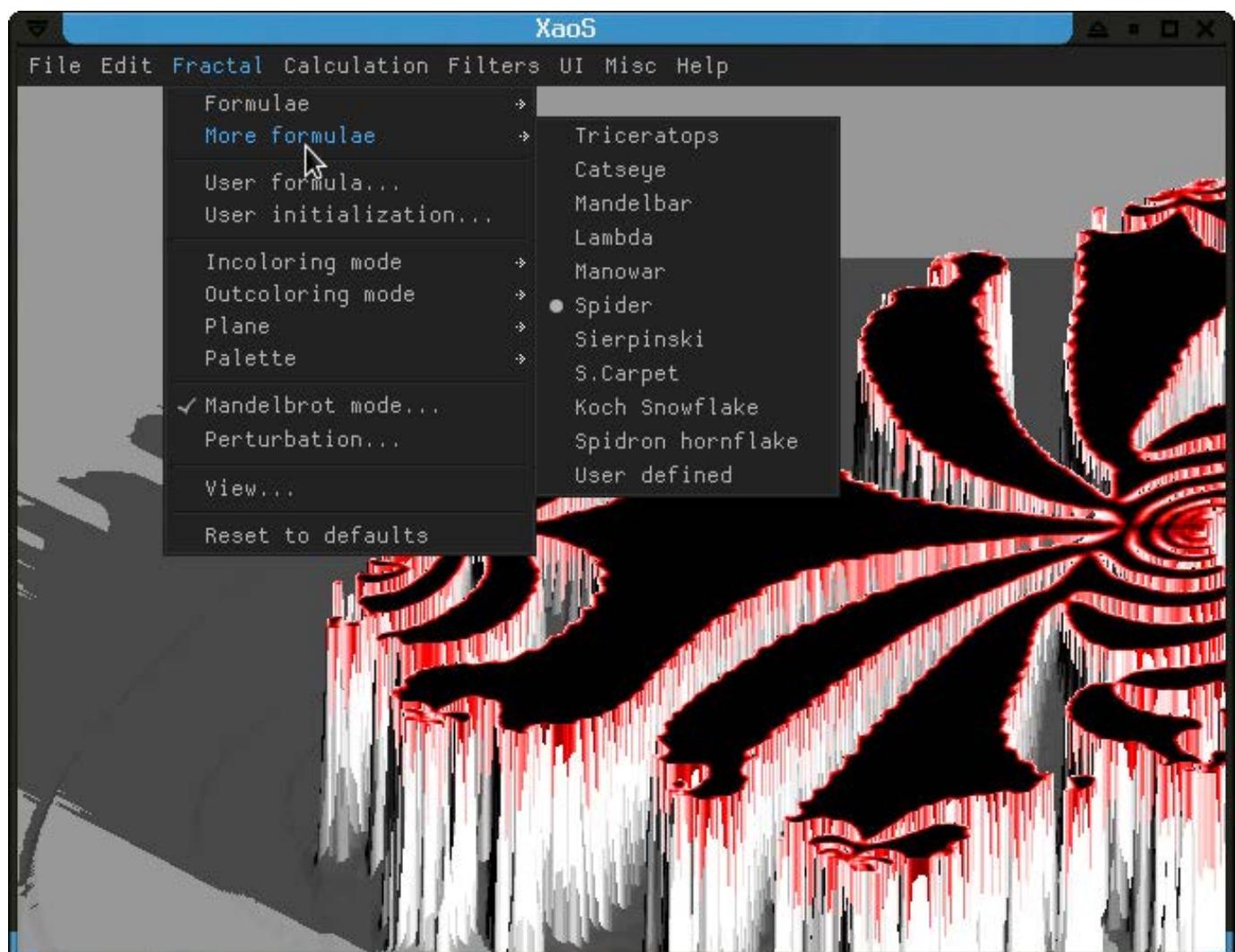
Figure 6. XaoS with the Pseudo-3D Filter Applied

of free software.

My preferred screen recorder is the bare-bones, command-line version of recordMyDesktop. Your distribution's repositories also might provide the graphical front ends GTK-recordmydesktop and QT-recordmydesktop, but I find those to be buggy and prone to crashes when recording long scenes. You can

record your entire screen with:

```
recordmydesktop screenrecording.ogv
```

The recording will start as soon as you enter the command, and it will stop when you press Ctrl-c. Read the man page for more options, such as recording a specific window (tip: find a window's ID with xwininfo).

Electric Sheep If you aren't familiar with the trippiest screensaver in the world, go on-line and look up some images of Electric Sheep. The software artist Scott Draves created Electric Sheep as a dynamic, collaborative fractal flame animation that runs on and by thousands of computers worldwide. Networked data determines the mutations of the various "sheep" in the animation, and users can vote on and contribute their own sheep. And because it's all free, anyone can use the images generated in this android dream.

So how do you take a screen recording of a screensaver? Well, guess what: Electric Sheep is a binary. Just enter electricsheep into your terminal and watch the magic in MPlayer.

If you want to create your own sheep, check out the program Qosmic.

XaoS XaoS is a real-time, interactive fractal zoomer that will capture the eye of mathematicians and VJs alike. You can change the fractal formulae and colors with many different parameters and filters. My favorite is the Pseudo-3D filter, which
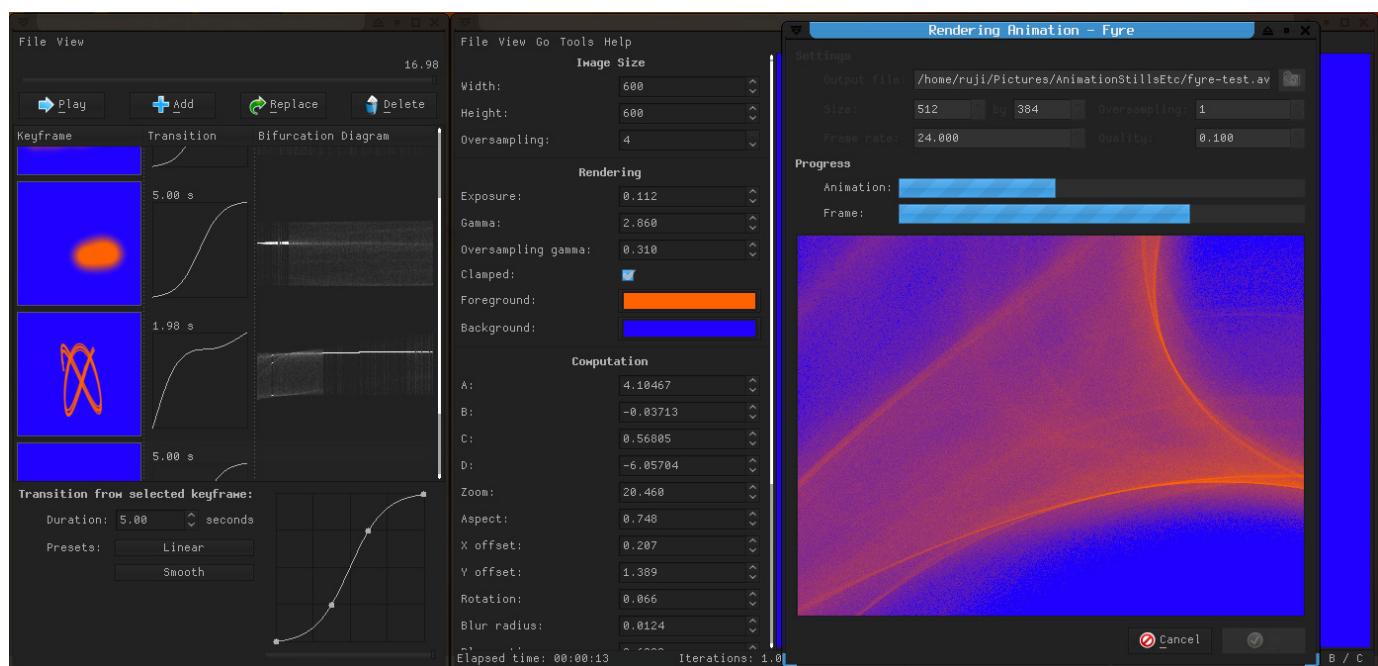
Figure 7. Animating in Fyre
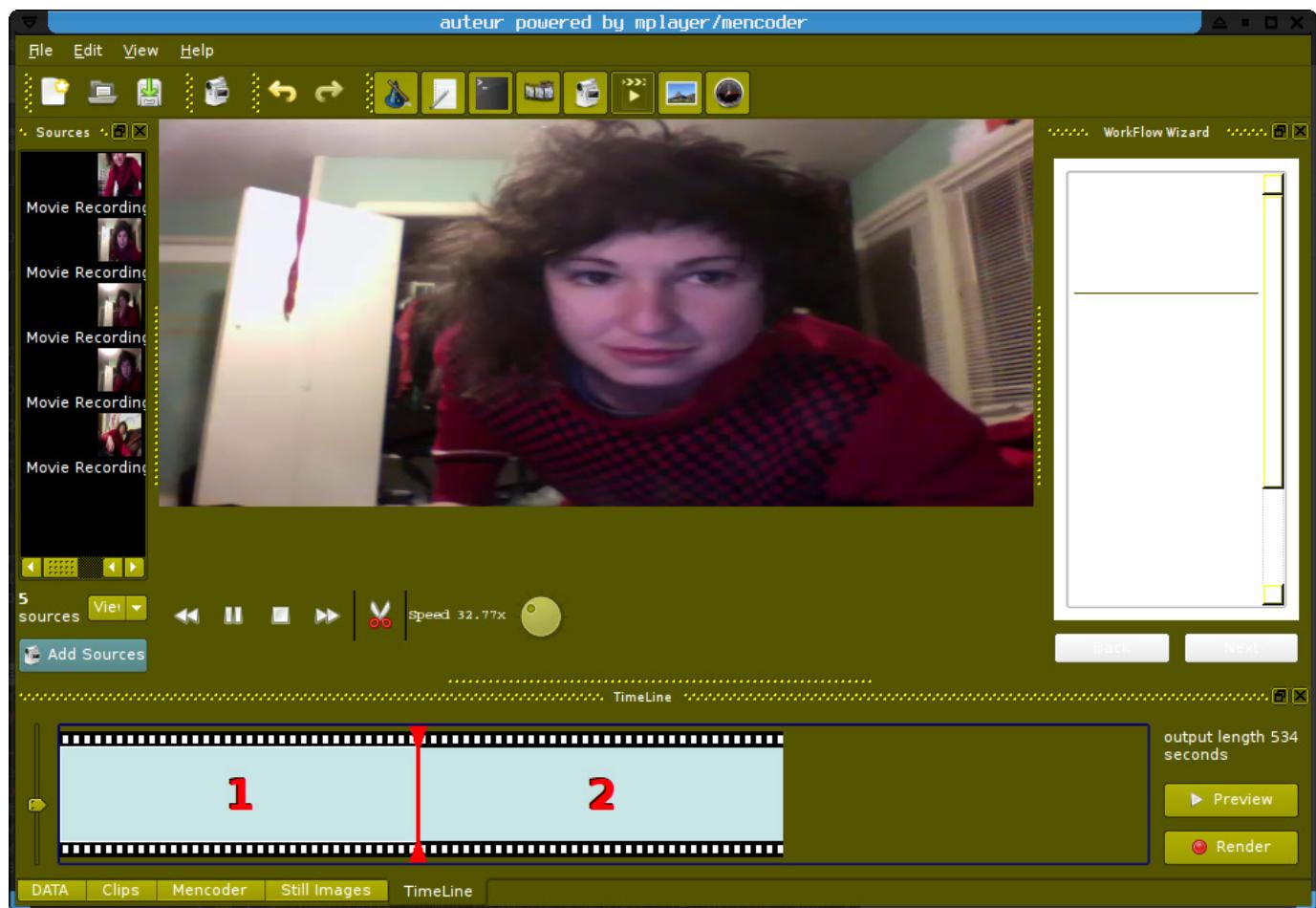
Figure 8. Testing Out Auteur
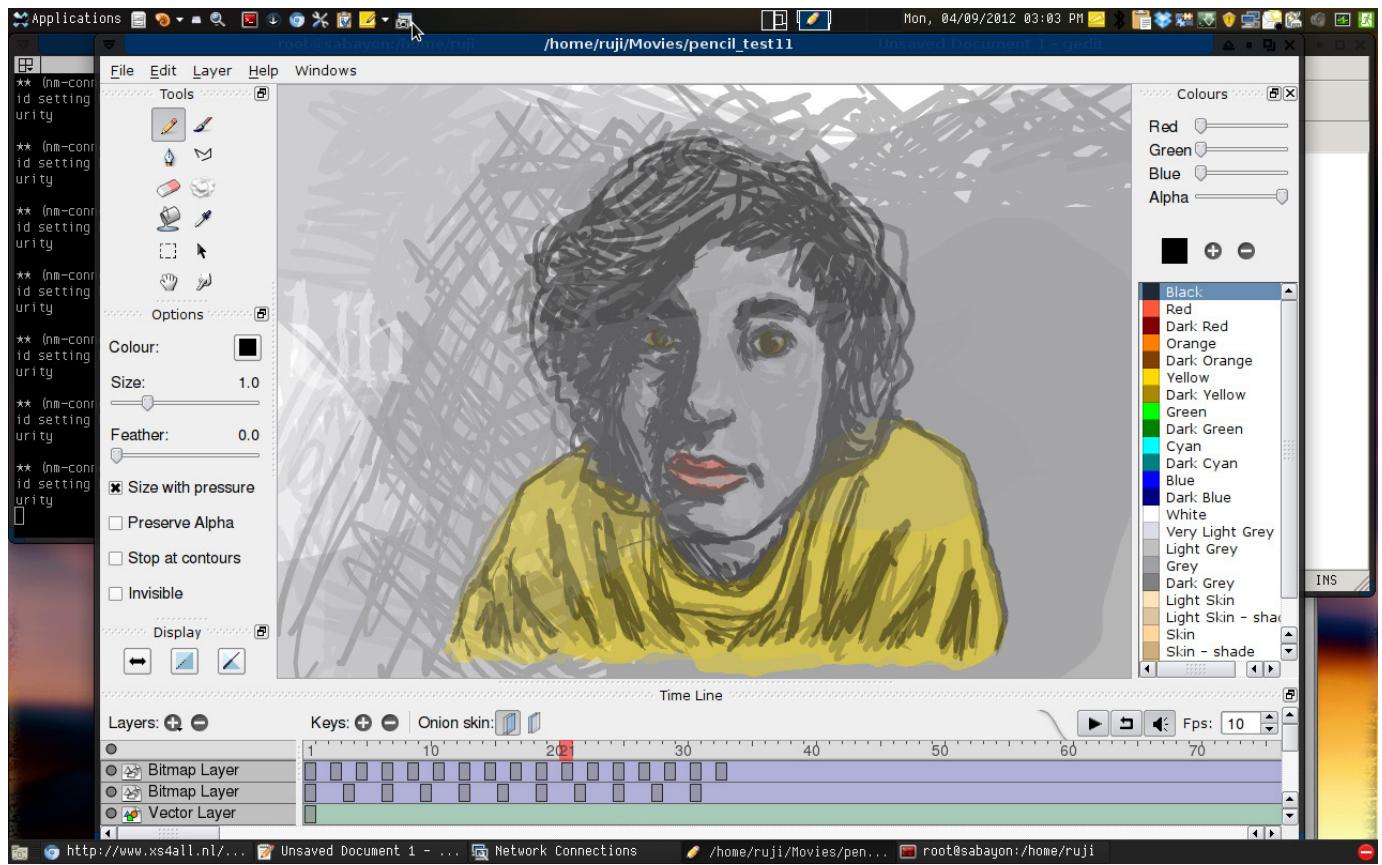
extrudes lines to generate what looks like a surreal landscape. Using the left and right mouse buttons, you can zoom in and out as if flying a plane over the "terrain".

Fyre Fyre is a program that generates and animates Peter de Jong maps. You don't need a screen recorder to make animations with this; you can enter key frames and render an AVI file directly from the program. As you can see from the screenshot shown in Figure 7, Peter de Jong maps make for some neat, abstract images.

Alphas and More to Look Out For

Unfortunately, there is not enough space in this article or in my brain to cover all the new video-related Linux software that's in development. In lieu of a complete list, I'll provide you with the names of a few projects that I expect to be worth checking out for both developers and end users.

Auteur Auteur is one cool new kid on the block. I first heard of this project in an episode of the podcast "The Bad Apples" (which has since been re-branded as "GNU World

Figure 9. An Animation I Made in Pencil

Order"), produced by Seth Kenlon, aka Klaatu, who is also a developer on the Auteur team. Klaatu noted the absence of a truly solid nonlinear video editor for Linux, so he set out to make one with all the features he felt existing software was lacking. The Web site currently says that the project is frozen due to lack of programmers—so programmers, why not help out with a promising alpha?

VLMC The folks behind the VLC media player have a nascent project called VLMC (VideoLAN Movie Creator). The latest releases are still basic and not quite stable, but knowing what the VideoLAN team is capable of, I am sure this will mature into a serious nonlinear video editor. They currently are looking for contributors.

Pencil Pencil is a traditional 2-D animation program, which, although still in beta, already fills a gaping hole in the sphere of Linux animation tools. It allows drawing in both vector and bitmap formats as well as importing images and sounds. My trials with Pencil have been basic but mostly satisfactory, although the video export feature appears broken in Linux. I have compensated for that and made some cool videos anyway

simply by taking a screen recording during animation playback in Pencil. There is an active community of Pencil users who post animations on the Pencil Web site's gallery. Pencil is similar to Synfig Studio, but I find the interface easier to navigate.

Puredyne Puredyne is a multimedia Linux distribution based on Ubuntu and Debian Live, specifically designed for real-time audio-visual processing. Many of the tools and APIs I haven't had the verbal real estate to cover in this article (such as FreeJ, Gephex and DataDada) are included either in the distribution itself or in optional modules.

And, there you have it, animators and filmmakers. I hope this article inspires a cool music video or two!■

Rebecca "Ruji" Chapnik is a freelance creator of miscellanea, including but not limited to text and images. She studied art at the University of California, Santa Cruz, and book publishing at Portland State University. She went on to study Linux in her bedroom and also in various other people's bedrooms, crouched anti-ergonomically before abandoned Windows computers. Ruji currently lives in Portland, Oregon. You can find her experiments at http://rujic.net and her comics at http://dondepresso.rujic.net.

Resources

LiVES: **http://lives.sourceforge.net**

Frei0r: **http://piksel.org/frei0r**

ZS4 Video Compositing: **http://zs4.org**

t@b Software: **http://www.thugsatbay.com/tab/?q=software**

ZS4 Download: **http://zs4.net/download**

GIMP: **http://www.gimp.org**

Psychological Image Collection at Stirling (PICS): **http://pics.psych.stir.ac.uk**

Stopmotion: **http://stopmotion.bjoernen.com**

JPEG to MJPEG-AVI Converter: **http://sourceforge.net/projects/jpegtoavi**

FFmpeg: **http://ffmpeg.org**

Gifsicle: **http://www.lcdf.org/gifsicle**

recordMyDesktop: **http://recordmydesktop.sourceforge.net**

Electric Sheep: **http://www.electricsheep.org**

GNU XaoS: **http://wmi.math.u-szeged.hu/xaos/doku.php**

Fyre: **http://fyre.navi.cx**

Auteur Non-Linear Editor: **http://auteur-editor.info**

VLMC: **http://www.videolan.org/vlmc**

Pencil—Traditional Animation Software: **http://www.pencil-animation.org**

Puredyne: **http://www.puredyne.org**

Wall-Mounted Dual-Head Desktop

Would you like to do less and get more? Try a wall-mounted PC with Linux, multihead with Xinerama and input sharing with a tiling window manager for less input and more output.

MARCIN TEODORCZYK

Would you like to do less and get more? Yes, that's a standard salesman question, but I couldn't resist using it in this article, because it just fits so well.

One thing that always has bothered me is the amount of work I've had to do to change hardware on my PC. Of course, I don't have screws in my case—it's quite pointless when you take your case off every few days. But even so, adding or removing some hardware was a painful job.

Another inconvenience I've repeatedly encountered was extensive use of virtual desktops. My 24" display just wasn't enough, and bigger ones were too high to be comfortably usable for me. Even with a relatively small font size, a Web browser and two terminals for input plus two terminals with logs preview didn't fit well on one desktop. Thus, very often I had to switch every few seconds between two—a browser with input terminals and terminals with logs.

The above reasons were sufficient enough to end my laziness for a while and make me to do a little research. I found two solutions that suited me perfectly. The first one was a wall-mounted PC. And, the second was multihead with Xinerama and a tiling window manager. In this article, I explain how I made my first dual-head wall-mounted PC and shared its input with a laptop.

Wall-Mounted PCs

A wall-mounted PC is exactly what the name says—a PC mounted on a wall. As an example, you can see my machine, called VMS (Virtual Machine Server), shown in Figure 1. Why would anyone do something like this?

First, it's a lot of fun to make it. If you're into case-modding, you might like making wall-mounted PCs. Second, such a computer has easier accessible modules. If you have a constantly unscrewed or opened case, a wall-mounted PC might be a good solution for you. Third, it is generally easier to keep clean, especially if it's uncovered.

Figure 1. VMS (Virtual Machine Server): a wall-mounted, dual-head PC and a laptop connected with Synergy.

Also, it uses less space, because it hangs flat on the wall, usually at a fair height—taking space that typically is unused. It's also easier to control the wires. Finally, and this might be a matter of opinion, but I think it just looks good. It has a bit of a working-on-a-laptop feel, because you have it all before your eyes.

As a disadvantage, messing with the case might break the air circulation and can be dangerous for the cooling system. Thus, when doing such a project, you should think it out and monitor/benchmark the hardware for the first few hours or days, especially the temperature sensor readings. Another disadvantage is mobility.

Assuming everything is wireless, you need to have one power supply cord. Typically, you also will have Ethernet and display cables, and possibly a keyboard and mouse as well. The mainboard is in a fixed position on the wall. You can move a PC with a case and peripherals relatively easily to other place, but a wall-mounted PC requires you to disconnect all cables and prepare a new wall. When it has no case, you have to be extremely cautious too.

Multihead with Linux

Whether or not you decide to use a wall-mounted PC, you might want to try a multihead display setup. You'll find it useful especially if you often switch between windows or virtual desktops. Personally, I find the three-display setup to be the most optimal. One display is for my activity, and the other two are for monitoring purposes, such as log tails, network traffic dumps or documentation. This doesn't require me to switch focus between displays, but it allows me to see actual information in a blink of an eye.

If you decide to try multihead with Linux, you'll have at least three options for the hardware: one multihead graphic card, a few typical graphic cards or some mix of those two.

Multihead graphic cards are more expensive than typical ones and not always well supported by Linux, although you often can find proprietary drivers. On the other hand, they take less slots on the mainboard, and if well supported, they are very easy to set up. Thus, when you decide to use such a solution, the main thing you should consider is Linux support. Typical multihead graphic cards deliver two or four video outputs, but you also can find ones with more.
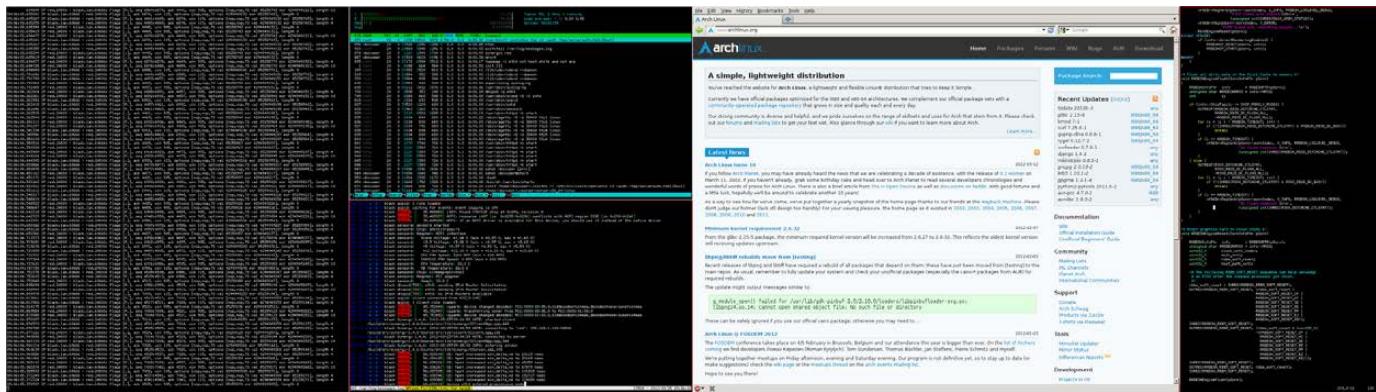
Alternatively, you can use a few typical graphic cards. Sure, they'll take more slots on the mainboard, but they have two advantages. First, if each one is supported by Linux, they should work with multihead with no problems. Second, you easily can add/remove displays/cards to/from your setup.

On the software side, you have again at least three possibilities: a proprietary solution, Xorg with a few screens or Xinerama.

Proprietary solutions are popular with multihead graphic cards. For a few graphic cards, you either can set up Xorg to use a few screens or use Xinerama. The first solution is very inconvenient for typical desktop use, because it requires you to manipulate the DISPLAY environment variable. I focus on Xinerama here. Xinerama makes one virtual desktop from all your displays, and this is what you usually would want.

Tiling Window Managers

A tiling window manager might come in handy when working with more than one display, especially if you mainly tend to use the keyboard. In such managers, windows always occupy

Figure 2. Two HD Displays Controlled by Xmonad

the whole desktop area—there is no window stacking (except in a few cases, such as video playing). Figure 2 shows an example screenshot of two displays controlled by the Xmonad tiling window manager. Other examples of tiling window managers include Awesome, WMII or Ratpoison.

Output/Input Sharing

Apart from the multihead setup, there are a few solutions for output (display) and input (keyboard and mouse) sharing between PCs over the network. Two examples are Xdmx and Synergy, respectively.

Xdmx lets you use another PC display as a second display for your first PC. From the user point of view, it works just like Xinerama, but instead of connecting the second display to the second graphic card, you connect the second PC to your network.

Synergy lets you use your computer's keyboard and mouse to control another computer that is connected to the

network. The focus can be switched either with mouse or keyboard. For example, if you have your second PC to the right of your first one, you can set up Synergy to switch focus when you move the cursor of your first PC to the right over the right edge of your screen. In such a case, the cursor will appear on the second PC screen immediately.

VMS (Virtual Machine Server)

VMS is my first wall-mounted, dual-head PC. It has no input devices, and it works as a personal lab and computing machine. It's connected to an Ethernet network and controlled only from the laptop connecting to the network from time to time. It works as a sort of server and docking station for the laptop, extending its power and screen area while keeping the input system.

I set up this system in three steps. First, I made the wall-mounted "case". Second, I set up dual-head, and finally, I removed the server keyboard and mouse and configured Synergy to share input

with the laptop connected to the LAN.

For the wall-mounted "case", I selected the simplest and fastest solution: a board, a few screws and perforated metal tape. It took me no more than one hour to build it, but you can go further. For example, you could use a metal plate instead of a wooden board as well as hinges and Plexiglas as a cover.

For the dual-head setup, I chose two cards. First, I checked whether they worked separately. After making sure they did, I plugged in both of them and checked the BIOS settings. Typically, one card always is set up as a master—this one will work during bootup.

After booting Linux, I checked how the cards were connected with lspci:

```
[root@vms ~]# lspci | grep -i vga
01:00.0 VGA compatible controller: Advanced Micro
  ➥Devices [AMD] nee ATI RV370 [Radeon X550]
03:05.0 VGA compatible controller: Advanced Micro
  ➥Devices [AMD] nee ATI RV100 QY [Radeon 7000/VE]
```

The above shows what driver to use as well as the bus ID (first column), which will be necessary for proper Xorg configuration. In my case, the driver for both cards is Radeon. The location of the configuration file depends on your installation, but usually it is located by Xorg in /etc/xorg.conf. Listing 1 shows my configuration.

The Device, Monitor and Screen sections are not much different from a single-head setup, but they are

duplicated—one for each graphic card and display. The only difference that you might see is the bus ID I read earlier using PCI—this is used to distinguish the cards.

The Xinerama setup is done in the ServerLayout section. I define two screens and their relative position. Screen 1 is on the right of Screen 0. Finally, I turn on Xinerama with `Option "Xinerama" on`.

The last thing I needed to do before running Xorg with two displays was choose a window manager.

Xmonad

I've been using Xmonad for a while on my laptop, and since it has great support for Xinerama, I decided to try it first for the multihead setup. Xmonad delivers nine desktops and three layouts by default.

In Xinerama setup, virtual desktops are shared between the screens. That means you can switch between the screens and between the virtual desktops on each screen, but virtual desktop nr 3 on screen 0 is the same as virtual desktop nr 3 on screen 1. Also, you can't see the same virtual desktop on both screens. Thus, when, for example, you have virtual desktop nr 3 present on screen 0 and want to switch to virtual desktop nr 3 on screen 1, screen 0 goes blank.

On each desktop, by default, you can choose either vertical, horizontal or full-screen layout. You easily can

Listing 1. Dual-Head Xinerama Xorg Configuration Example

```
Section "ServerLayout"
        Identifier      "X.org Configured"
        Screen       0  "Screen0" 0 0
        Screen       1  "Screen1" RightOf "Screen0"
        InputDevice     "Mouse0" "CorePointer"
        InputDevice     "Keyboard0" "CoreKeyboard"
        Option          "Xinerama" "on"
EndSection

Section "Files"
        ModulePath   "/usr/lib/xorg/modules"
        FontPath     "/usr/share/fonts/misc/"
        FontPath     "/usr/share/fonts/TTF/"
        FontPath     "/usr/share/fonts/OTF/"
        FontPath     "/usr/share/fonts/Type1/"
        FontPath     "/usr/share/fonts/100dpi/"
        FontPath     "/usr/share/fonts/75dpi/"
EndSection

Section "Module"
        Load  "glx"
        Load  "dri"
        Load  "record"
        Load  "dbe"
        Load  "dri2"
        Load  "extmod"
EndSection

Section "InputDevice"
        Identifier   "Keyboard0"
        Driver       "kbd"
EndSection

Section "InputDevice"
        Identifier   "Mouse0"
        Driver       "mouse"
        Option       "Protocol" "auto"
        Option       "Device" "/dev/input/mice"
        Option       "ZAxisMapping" "4 5 6 7"
EndSection

Section "Monitor"
        Identifier   "Monitor0"
        VendorName   "Monitor Vendor"
        ModelName    "Monitor Model"
EndSection


Section "Monitor"
        Identifier   "Monitor1"
        VendorName   "Monitor Vendor"
        ModelName    "Monitor Model"
EndSection

Section "Device"
        Identifier   "Card0"
        Driver       "radeon"
        BusID        "PCI:3:5:0"
EndSection

Section "Device"
        Identifier   "Card1"
        Driver       "radeon"
        BusID        "PCI:1:0:0"
EndSection

Section "Screen"
        Identifier   "Screen0"
        Device       "Card0"
        Monitor      "Monitor0"
        SubSection "Display"
                Viewport  0 0
                Depth     16
        EndSubSection
        SubSection "Display"
                Viewport  0 0
                Depth     24
        EndSubSection
EndSection

Section "Screen"
        Identifier   "Screen1"
        Device       "Card1"
        Monitor      "Monitor1"
        SubSection "Display"
                Viewport  0 0
                Depth     16
        EndSubSection
        SubSection "Display"
                Viewport  0 0
                Depth     24
        EndSubSection
EndSection
```
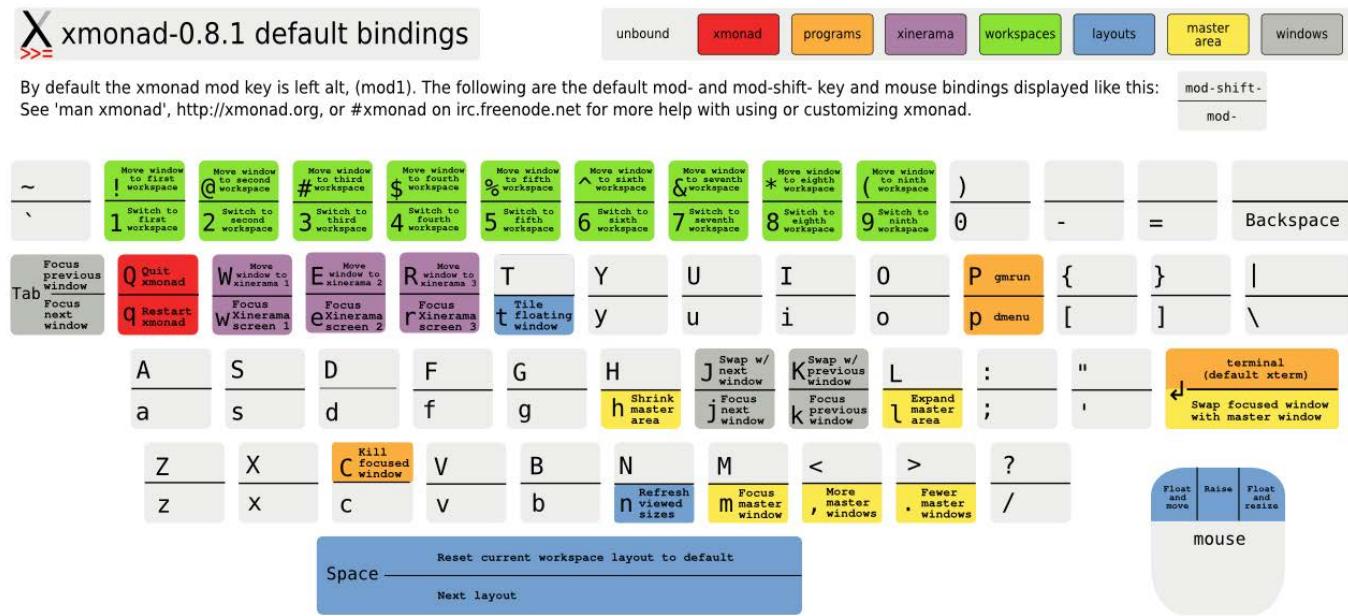
Figure 3. Xmonad Cheat Sheet

extend Xmonad with other layouts, but for 16:9 displays, I recommend trying three-column.

The Xmonad's default shortcuts are based on Vi-like keybindings. Figure 3 shows Xmonad's cheat sheet. You also can define custom keybindings.

Connecting a Laptop with Synergy

Next, I took advantage of the wall-mounted PC setup and connected a laptop to it with Synergy. The laptop works as a third (middle) screen and a master input device. In practice, when connected to the network, I'm able to move the laptop's mouse pointer seamlessly between screens as well as switch the focus with the keyboard.

Synergy works as a client/server application. The server has control over the input. The client gives away control

over its input, but its input devices also can be used simultaneously. In my case, the laptop is the server.

You need to configure only the server. Listing 2 shows an example configuration. It is placed in /etc/synergy.conf.

In the section screens, I define the hosts that I want to be connected. In the section links, I define the relations between the hosts and their screens. Finally, I define the keystrokes that can be used to switch focus.

With the server configured, you can start the synergys dæmon as well as the client on the laptop with `synergyc servername`, where `servername` is the name of synergy server machine running syngergys. It is important to make sure that the hostnames you used in the configuration file (in my case red and

Listing 2. Example Synergy Configuration

```
section: screens
        red:
        vms:
end


section: links
        red:
                right    = vms
                left     = vms
        vms:
                right    = red
                left     = red
end


section: options
    keystroke(control+alt+l) = switchInDirection(right)
    keystroke(control+alt+h) = switchInDirection(left)
end
```

vms) are resolvable.

Also, keep in mind that communication between hosts isn't secure. If you want to secure it, you can use SSH tunneling. For example, on the client machine, you can run:

```
ssh -L 24800:server:24800 server
```

24800 is the default listening port for the Synergy server, and server is the hostname for the server. After that, you can run the client with:

```
synergyc localhost.
```

Summary

Building VMS and connecting it with my laptop was a lot of fun, but using it is a real pleasure compared to how I worked earlier. Nonetheless, what I've shown here is only one example of what you can do to make your own power-user desktop. The possibilities are endless. For example, I'm thinking about a five-display setup now, or one of those wearable screens that looks like glasses and gives the impression of a 16-meter display—if only the resolution of such a solution was more than 800x600.■

Marcin Teodorczyk is a GNU/Linux user with more than ten years of experience. For the past three years, he's been using Arch Linux exclusively on his personal computers. Marcin has an M.Sc. degree in IT and works as a security officer. In his spare time, he works out or writes articles for IT-related magazines.

Resources

Xinerama: **http://sourceforge.net/projects/xinerama**

Xmonad: **http://xmonad.org**

Distributed Multihead X Project: **http://dmx.sourceforge.net**

Synergy: **http://synergy-foss.org**

ZFS and Btrfs: a Quick Introduction to Modern Filesystems

In this article, I explain how to install a ZFS kernel module on Linux, create a filesystem and compare it to existing Btrfs utilities bundled with your Linux distro. HOWARD POWELL

I am a filesystem geek. I have to be—I run a small department and manage the backups of about ten different mission-critical servers. I buy hard drives by the 20-pack, several times a year. Disks are cheap, but not always reliable, so I've learned to adapt by using the latest generation of filesystems to overcome drive failures.

In the past, I relied on RAID hardware (and/or software) to handle drive redundancy and to increase performance, but that no longer is necessary. I also used LVM to make my storage more flexible and easier to manage. The newer

filesystems like ZFS and Btrfs (have or will soon) implement redundancy and internal checking for consistency without that extra layer of expensive RAID controllers or slow software. They also handle pools of storage in a fundamental way that makes an LVM layer less useful.

The biggest concept to grasp with ZFS and Btrfs is that ZFS and Btrfs expect disks to be disks. The usage of expensive hardware or slow software RAID systems is unnecessary and conflicts with how these filesystems expect to talk to storage disks. You still can use hardware RAID if you want, but you're removing

some of the built-in safeguards that help your filesystem prevent loss of data.

ZFS

First, I'm going to talk about ZFS. I start with a quick overview of how to download and install the source code for this system on your Red Hat-compatible OS, and help you set up a small ZFS storage pool. Then, I provide some code snippets you can use to create a rotating snapshot-like backup script using ZFS's native snapshot capability. Assuming you create your storage pool on multiple disks in a mirror or RAID-Z configuration, you will have built-in protection against bitrot—the natural propagation of errors in your data from misreads or writes to disk. If you then add to your system some off-site backups, you'll have a pretty good start to a robust backup solution.

The ZFS filesystem originally was created by Sun Microsystems for its Solaris operating system. ZFS has been something that many Linux users have desired for almost a decade because of its amazing features and simple-to-implement flexibility. Unfortunately, Sun decided to release ZFS under the CDDL license, which ultimately has caused the kernel gurus never to add support to the main Linux kernel. Some people have ported ZFS to FUSE so that it can run in userspace and not taint the kernel, but this is slower than native

kernel-space access. Recently, a group at Lawrence Livermore National Labs ported ZFS so that it's a separately compiled kernel module that can be installed on your system. (This, of course, taints your kernel's license.) This module is considered to be in release candidate state, but the core ZFS code itself is based on the fairly stable Solaris version.

ZFS is notable because it's a mature implementation of a copy-on-write filesystem. Copy-on-write (COW) means that when your data is read (for example, when you open a file) and then modified, a new copy is saved to disk, and the old copy is either re-allocated for future use or preserved as a "snapshot" of the current state of the filesystem. This means that no file can ever be "in the middle" of a write operation, so corruption leading to long fscks are greatly reduced.

To get started with ZFS, first you need to download the source from **http://zfsonlinux.com**. You need two packages: the Solaris Porting Layer (this provides the Solaris kernel APIs) and the ZFS source.

You also need kernel-devel and a few other packages—this should work on a Red Hat-compatible distribution:

```
yum groupinstall "Development Tools"
yum install zlib-devel libuuid-devel libblkid-devel
  ➥libselinux-devel parted lsscsi
```

Once the dependencies are installed, first you should compile the Solaris Porting Layer. You have to unzip the SPL, `cd` into the directory, run `./configure` and `make rpms`. When done, you should have XYZ rpm files for your system, which you can next install with your package manager. For example, `rpm --Uvh *.x86_64.rpm` should work on a 64-bit system.

Next, you need to unzip and configure the ZFS source directory. Again, run `make rpms` and install the resulting rpm files it creates.

Once installed, you can begin creating a filesystem. Reboot or do `service zfs start` to load the modules and prepare your system for ZFS:

```
zpool create puddle /dev/sdb /dev/sdc
```

(Note that ZFS can use partitions, but if you give it a raw disk device, it automatically will create partitions and use the entire space effectively. This is the best practice for ZFS.)

The above example creates a concatenated storage pool called "puddle". Storage on zfs is called "pools" or collections of disks and disk partitions. A storage pool is roughly analogous to the metadevices created by mdadm or lvm. The puddle filesystem you just created will be mounted at /puddle and can be browsed and used just like any other filesystem. Note the first big change—you do not need an entry in fstab or to mount the filesystem manually; ZFS takes care of this for you.

A mirror (equivalent in concept to a RAID-1) can be created with two or more devices by using the following command:

```
zpool create mirror puddle /dev/sdb /dev/sdc
```

Finally, a RAID-5 equivalent pool can be created with:

```
zpool create raidz puddle /dev/sdb /dev/sdc /dev/sdd
```

Oh, I almost forgot—if you prefer RAID-6 instead of RAID-5, and have the disks to throw at this, there's a solution for you also:

```
zpool create raidz-2 puddle /dev/sdb /dev/sdc /dev/sdd /dev/sde
```

It should take only a moment or two to initialize the device.

You can check your new storage pool's status by using:

```
zpool status
```

Next, you may decide you want to move your mounted filesystem to a different mountpoint. You can use the ZFS command-line utility to do this:

```
zfs setmount puddle /mnt/puddle
```

You also can use the ZFS utility to create new sub-pools of storage that can be handled independently or together with the main pool:

```
zfs create puddle/droplet
```

Snapshots are a very effective way to make instantaneous same-disk backups of your files:

```
zfs snapshot puddle/droplet@today
```

You can mount these snapshots to copy or recover data:

```
mount -t zfs puddle/droplet@today /mnt/
```

You also can list the filesystems and snapshots, and you can see how much disk space each is consumed by using the zfs list command:

```
zfs list puddle     # this will list all subvolumes below puddle

zfs list -t snapshot puddle   # this will list all snapshots
                              # of puddle and subvolumes
```

Finally, maintenance of your filesystem is vital. With ext3/4 and other filesystems, you use fsck to make sure your system is coherent and healthy, but this requires you to take the filesystem and/or machine off-line, and it could take hours to check a large multi-terabyte disk. With ZFS, file "scrubbing" happens on-line while the system is active and available for use. Scrubbing scans through every file and makes sure that the internal checksums are still valid and correct. If you have redundant storage (a RAID-Z or a mirror), the filesystem will be self-healing and your data will be fixed automatically if any

filesystem problems are detected:

```
zpool scrub puddle
```

Much like fsck, a scrub operation can take a few hours, but the big difference is that your system remains on-line and ready for use! Go ahead and work with your system; you will not cause the scrub operation any problems.

Technically, a scrub of each file happens automatically every time that file is opened. The above scrub command will check every file on that storage pool,

and best practices suggest the scrub command should be run periodically to check files that are rarely accessed.

Okay, now for a few code snippets to show you how you can use ZFS to your advantage for backups:

```
zfs list -H -r -d1 -t snapshot puddle
```

This snippet lists, in chronological order, all of the snapshots for your pool.

You easily can modify it to show just a sub-pool:

```
zfs list -H -r -d1 -t snapshot puddle/droplet
```

If you feed that to a `head -1` command, you instantly have the oldest snapshot on your system. You can use your favorite shell-fu commands to decide whether to keep or delete the old snapshot by:

```
zfs destroy puddle/droplet@today
```

Let's say you're copying /home to /puddle once a day, and you now have five days' worth of snapshots. You want only the last five days of backups, so let's write a script to delete the oldest snapshot, rsync /home to /puddle, and create a new snapshot:

```
OLDEST=`zfs list -H -r -d1 -t snapshot puddle|head -1`
zfs destroy ${OLDEST}
rsync -av --delete /home /puddle
zfs snapshot puddle@'date +%Y%m%d-%H.%M.%S'
```

And there you go—with as few as four lines of code, you have a rotating backup script for your data.

For those of you really interested in learning about ZFS, remember that it's a mature filesystem that's been around for almost a decade and heavily developed and invested in by many organizations, notably Sun and Oracle. Only the Linux kernel module implementation is new, so if you already have a ZFS filesystem created on a BSD UNIX or on Solaris, you easily can import it into your Linux system or vice versa. The Linux module is being actively maintained and updated periodically.

There is a lot of great documentation on-line about ZFS, but one of the best documents for people investigating using ZFS in a real environment is the ZFS Best Practices Guide, which references Solaris heavily, but don't let that scare you away (see Resources).

Btrfs

Now, some people may prefer to avoid downloading source code and compiling their own kernel modules (even though it's trivial to do so on a standard distribution these days). For these people, Btrfs (B-tree File System) is a GPL-licensed copy-on-write filesystem already included in most Linux distributions. Btrfs still is undergoing heavy development, and many features, such as parity-based

RAID, aren't yet complete.

Btrfs modules already should be installed as part of modern kernels (I can confirm it's in RHEL 6.0 and above). You need to install the btrfs-progs package to create a filesystem and work with it:

```
yum install btrfs-progs
```

Now, you can create the filesystem on a second, unused partition (sdf1) and assign it a label:

```
mkfs.btrfs -L/local /dev/sdf1
```

To pool two or more devices together:

```
mkfs.btrfs -L/local /dev/sdf1 /dev/sdg1
```

To create a mirrored filesystem for your data on two or more devices:

```
mkfs.btrfs -L/local -d raid1 /dev/sdf1 /dev/sdg1
```

With Btrfs, you need to create an entry in /etc/fstab. You could use "LABEL=/local" with the above example, but because RHEL 6 prefers UUIDs instead of labels, you can discover the UUID and use it instead with btrfs-show:

```
btrfs filesystem show /local
```

Now you can add it to the fstab, such as:

```
UUID=[something here]    /local       btrfs   defaults   1 2
```

Finally, to mount the disk, just run the `mount` command. If you want to relocate the filesystem to a different mountpoint, update fstab and remount the filesystem.

To create a snapshot of a Btrfs mount:

```
btrfs subvolume snapshot /local /local/snapshot
```

The snapshot is mounted automatically at /local/snapshot. You cannot mount these snapshots outside of the Btrfs tree—they have to be subvolumes of your main Btrfs mountpoint.

Just like ZFS, the "scrub" equivalent of a file happens automatically each time that file is accessed or read.

To list all of the snapshots on your Btrfs mount:

```
btrfs subvolume list /local
```

Finally, to destroy a snapshot:

```
btrfs subvolume destroy /local/snapshot
```

Now for a few other notes on Btrfs. Just like ZFS, the "scrub" equivalent of a file happens automatically each time that file is accessed or read. However, no on-line scrub of the entire filesystem currently is available. You could, however, use the find command to simulate a scrub:

```
find /local -mount -type f -exec cat '{}' > /dev/null \;
```

Okay, and now to replicate the functionality of the ZFS rotating snapshot backup I demoed before:

```
OLDEST=`btrfs subvolume list /local|head -1| awk '{print $NF}'`
btrfs subvolume destroy /local/${OLDEST}
rsync -av --delete /home /local
btrfs subvolume snapshot /local /local/snap-`date +%Y%m%d-%H.%M.%S`
```

Now you can wander through your snapshot directory at will and copy data as necessary.

Other Similarities

I've demonstrated some simple things you can do with ZFS and Btrfs here. Both filesystems have other features implemented that I did not mention, such as on-line compression using GZIP or LZO algorithms. Both filesystems support user and group quotas.

Conclusions

Personally, I have worked with ZFS longer and consider it to be more stable than Btrfs due to its heritage and inherited code base from Solaris and the BSD UNIXes. However, Btrfs is being worked on heavily by multiple groups, and at some point, it's likely that the features of Btrfs will advance further than those available in ZFS as more distributions add support for it and as more hackers get to play with it.

If you have pre-existing scripts or filesystems from other OSes that use ZFS, the ZFS on Linux Project is just what you need to get these filesystems working with your Linux OS efficiently and easily.

On the other hand, Btrfs offers the possibility to convert an ext3 or ext4 filesystem to Btrfs, which is perfect if you already have data in place on your disks. This is a powerful tool on large storage servers where downtime due to data migration must be minimized.

I hope these examples and this quick introduction inspire you to go out and look at the new filesystems available and help contribute feedback to the developers on features you need. With your help, we finally can break free from expensive RAID hardware and start to think of disks as just pools of storage to be used.■

Howard Powell has been working with Linux and Solaris systems for a decade at the University of Virginia Astronomy Department. He loves filesystems, and you can reach him at bofh@virginia.edu.

Resources

ZFS on Linux Project: **http://zfsonlinux.org**

Best Practices Guide: **http://www.solarisinternals.com/wiki/index.php/ZFS_Best_Practices_Guide**

Btrfs: **http://en.wikipedia.org/wiki/Btrfs**

BTRFS Fu: **http://www.funtoo.org/wiki/BTRFS_Fun**

DOC SEARLS

What Comes after Smartphones?

Will it just be smartTV?

Android (and therefore Linux) is winning the smartphone war, at least in terms of market share. Eric Raymond has been charting Android progress with great care during the past several years, and he noted in April 2012 that Android has moved into the majority position, breaking past 50%, according to comScore data. (You can keep up at this link: **http://www.catb.org/esr/comscore**.)

ESR's chart shows that the smartphone war is now between just two OSes. Only Android and Apple are on the increase, with Apple breaking past 30% (though gaining on a lower slope than Android). All the other OSes are, as telecom maven Tomi Ahonen says, going over a cliff (**http://www.brightsideofnews.com/news/2012/4/3/introducing-the-cliff-theory-how-handset-makers-die.aspx**). Microsoft is in the single digits and dropping, and RIM announced in April that it was dropping out of the consumer marketplace. That's a death rattle, without the sincerity.

History happens fast in the smartphone marketplace. It's amazing to think that the smartphone as we know it today was born only four years ago, when Apple began distributing apps through the iTunes Store and the first Android phones appeared.

Smartphones are two things. On the smart side, they are whatever we make of them, using any of the countless apps that run on them. They can be magazines, flashlights, maps, seismographs, GPSes, cameras, calendars, games, televisions or whatever. On the phone side, they are still, well, phones. Nearly all of them are bought with contracts that tie them to a phone company. Each has its own phone number and its own billing plans for voice, data and text. In terms of form and status, smart is the adjective and phone is the noun.

So, even though talking to other people in real time on a handset no longer requires phone companies (witness the success of Skype, GoToMeeting and the rest), our phones are still very much tied to phone systems. To a degree that should make us uncomfortable, our mobile networked world is one contained and controlled by our relationships with phone companies.

What those companies want most these days is to get into the TV distribution business. That is, they want to move TV "content" over their mobile networks, onto phones and tablets. Given that TV always has had a huge share of the time we spend staring at glowing rectangles, companies like Verizon can't wait to turn their well-oiled billing machinery loose on TV viewing over handhelds on mobile phone networks.

On March 30, 2012, *The Wall Street Journal* reported that Verizon CEO Lowell C. McAdam said the company would have a cable-like video service going by the end of the year, if regulators approve a $3.9 billion deal to acquire spectrum from cable companies. Says the *Journal*, "Serving up TV shows and movies on mobile devices—and getting paid for it—has been a long-held dream for the wireless industry. With the introduction of next-generation high-speed wireless networks and big-screen devices like Apple Inc.'s latest iPad,

Advertiser Index

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

ubiquitous mobile video seems closer than ever." McAdams also said, "We are willing to do an à la carte approach here." This is from a company that weighs (and bills) every bit, minute and text. Billing for à la carte "content" will be a piece of cake.

While the deal as it's laid out still will require users to have a cable TV subscription, it's also clear that moving TV onto mobile devices is the long-term strategy here, not only for Verizon, but for the other major carriers as well. The setting for this game isn't the open marketplace, where the infrastructure that matters is the Internet, and where PCs and servers can still operate in relative freedom. This game happens in a regulatory space where the biggest weapons are licenses, patents, money and lobbying clout.

The stakes are very high. For example, it's clear that Google paid $12.5 billion to secure Motorola's patent portfolio, not because Google wanted to compete with all the other Android OEMs. Apple, Google, Samsung and others have been caught up in various patent disputes over mobile stuff, and Microsoft recently bought AOL's patents (including many inherited from Netscape) for $1.1 billion. Living in the patented world is not a Linux-lover's fantasy. But the patented world is a better description of the mobile data environment than of the old

home- and office-based Internet.

PCs and the Internet both grew to ubiquity thanks in large measure to the absence of regulatory, licensing and intellectual property burdens. Thanks to the same absent burdens, Linux experts today are in far more demand than the market can supply.

Yet, as smartphones morph into smartTVs, the openness of Android will take a backseat to whatever deals the carriers and the device-makers do with the "content providers" in Hollywood and New York.

In Silicon Valley, Google announced in January 2012 that it was investing $100 million in original programming for YouTube. While that's nice in the sense that it's for distribution over the Net rather than through cable or new cable-like mobile telephony systems, we're still talking about TV.

Can we move past that? Maybe we're there already, just given all the other apps a smartphone can run. But I'm not sure. Again, the smartphone business is still very young. If smartphones evolve into smartTVs that also happen to run a lot of apps, is that a good thing?

Whether or not it is, I don't think we should settle for it.■

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