LINUX™

JOURNAL

Since 1994: T

DEVELOP CUSTOM
CHANNELS FOR THE ROKU

NETWORK
TROUBLESHOOTING TIPS

L2Ork

WORLD'S FIRST LINUX LAPTOP ORCHESTRA

HOW-TO:
THE HANDBRAKE
VIDEO CONVERSION
PROGRAM

COMPARING
FIVE LINUX
MUSIC PLAYERS

$5.99US $5.99CAN

PLUS

MythTV vs. XBMC | **REVIEWED: Nokia N900**

QUALITY

WEB HOSTING AT THE BEST PRICE!

6 MONTHS
50% OFF
ON THE 1&1 BUSINESS PACKAGE*

"Working with startup companies on a tight budget, my clients know an online presence is needed. I use 1&1 exclusively to satisfy their requests for reliable hosting at affordable rates."

Lance Ochs, www.vacantpixels.com

Hosting Package	1&1 Business	Network Solutions Small	Go Daddy Unlimited
FREE Domains (.com, .net, .org, .info, .biz)	3 (for the life of your package)	1 (first year only)	—
Monthly Transfer Volume	Unlimited	3,000 GB	Unlimited
Mailbox Size	2,000 MB	1,000 MB	1,000 MB
Private Domain Registration	FREE	$9.00/year	—
Search Engine Submission	✓	✓	Extra charge applies
Money Back Guarantee	90 days	30 days	—
Special Offer	6 MONTHS 50% OFF	—	—
Price Per Month	~~$9.99~~ $4.99 for 6 months*	$9.96	$14.24
First Year Hosting Total (with discount)	$89.91/year	$128.52/year	$170.88/year

HURRY, SPECIAL OFFER VALID THROUGH APRIL 30, 2010!

99.9% Uptime!

MONEY BACK GUARANTEE

90-Day Money Back Guarantee!

24/7 Toll Free Support

More special offers are available online. For details, visit www.1and1.com

1&1 ®

MEMBER OF
united internet

Call **1-877-GO-1AND1**

Visit us now **www.1and1.com**

CONTENTS MAY 2010
Issue 193

FEATURES

ENTERTAINMENT

Cover photo by Jim Stroup, courtesy Virginia Tech.

CONTENTS
MAY 2010
Issue 193

46 NOKIA N900

42 GBRAINY

58 MYTHTV AND XBMC

64 HANDBRAKE

RAILSCONF JUNE7–10 2010
BALTIMORE, MARYLAND

▶ REGISTER NOW & SAVE 15%
Use discount code **rc10ljr**

co-presented by **O'REILLY** RubyCentral

Entertainment, Open-Source Style

SHAWN POWERS

Anyone who ever has been to a Linux or open-source conference knows that Linux geeks know how to have fun. Whether it's a late-night Atari 2600 fragfest (Penguicon 2009) or the SourceForge parties at OSCON (pretty much every year, from what I can tell), Linux is not just for work. We figured *Linux Journal* shouldn't just be for work either, and with summer right around the corner, we decided to dedicate this month to entertainment. Oh, and if you are reading this in the southern hemisphere? Well, summer is right around the corner for you too—it's just a larger corner!

Reuven M. Lerner starts out the issue with an article on MongoDB. I suspect he chose the topic because MongoDB is an entertaining name, but whatever Reuven's motivation, MongoDB is an interesting non-relational database. This month, he shows us how to install and work with Mongo. Next month, we'll get down and dirty with it. Dave Taylor, on the other hand, gets down and dirty this month with HTML forms and shell scripts. If that sounds like an odd mix, be sure to check out his column, because Dave is doing some cool things with seemingly mundane HTML forms.

If you're like me, entertainment generally means either reading a book or watching TV. When you're a Linux user, watching TV probably is a bit more geeky than it is for the rest of the world, and this issue, we have a lot of toys to cover. We start with Dirk Elmendorf's Economy Size Geek column. I've talked about the Roku device several times in past issues, and although it's unlikely my pleading is what got the company to open up development for its product, for whatever reason, Roku decided it was time. Dirk shows us how to create custom channels for the Roku and explains how to install them for your Roku-riffic enjoyment. If the Roku isn't up your alley or, at the very least, isn't in your entertainment center, perhaps Michael J. Hammel's article comparing MythTV and XBMC is more your cup of tea. Both programs are designed for consuming multimedia, and both programs are amazingly awesome. Michael does the heavy lifting for us, and he shows the ins and outs of these highly functional, if not slightly different, programs.

In order to enjoy a video on your computer or living-room entertainment center, it's important to get that video in a format that is playable. Anthony Dean shows us Handbrake this month, which does a great job of converting DVDs into formats playable on pretty much every device on the planet. I use Handbrake to convert DVDs so I can watch them on my Nokia N900, which obviously doesn't have a DVD player of its own. Although Handbrake will convert videos for the N900, playing movies certainly is not all the little handheld computer is capable of doing. Kyle Rankin has a review this month of his N900, and you'll want to check it out.

It's possible television really isn't your idea of entertainment. We totally understand that. Whether you just want to play music, or you want an entire orchestra in your laptop, we've got you covered. Dave Phillips looks at L20rk, which is much more than a bunch of MIDI files playing together like an orchestra. Maybe that's more than you're up for, and you just want to listen to music on your computer. Sometimes the hardest part is choosing which program to use for the playing. Windows and Macintosh users generally choose between WMP and iTunes, but in Linux, we have a wide variety of players. This month, Bruce Byfield breaks down five of the more common ones for us. Even if you are a die-hard XMMS fan, you'll want to check out Bruce's article comparing Amarok, Banshee, Exaile, Rhythmbox and Songbird.

Finally, we realize many of you are cyborgs and have little interest in human entertainment. Or, perhaps you're just the type of person who prefers to secure your network for fun. Well, we care about you too, and Mick Bauer and Kyle Rankin will keep you entertained this month with the continuation of their articles on OpenVPN and network troubleshooting. When you add the product reviews, UpFront articles, New Products and Doc Searls' look at distribution models, we have an issue that is certain to please. Hopefully, it will even entertain. As for me, I'm getting ready for Penguicon 2010. Although I don't think we'll have another Atari 2600 battle, I'm sure it will be great. With a bunch of Linux geeks, it's hard not to have fun!■

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.

Ruby and More Than a Single Database Table

This message is for *LJ*'s At the Forge columnist Reuven Lerner: I've read both the Ruby beginning books and looked all over the place, and I've not been able to find *anything* that takes you from a nice beginner setup with a single database table to something real—with a "well-designed" database—which means *lots* of tables. I know there's an official name for it, but I don't remember it—basically, no data is duplicated anywhere, you just have "pointers" (IDs) everywhere.

The reason I care is that I'm trying to do just that—the application is an on-line user-based help system (which doesn't even come *close* to describing it). The short version is that I've got user tables, user e-mail tables, other sorts of user identification (which can be one entry or many more), lists of things users are interested in (like hobbies), when they last verified their e-mail addresses, and on and on. Once I get past a single table though, I can find no help *anywhere*.

Please, point me to a book or *something* that goes beyond a one or two table database into the "real world". This is a "spare-time" project for a nonpaying hobby. I've been fooling with this now for more than ten years

(I started with Perl, but gave up on Perl once I had all the database entry done and had to start working on the database searching). Help!

--
Rusty

Reuven M. Lerner replies: Unfortunately, you're right. Most tutorials (including my own!) have one table, or maybe two, and don't go much beyond that. It's hard enough to write something understandable that fits into a normal-sized article length. Several tables would require even more time and space, which would make things even more complicated.

One solution is just to extrapolate a bit from those examples. An association between two tables works the same way between any pair of tables. Indeed, associations only exist between two tables, so you just need to repeat the association between however many tables you have.

But I believe, based on what you've written here, that you're looking for something more concrete—something you can really sink your teeth into to understand the techniques.

One of my favorite Rails books is Enterprise Rails *written by Dan Chak and published by O'Reilly. His examples might be too much in the other direction, using a variety of advanced techniques that might be overkill for your simple application or for simply seeing how many tables can fit together. But, several of the examples involve multiple tables joined in different ways to demonstrate a variety of Rails techniques.*

However, if you're really looking to understand how a number of tables might fit together, look at one of the Rails-based open-source applications available on the Internet. There are a number of social-networking platforms (such as Insoshi and LovedByLess), at least one e-commerce system (Spree), and at least one content-management system (Radiant). You can download, explore and try to understand the code. In your particular case, it sounds like the most interesting part of these

applications will be the models and the associations among them, but there are lots of other parts to a Rails application, and looking at these open-source applications can help you better understand those too. I hope this is helpful! Please let me know if you have any further questions.

Xen Correction

I have been a reader and subscriber to *Linux Journal* for a number of years. Almost every month I learn something new that can be applied immediately. With the help of the excellent *Linux Journal* articles over the years, I have had the opportunity to install database servers, backup servers, network monitoring systems and PBX systems.

This month, I was trying out Xen server with the intention of installing sogo. As I was preparing to deploy my first appliance, I followed the commands on page 74 of the January 2010 issue ["Simple Virtual Appliances with Linux and Xen" by Matthew Hoskins] and promptly wiped out my previous work. The tar command should read:

```
tar -cvzf appliance-base.img.tar.gz
➥appliance-base.img appliance-base.cfg
```

instead of:

```
tar -cvzf appliance-base.img appliance-base.cfg
```

Keep up the good work.

--
Rob Underwood

Matthew Hoskins replies: Rob, my sincerest apologies. I don't know how that typo crept in there. You are correct. Thanks for your feedback.

Dave Taylor's Trap

As a bash teacher at Marseille University, I like to read Dave Taylor's columns. In the February 2010 issue, on page 24, see the line that says `trap "..." 0 1 9 15`. If you try `trap -l` to get the list of UNIX signals, you'll see that 0 is not a signal (but has special meaning for kill: `kill -0 pid` succeeds if pid exists, without sending any signal). Moreover, 9 stands

for SIGKILL, which is not "trapable". Finally, a reasonable choice is: 1 2 3 15.

--
E. Thiel

Dave Taylor replies: *Sacré bleu! You're right, there is no* `trap 0` *signal to catch, and you can't catch SIGKILL. Thanks!*

Always Innovating Touch Book

Daniel Bartholomew's review of the Always Innovating Touch Book in the February 2010 issue mentioned that the software wasn't very good. I was wondering if there was a way to install Ubuntu or Windows XP instead?

--
jeff

Daniel Bartholomew replies: *The Touch Book is built around an ARM Cortex-A8 CPU from Texas Instruments. Because of this, your choices are limited. Windows, for example, does not have an ARM-compatible version. There are some choices available though. Ubuntu has an ARM port, and both Android and Chrome OS run on ARM processors. Also, other Linux distributions run on ARM processors that probably could be made to run on the Touch Book. The Touch Book Wiki has the best information on the various distributions you can run on the Touch Book:* **www.alwaysinnovating.com/wiki**.

Always Innovating also seems to have recognized that its Linux OS is not the greatest, because the latest version of the Touch Book OS (at the time of this writing, it's version 2010-01.b) has Ubuntu and Android included as boot-time options with Chrome OS promised in a future version.

The Ubuntu boot option boots you into a vanilla Ubuntu Xfce desktop environment. In my limited testing, it appears to work well enough, but I wish they would have used the Ubuntu Netbook Remix or the MID edition, as what you get isn't optimized for the touchscreen, and there doesn't appear to be an on-screen keyboard either. The Android boot option isn't fully functional yet. It boots, and you get to the desktop, but you can't do much else. For example, the two hardware buttons aren't mapped to any of the standard Android hardware buttons (home, menu and

back), and there doesn't appear to be anything set up to emulate them, which makes Android unusable for now.

The default Touch Book OS has improved during the past few months, but it still has too many issues for me to recommend it, unless you like getting up close and personal with your hardware and software.

Troubleshooting Clarification

Kyle Rankin is usually spot on with his Hack and / column, but he may have confused readers with his explanation of CPU load and the output of the `w/uptime` command on Linux [March 2010 issue, "Linux Troubleshooting, Part I: High Loads"]. Contrary to what he says, `w` *does not* show the number of processes waiting for the CPU to become available.

On Linux, it includes both those ready to run and those waiting for any type of I/O. Later in the article, he talks about CPU and I/O bound load situations and is correct on how they can be monitored. It's just a bad summary that might confuse folks. That is why you can have a responsive system even though `w` reports a load of 40.

I used to administer Sun OS/Solaris servers, and on those, `w` output was genuinely the number of processes ready to use the CPU. I got confused myself when I started working on Linux—the semantics of the command were different.

--
John

Kyle Rankin replies: *Thanks for the clarification! In trying to explain the idea of load in a simple way, I definitely left out the more complete definition. Here it is from the uptime man page: "System load average is the average number of processes that are either in a runnable or uninterruptable state. A process in a runnable state is either using the CPU or waiting to use the CPU. A process in an uninterruptable state is waiting for some I/O access, eg waiting for disk."*

Re: Security Tip

In the March 2010 issue, Paul wrote a very interesting letter about filling the

/etc/hosts file with IPs of important DNS names to avoid any man-in-the-middle spoofing or phishing attacks that could be used on free/public LANs. I use OpenDNS for all my computers, mainly because they offer a faster service than my ISP (in the UK) as well as running an anti-phishing database service to protect the rest of the family.

I was wondering if setting the computer to use OpenDNS instead of the default (and the possibly corrupt) DNS servers issued by the router/DHCP server is as safe as inserting IPs into the hosts file?

Is it possible for a compromised DNS server (say, inside a router) to intercept DNS queries destined for an external IP address and return false address data to the original node?

--
Daniel Craigie

Sadly, while a compromised DNS server wouldn't be able to poison your DNS results, a compromised router certainly could. And in that instance, it doesn't matter if we're talking about DNS or just traffic in general—an untrusted network is quite untrustworthy!

As an example, the router in your hypothetical network could just reply, pretending to be the OpenDNS servers. Because the router routes traffic, and there is no way to verify where it's coming from, it's easy to spoof. In fact, that's the reason so many people at coffee shops immediately start a VPN session, because the encryption guarantees you're connecting to the proper endpoint.

So in the end, the only way to be safe behind an untrusted router is to use some sort of VPN. Hope that helps!—Ed.

H Note?

Concerning John Knight's description of wxGuitar in his New Projects column in the March 2010 issue: H is the German name for the note we know in English as B. B to the German means our English B-flat. So wxGuitar probably comes from Germany or another country with a German-speaking musical heritage. Bach's B minor mass is, in German, in the key of H moll.

--
Charles McCarthy

[LETTERS]

Legally Using Linux?

The company I work for takes legal compliance with licenses seriously. This presents a difficulty when using free Linux distros, because the clearest statement they seem to make (if you're lucky), is that the licenses of the software in the distro are compatible. Since distros typically contain 1,000s of packages, each with its own license, for a company to check that it can comply with the terms of each (a simple example: that no package says it may not be used for commercial purposes) is quite expensive. It could tie up a legal department for weeks.

IANAL, but even Red Hat's licenses look slightly tricky. They say that the core stuff is all GPL2, but they also say that many contain many components that each have their own license. I guess Red Hat doesn't distribute OOo (now GPL3). Red Hat also has a set of 16 third-party licenses, one of which (Monotype) says you may take only a single copy for backup. So, let's hope no one has multiple level 0 system backups! Another (the "Macromedia" aka Adobe) license would be even harder to comply with. If you install the Adobe Reader on two servers (one for failover), you're in breach.

I just used Red Hat as an example, because you would think commercial distros would have the clearest statement of a user's legal obligations, but

even their licenses would require a while to check properly. (Did I mention that the "Macromedia" links to a further set of dozens of licenses for other Adobe software that may be relevant?)

This seems like a crazy situation. Surely it makes sense for the legal position of each distro to be clearly set out and summarized, for companies who want to use it in good faith. Instead, it seems every user is expected to duplicate the effort of checking for the typical problematic restrictions (such as not for commercial use or NAP).

Of course, the same situation applies to other software collections, like the wonderful Cygwin. I'm writing to you to bring the matter to the community's attention, and in the hope that the situation is not really as impossible as it seems.

--
luke

I know exactly what you mean, and our community (Linux users) knows all too painfully how licensing, even open source, can be so controversial. The GPL itself, like you mention, with multiple versions, is confusing. I'm not a lawyer either, but I fear this won't end any time soon. When companies like Adobe try to stretch their comfort level and delve into open source, they do so cautiously, so that their intellectual property isn't stolen. Quite frankly, I understand their concerns, and I applaud them for making any movement into open source at all.

Perhaps in the future, licensing will be less complex, as time proves open source is a "safe" environment to work in and still make money. Until that time, it is a complicated mess to say the least.—Ed.

PHOTO OF THE MONTH

Have a photo you'd like to share with *LJ* readers? Send your submission to publisher@linuxjournal.com. If we run yours in the magazine, we'll send you a free T-shirt.

Since there were no moose, I decided to catch up on my reading. Submitted by Matt Vermette.

diff -u
WHAT'S NEW IN KERNEL DEVELOPMENT

Last year, I wrote about **Mathieu Desnoyers**' efforts to re-license some **userspace read/copy/update** (RCU) code from the **GPL** to the **LGPL**. At the time, Mathieu was going through the laborious process of getting permission from all the contributors, but **IBM** held a patent on some of the ideas in the code and had licensed the patent only for use in GPL software. Subsequently, Mathieu wrote to me to give an update on the situation. Apparently, IBM graciously extended its patent license to include the LGPL. And, regarding the remaining kernel contributors, such as **Ingo Molnar**, who didn't want to release their code under the LGPL, Mathieu was able either to redo the work of those contributions himself or to piece together those bits successfully into a separate GPL-only file, while the lion's share of the code now also could be released as LGPL and the headers under a **BSD**-style license.

This is one of the few cases of a relatively large number of contributors being sought out and asked permission to re-license their code. As you can see, Mathieu had to resort to some workarounds in a couple cases, such as redoing a contributor's work himself, and his experience may pave the way for other re-licensing efforts. I know several folks have expressed an interest in re-licensing the entire Linux kernel under the **Open Software License**, and **Linus Torvalds** has said he might prefer that license over the GPL v2 if he had the choice. But, the Linux kernel is most likely way too big ever to get permission from enough contributors to re-license.

By the way, anyone who wants to correct or augment anything I say here each month is welcome to contact me at zacharyb@gmail.com, as Mathieu did with his update.

A recent effort to enable **devtmpfs** by default on all kernel compiles has met with some opposition. **Kay Sievers** pointed out that all major distributions include it in their upcoming releases. Although that may be true, **Alan Cox** noted that the current releases of those same distributions—that is, the releases in use by nearly everyone—don't include it. At the very least though, it does seem that devtmpfs no longer will be marked "experimental". It's a bona fide feature, at this point, and probably in the relatively near future when the standard distributions really do ship with it, the kernel folks will enable it by default, as Kay recommends.

The list of stable trees that will continue to be supported is being updated. **Greg Kroah-Hartman** announced that the **2.6.27-stable tree** will be actively maintained by him until sometime around the middle of 2010. At that point, it's expected that **Willy Tarreau** will take it over from him, but through that time, patches will be fewer and fewer. Instead, Greg said that the **2.6.32-stable tree** will be the next long-term stable tree, and he expects to maintain it for another two or three years. One of the primary reasons for keeping this kernel alive is that the major distributions went with that version, so it made sense to provide the distribution maintainers with clean upgrade paths. But, Greg also made it clear that this would have to be a two-way street— he'd maintain the 2.6.32-stable kernel just as long as the distribution maintainers kept feeding him fixes. There's no need to wait until 2.6.27 is put to bed either. As Willy said, everyone should feel free to start sending in fixes for the 2.6.32 tree right away.

—ZACK BROWN

Vendors: Let Us Do Your Dirty Work!

It wasn't too long ago that all geeks worth their salt had full-blown computers in their living rooms connected to their TVs, running software packages like MythTV, GeeXBoX, XBMC or something similar. Although those packages still are great ways to consume media, they're far from the only ways to do it. Now you can buy embedded media players off the shelf. Popcorn Hour, WDTV Live, Roku, ASUS O!Play and dozens of other embedded devices (most if not all running Linux) will plug in to a television and play video, audio, photos and so on.

The only problem is with customization. Although vendors certainly win when it comes to hardware, their software often leaves a lot to be desired. Hopefully, hardware manufacturers will take what they've learned in the computer world and transition to the set-top box world. Spend time on the hardware and let someone else do the software! Imagine an off-the-shelf device that supplied HDMI, infrared remote, onboard Flash storage and enough flexibility to install your own front end. Now, imagine that device was less than $100. With the Roku SD costing less than $80, it's certainly not a pipe dream.

So vendors, are you listening? We just want to make your life easier! You supply us with awesome hardware, and we'll supply you with awesome software. And, it will be open source—no licensing required.

—SHAWN POWERS

They Said It

If computers get too powerful, we can organize them into a committee. That will do them in.
—Bradley's Bromide

The Internet is the Viagra of big business.
—Jack Welch, Chairman and CEO, General Electric

At some point we must have faith in the intelligence of the end user.
—Anonymous

There are two major products that came out of Berkeley: LSD and UNIX. We don't believe this to be a coincidence.
—Jeremy S. Anderson

One machine can do the work of fifty ordinary men. No machine can do the work of one extraordinary man.
—Elbert Hubbard

Any teacher that can be replaced by a computer, deserves to be.
—David Thornburg

Those parts of the system that you can hit with a hammer are called hardware; those program instructions that you can only curse at are called software.
—Anonymous

NON-LINUX FOSS

PuTTY Configuration

For many Linux power users lost in a Windows world, the most glaring omission is the lack of an SSH client. Of course, there's Cygwin and all the good things that come with it in addition to an SSH client (and server), but if you're looking for something a little more lightweight or if you want a GUI SSH client, PuTTY probably is the tool you need. In addition to SSH, PuTTY also supports Telnet and Rlogin.

PuTTY supports most of the features Linux users are used to having: X11 forwarding (you'll need a separate X server for this), port forwarding, tunneling and so on. PuTTY supports SSH-1 and SSH-2 and has support for public key authentication. PuTTY comes with the pscp and psftp companion tools for command-line access to SCP and SFTP.

Because it's a GUI application, PuTTY's settings, including those for different hosts, are configured and stored via the GUI interface. Only when you've actually made a connection to a host do you interact with a terminal window.

PuTTY is mature program. The first release listed in the changelog, 0.45, is from 1999. The latest is 0.60, released in 2007. PuTTY is available from **www.chiark.greenend.org.uk/~sgtatham/putty**.

—MITCH FRAZIER

Participate at LinuxJournal.com

Featured *Linux Journal* Staff

One of the best things about the new LinuxJournal.com is the ability to interact with the *Linux Journal* staff and your fellow readers. Visit **www.linuxjournal.com/participate** to search for other readers or find *Linux Journal* editors and make connections. You'll also find forums, community events and our IRC channel. There are many ways to be a part of our growing community, so check them out and jump right in.

When you visit authors' profiles, you'll see a list of all their articles as well as what they've been up to on LinuxJournal.com. You'll also see their favorite articles, and if you log in, you can leave them a message. We hope you'll make some connections and think of us as your virtual LUG. I look forward to connecting with all of you!

—KATHERINE DRUCKMAN

LJ Index
May 2010

1. Millions of hits for a search for "Google" on Google: **1,980**

2. Millions of hits for "Yahoo" on Google: **1,590**

3. Millions of hits for "Microsoft Bing" on Google: **22.1**

4. Millions of hits for "Google" on Yahoo: **5,790**

5. Millions of hits for "Yahoo" on Yahoo: **10,700**

6. Millions of hits for "Microsoft Bing" on Yahoo: **88.4**

7. Millions of hits for "Google" on Bing: **177**

8. Millions of hits for "Yahoo" on Bing: **270**

9. Millions of hits for "Microsoft Bing" on Bing: **3.9**

10. Millions of hits for "Linux" on Google: **294**

11. Millions of hits for "Linux" on Yahoo: **1,550**

12. Millions of hits for "Linux" on Bing: **253**

13. Millions of hits for "Windows" on Google: **831**

14. Millions of hits for "Windows" on Yahoo: **3,660**

15. Millions of hits for "Windows" on Bing: **384**

16. Rank of Google in results for "search engine" on Google: **5**

17. Rank of Yahoo in results for "search engine" on Yahoo: **1**

18. Rank of Bing in results for "search engine" on Bing: **100**

19. Rank of Dogpile in results for "search engine" on Google: **1**

20. Rank of Dogpile in results for "search engine" on Bing: **1**

Sources: *1–3, 10, 13, 16, 19: Google | 4–6, 11, 14, 17: Yahoo | 7–9, 12, 15, 18, 20: Bing*

LISTENING ON THE COMMAND LINE

Linux has tons of programs for listening to audio, including Totem, Rhythmbox, MPlayer and VLC. Each of those examples run in an X11 environment. What if you want to listen to music on the command line? Well, let's take a look. Before we do though, note that this article assumes you have a working audio subsystem on your machine.

Most distributions now use PulseAudio as the audio server, which provides a standard wrapper around the actual audio hardware. This way, software simply needs to talk to the PulseAudio server and doesn't need to worry about the messy details of how to talk to each soundcard out there. The PulseAudio system also includes utilities you can use on the command line. To simply play an audio file, do this:

```
paplay --volume=32768 example.mp3
```

This plays the audio file example.mp3 at 50% volume (volume is set between 0 and 65536, or it can be silent or 100%). The paplay utility can play any audio format supported by the libsndfile library, so you should be able to play most audio files you encounter. If you don't have any audio files, use the included parec utility. This program grabs raw audio data from your audio card's input and dumps it to standard output; you can pipe this off to a file to save for later. To play back this raw data, use the included pacat utility. It takes raw audio data and dumps it out to the audio card's output speaker.

What if you want to listen to a whole list of audio files? Several choices are available for handling entire playlists from the command line, such as cplay and moc. Both programs give you a file list on startup, and from there, you can play individual files or construct playlists. Both programs use shortcut keys to create, edit and otherwise manage your playlists.

You also can make your computer talk to you from the command line. Several utilities can give your computer a voice, including Recite, Festival and eSpeak. The simplest is Recite. It takes text from standard input and outputs audio to the speakers. There aren't many options available. To make a text file speak, execute:

```
recite <example.txt
```

You also simply can execute `recite`, then start typing. Remember that Recite won't see the end of your typing until you press Ctrl-D to mark the end of input. Then, it speaks the entire text you just typed.

Festival provides more options. To get a basic output of text to audio, execute:

```
festival --tts example.txt
```

You also can use other languages. The default is English, but for Spanish, Russian or Welsh, use the command-line option `--language <string>`. Festival also uses Scheme as a scripting language, which opens it up to a huge amount of modification. Spend some time reading the manual at **www.cstr.ac.uk/projects/festival/manual** for more information.

The last utility is eSpeak, which has several other available command-line options. You can set the amplitude with `-a <integer>` (0–20;

the default is 10), the pitch with `-p <integer>` (0–99; the default is 50) and speed with `-s <integer>` (in words per minute; the default is 160). Several voices are available; find out which ones are installed on your system by using the option `--voices`. Once you select one, use the option `-v <voicename>` to use it. Then, dump it out to a .wav file with the `-w <wave file name>` option. A full command line looks like this:

```
espeak -v en-scottish -w example.wav -f example.txt
```

You can play it with:

```
paplay example.wav
```

Now your computer will speak in a Scottish accent, sort of. Speech synthesis still isn't perfect.

Now that you have all these audio files, you may want to do some processing on them, and sox is a useful tool for doing just that. One utility included in the package is called soxi. It gives you file information about your audio files. In its most basic form, you can use sox to change the file format of an audio file simply by running:

```
sox example.wav example.au
```

sox uses file extensions to figure out that you are intending to convert the file example.wav (in wav format) to example.au (in the Sun AU format). You also can do processing on the audio through command-line options. Set the number of bits per sample with `-b <bits>`. Set the number of channels with `-c <channels>`. Setting it to mono would be `-c 1`, and stereo would be `-c 2`. You can set the sample rate, in Hertz, with `-r <rate>`. Additionally, many options apply filters to audio files. Read the manual page for more information on sox. Here's an example command line:

```
sox example.wav -b 8 -c 1 -r 8000 example.au
```

The above takes an input file called example.wav and converts it to 8 bits per sample, mono (or 1 channel), with a sample rate of 8,000Hz and writes it out to a file with an AU format.

The sox package also contains two other utilities, play and rec, which provide another way of playing audio files and recording audio to a file. sox also provides the full spectrum of processing and filters.

I chose the above example so I could show you one last interesting trick. On Linux systems, you can cat this output file (with this specific file format) directly to the device file /dev/audio. This dumps the output directly to the soundcard. So, if you want to be sneaky, you can convert some files to the AU file format with the sox command above, copy them over to your target machine, and cat them to /dev/audio when you want to make a nuisance of yourself. Now you can enjoy your music and play with audio files, without the overhead of a GUI application.

—JOEY BERNARD

cplay
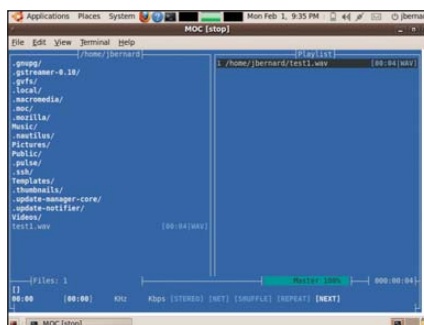
moc

Send E-mail with Telnet

Poor telnet has gotten a bad rap over the years due to the fact that if you use it for remote logins, your password shows up in plain text. However, even though you should use SSH for remote sessions, you shouldn't completely throw away telnet. After all, telnet is just one way to connect to a remote port and send text commands to it. When you start poking around at a number of protocols like HTTP and SMTP, you'll find that if you know the right commands, you can telnet into those ports and act like a more sophisticated client.

One way I use telnet is to test whether a mail server works. Sure, I could use an e-mail client, but then I'd have to reconfigure it. Plus, sending an e-mail with telnet is a nice geeky party trick (well, maybe the parties I go to), especially if you forge your FROM address.

First, use telnet to connect to port 25 on your mail server:

```
$ telnet mail.example.net 25
Trying 123.234.123.234...
Connected to mail.example.net.
Escape character is '^]'.
220 mail.example.net ESMTP Postfix
```

Next, use the HELO command to tell the server what domain you are coming from. It will respond with its name:

```
HELO microsoft.com
250 mail.example.net
```

After that, use the MAIL FROM: command to tell the server the e-mail address this e-mail is coming from. The fun part here is that you can make the FROM address appear to be from anyone. If the mail server accepts it, it will reply with 250 Ok:

```
MAIL FROM: bill.gates@microsoft.com
250 Ok
```

We're in! Okay, now use the RCPT TO: command to tell the server to whom you are sending the e-mail:

```
RCPT TO: lj@greenfly.net
250 Ok
```

Finally, let's type in the body of the e-mail. Type DATA and press Enter. The mail server will respond with instructions to end your e-mail body with an empty line containing a single dot. After this point, I typed in a Subject header and the rest of the body of my e-mail, but you can add as many headers as you want here. When you're done with the message, type a period on a line by itself:

```
DATA
354 End data with <CR><LF>.<CR><LF>
Subject: Linux is awesome
Hi,

It's Bill. I just wanted to let you know
Linux is awesome.

Sincerely,
Bill Gates
.
250 Ok: queued as 12BDBE6FEE9
```

When you're finished with your e-mail session, type quit to exit:

```
quit
221 Bye
Connection closed by foreign host.
```

—KYLE RANKIN

Let's Party Like It's 1999!

Studies show taking frequent breaks from work increases productivity, makes for a more pleasant work environment and reduces stress. Although there probably are healthier ways to spend your occasional breaks from work, game emulators certainly provide a fun way to relax. Assuming you're tech-savvy enough to land a job that allows you to use Linux, and that you're geeky enough to desire a way to play old Nintendo games, I think it's safe to assume you will be able to procure ROM images for those games you legally own—you know, those games sitting in a box in your parents' basement.

With the addition of emulators like ROCKnes, iNes, zsnes and their ilk, you can spend those 15-hour, er, -minute, breaks playing classics like *Super Mario Brothers*, *F-Zero*, *Zelda* and so on. If you can land a USB game controller, it will make the experience that much more fun. Let's not fool ourselves, although *World of Warcraft* might be addictive, it doesn't hold a candle to those 30-hour sessions of *Dragon Warrior* we used to play over the weekend instead of doing homework. Have fun, and try not to get fired!
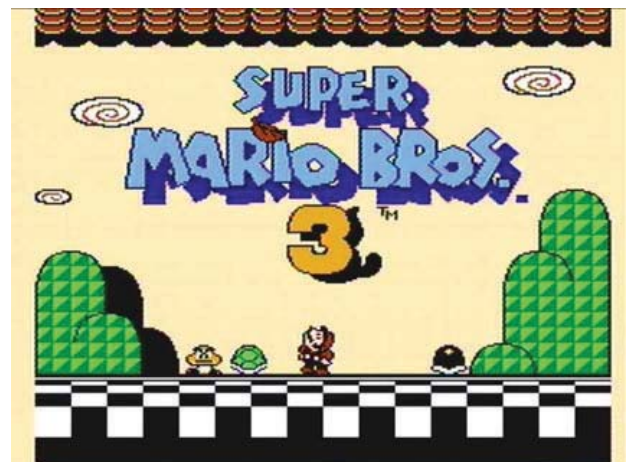
—SHAWN POWERS

MongoDB

A look at one of the best-known contenders in the non-relational database space.

REUVEN M. LERNER

Lately I've been teaching programming courses in both Python and Ruby, often to seasoned developers used to C++ and Java. Inevitably, the fact that Python and Ruby are dynamically typed languages, allowing any variable to contain any type of value, catches these students by surprise. They often are shocked to find that a given variable can, at any point in the program, be assigned to contain an integer, a string or an instance of an object, without any constraints. They wonder how it is that anyone could (or would) use such a language, given the possibility for runtime type errors. One of my jobs, as the instructor of this course, is to convince them that it is possible to work in such a language, but that doing so might require more adherence to conventions than they are used to.

So, it's ironic that during the last few months, as I have begun to experiment with non-relational databases, that I have found myself experiencing something akin to my students' shock. My long-standing beliefs about data integrity and what

My long-standing beliefs about data integrity and what constitutes a reliable database have gone through a bit of a shake-up.

constitutes a reliable database have gone through a bit of a shake-up. I'm still a bit wary of these non-relational (or NoSQL) databases, and I'm far from convinced that the time has come to throw out SQL and the relational model in favor of something that is often easier to work with.

I do think, as I outlined in last month's column, that these databases offer a type of storage and retrieval that often is a more natural fit for many data-storage requirements. And, just as memcached offered an alternative storage system that complemented relational databases rather than replacing them, so too can these non-relational databases perform many useful functions that would be difficult with a relational database.

One of the best-known contenders in the non-relational database space is MongoDB. MongoDB is an open-source project, sponsored by New

York-based 10gen (which intends to make money from licensing and support fees). It is written in C++, and there are drivers for all popular modern libraries. The software is licensed under the Affero GNU General Public License, which means if you modify the MongoDB source, and if those modifications are available on a publicly accessible Web site, you must distribute the source to your modifications. This is different from the standard GPL, which does not require that you divulge the source code to server-side applications with which people interact via a browser or other Internet client.

MongoDB has gained a large number of adherents because of its combination of features. It is easy to work with from a variety of languages, is extremely fast (written in C++), is actively supported by both a company and a large community and has proven itself to be stable in many situations and under high-stress conditions. It also includes a number of features for indexing and scaling that make it attractive.

MongoDB, like several of its competitors, describes itself as a document database. This does not mean it is a filesystem meant to store documents, but rather that it replaces the model of tables, rows and columns with that of "documents" consisting of one or more name-value pairs. I find it easier to think of documents as hash tables (or Python dictionaries), in which the keys are strings and the values can be just about anything. Each of these documents exists in a collection, and you can have one or more collections.

In many ways, you can think of MongoDB as an object database, because it allows you to store and retrieve items as objects, rather than force them into two-dimensional tables. However, this object database stores only basic object types—numbers, strings, lists and hashes, for example. Fortunately, these types can store a wide variety of data, flexibly and reliably, so this is not much of a concern.

Downloading and Installing

To download MongoDB, go to **mongodb.org**, and retrieve the version appropriate for your system. For my server running Ubuntu 8.10, I retrieved the 32-bit version of MongoDB 1.2.2. There is an option to retrieve a statically linked version, but the site itself indicates that this is a fallback, in case the dynamically linked version fails.

After unpacking the MongoDB server, create a directory in which it can store its data. By default, this is /data/db, which you can create with:

```
mkdir -p /data/db
```

Start the MongoDB server process with:

```
./bin/mongod
```

Now that you have a server running, you need to create a database. However, this step is unnecessary. If you try to connect to a database that has not yet been defined, MongoDB creates it for you. I tend to do most of my MongoDB work in Ruby, so I downloaded and installed the driver for Ruby from GitHub and started up the interactive Ruby interpreter, irb. Then, I typed:

```
irb(main):001:0> require 'rubygems'
irb(main):002:0> require 'mongo'
```

With the MongDB driver loaded, I was able to connect to the already-running server, creating an "atf" database:

```
irb(main):005:0> db = Mongo::Connection.new.db("atf")
```

After this, db is an instance of the Mongo::DB class, representing a MongoDB database. Each database may contain any number of collections, analogous to tables in a relational database. By default, this example database contains no collections, as you can see with this small snippet of code:

```
irb(main):008:0> db.collection_names.each { |name| puts name }
=> [ ]
```

The return value of an empty list shows that the database is currently empty.

You can create a new collection by invoking the collection method on your database connection:

```
irb(main):012:0> c = db.collection("stuff")
```

Once you have created your collection, you also can see that MongoDB has silently created a second collection, named system.indexes, used for indexing the contents:

```
irb(main):032:0> db.collection_names
=> ["stuff", "system.indexes"]
```

Because MongoDB is a schema-less database, you can begin to store items to your collection immediately, without defining its columns or data types. In practice, this means you can store

hashes with any keys and values that you choose. For example, you can use the insert method to add a new item to your collection:

```
irb(main):017:0> c.insert({:a => 1, :b => 2})
=> 4b6fe8983c1c7d6a6a000001
```

The return value is the unique ID for this document (or object) that has just been stored. You can ask the collection to show what you have stored by invoking its find_one method:

```
irb(main):021:0> c.find_one
=> {"_id"=>4b6fe8983c1c7d6a6a000001, "a"=>1, "b"=>2}
```

Notice that two things have happened here. First, the keys have been turned from Ruby symbols into strings. Indeed, MongoDB requires that all keys be strings; because symbols are used so pervasively in the Ruby world for hash keys, they are translated into strings silently if you use them.

Second, you can see that another key, named _id, has been added to the document, and its value matches the return value that you received with your first insert.

You can ask the collection to tell how many documents it contains with the count method:

```
irb(main):026:0> c.count
=> 1
```

As you might expect, you can store and retrieve data using any number of different languages. Although you are likely to work in a single language, MongoDB (like relational databases) doesn't care what language you use and lets you mix and match them freely.

In the above examples, I used Ruby to store data. I should be able to retrieve this data using Python, as follows:

```
>>> import pymongo
>>> from pymongo import Connection
>>> connection = Connection()
>>> db = connection.atf
>>> db.collection_names()
    [u'stuff', u'system.indexes']
>>> c = db.stuff

>>> c
    Collection(Database(Connection('localhost', 27017), u'atf'),
➥u'stuff')

>>> c.find_one()
    {u'a': 1, u'_id': ObjectId('4b6fe8983c1c7d6a6a000001'), u'b': 2}
```

The only surprises here are probably that the

> **If you are used to a functional style of programming, in which you chain a number of methods to one another, this style easily will lend itself to working with MongoDB.**

strings are all stored as Unicode, represented with the u' ' syntax in Python 2.6 (which I am using here). Also, the document ID, with the key of _id, still is there, but is an object, rather than a string.

You also can see that the MongoDB developers have gone to great efforts to keep the APIs similar across different languages. This means if you work in more than one language, you likely will be able to depend on similar (or identical) method names to perform the same task.

Queries
The find_one method, as you have seen, returns a single element from a collection. A similar find method returns all of the elements using the Enumerable module, allowing you to iterate over all of the documents in a collection using each. For example, if you add another document:

```
irb(main):026:0> c.insert({'name' => 'Reuven',
                           'email_address' => 'reuven@lerner.co.il'})
=> 4b6ff0693c1c7d6ecd000001
```

you can retrieve the IDs as follows:

```
irb(main):030:0> c.find.each {|i| puts i['_id']}
4b6fe8983c1c7d6a6a000001
4b6ff0693c1c7d6ecd000001
```

Notice how you can pull out the _id column by treating the document as a hash. Indeed, if you ask Ruby to show the class of the object, rather than its ID, this suspicion is confirmed:

```
irb(main):031:0> c.find.each {|i| puts i.class}
OrderedHash
OrderedHash
```

But, perhaps you're interested only in some of the documents. By invoking find with a hash, it will return only those documents that match the contents of your hash. For example:

```
irb(main):040:0> c.find({'name' => 'Reuven'}).count
=> 1
```

If nothing matches the hash that you passed,

you will get an empty result set:

```
irb(main):041:0> c.find({'name' => 'Reuvennn'}).count
=> 0
```

You also can search for regular expressions:

```
irb(main):042:0> c.find({'name' => /eu/}).count
=> 1
```

```
irb(main):043:0> c.find({'name' => /ez/}).count
=> 0
```

By passing a hash as the value for a key, you also can modify the query, passing parameters that define MongoDB's query syntax. These query operators all begin with the dollar sign ($) and are passed as the key to a sub-hash. For example, you can retrieve all of the documents whose "name" is one of the values in a specified array, as follows:

```
irb(main):049:0> c.find({'name' =>
                        {'$in' => ['Reuven', 'Atara', 'Shikma',
➡'Amotz'] } }
                        ).count
=> 1
```

You also can sort the results by invoking the sort method on the result set, using a similar syntax:

```
irb(main):049:0> c.find({'name' =>
                        {'$in' => ['Reuven', 'Atara', 'Shikma',
➡'Amotz'] } }
                        ).sort({"name" => 1})
```

Just as you can sort a result set, you also can perform other actions on it that are analogous to several relational counterparts, such as grouping and limiting the number of results. If you are used to a functional style of programming, in which you chain a number of methods to one another, this style easily will lend itself to working with MongoDB.

Conclusion
MongoDB is causing many ripples in the open-source and database worlds because of its high performance and easy learning curve. This month, I covered the basics of installing and working with MongoDB. Next month, I'll look at some more-advanced topics, such as indexing (which makes queries execute much faster), embedding objects in one another and referencing objects across collections.■

Reuven M. Lerner is a longtime Web developer, trainer, and consultant. He is a PhD candidate in Learning Sciences at Northwestern University. Reuven lives with his wife and three children in Modi'in, Israel.

Resources

The main site for MongoDb, including source code and documentation, is at **mongodb.org**.

The Ruby driver for MongoDB is hosted at GitHub: **github.com/mongodb/mongo-ruby-driver**. The Python drivers are at **github.com/mongodb/mongo-python-driver**.

For an excellent introduction to MongoDB, including some corporate background on 10gen and how it can be used in your applications, listen to episode 105 of the "FLOSS Weekly" podcast. I found the podcast to be both entertaining and informative.

Another good introduction is from John Nunemaker, a well-known blogger in the Ruby world: **railstips.org/blog/archives/2009/06/03/what-if-a-key-value-store-mated-with-a-relational-database-system**.

A blog posting having to do with MongDB's speed, relative to that of CouchDB and MySQL is at **www.idiotsabound.com/did-i-mention-mongodb-is-fast-way-to-go-mongo**.

Finally, I continue to think that there are parallels between dynamic languages and document-oriented databases. Steve Yegge, an engineer at Google, has written about the growing popularity of dynamic languages, and I strongly recommend his presentation for an interesting perspective on the issue: **steve-yegge.blogspot.com/2008/05/dynamic-languages-strike-back.html**.

Converting HTML Forms into Complex Shell Variables

DAVE TAYLOR

Web browser? We don't need no stinkin' Web browser for submitting HTML forms, that's what the shell is for.

I know, there are a million shell scripts waiting to be written to help administer your computer, run your server and fine-tune your back end, but I'm obsessed with scripts that interact with on-line data, so that's what I'm focusing on. My last column marked the end of our Twitterbot, a simple script that listens and responds to Twitter queries. You can try it by sending an "@" message from your Twitter account to @davesbot.

This month, I thought that given the issue's Entertainment theme, it'd be fun to dig into another facet of shell scripts that interact with the Web by looking at how to emulate a complex form. The form we'll emulate? Yahoo Movies advanced search.

Start by checking out Figure 1 (it shows the form). You can see it live by going to **movies.yahoo.com/mv/advsearch** too.

We can crack open the HTML and read through the source, but I think it's more interesting to reverse-engineer it, because, like most search forms, this one uses the GET method and, therefore, exposes all of its parameters within the URL of the results page. For example, a search for the title "Strangelove", without any other tweaks, produces the URL below. Normally, this URL would be all on one line, but I've separated the URL and the parameters onto multiple lines to make them a bit easier to see:

Figure 1. Yahoo
Movies Advanced
Title Search

```
http://movies.yahoo.com/mv/search
        ?p=strangelove
        &yr=all
        &gen=all
```

```
&syn=
&syn_match=all
&type=feature
&adv=y
```

The search engine itself is at the URL shown in the first line of the listing above. The rest of the lines are parameters sent to the search engine. You can see that the search term is "p" ("p=strangelove"). You can infer the other parameters by looking at the form: yr = release decade, gen = genre, syn = synopsis keywords and so on.

Because there are so many possible values, however, we're going to have to look at the source after all. For example, those genres? Here's how Yahoo Movies breaks it down:

- act = Action/Adventure

- ada = Adaptation

- ani = Animation

- ... (lots of entries skipped for space)

- tee = Teen

- thr = Thriller

- war = War

- wes = Western

It's quite a list, really!

The question is, can we turn a form of this nature into a simple interactive shell script that will let users specify constraints on a search and pop open a Web browser with the resultant search? Of course we can!

Turning HTML into a Script

It would be cool to normalize the problem and come up with a general-purpose solution, some sort of parser

Search by Title

Title:
Decade: All
Genre: All
Synopsis Keywords:

Match all synopsis keywords

Find only movies with:
☐ Showtimes   ☐ Trailers & More
☐ Prod. Photos   ☐ Premiere Photos
☐ News/Features   ☐ Critics Reviews
☐ Prod. Notes   ☐ Film Bios

Search

that would take HTML form tags as input and produce shell script segments as output. Uh, no thanks.

Instead, with a few hacks in vi (yeah, I don't use Emacs), I have the following, as part of a usage() function:

```
usage()
{
cat << EOF
USAGE: findmovie -g genre -k keywords -nrst title
Where

    -n    only match those that have news or features

    -r    only match those with reviews

    -s    only match those that have showtimes

    -t    only match those that have trailers


and genre can be one of:

  act (Action/Adventure), ada (Adaptation), ani (Animation),

  ...

  tee (Teen), thr (Thriller), war (War) or wes (Western).
EOF

}
```

This makes life easy and pushes the trick of remembering the three-letter abbreviation for the genre onto the user. Sneaky, eh? Now, to be fair, good interface design would have me writing a more sophisticated script that lets users enter a variety of abbreviations (or the full word) and converts them into the proper Yahoo-approved abbreviation, but that's actually work, so we'll skip that too, okay?

Now, note the actual usage I've created:

```
USAGE: findmovie -g genre -k keywords -nrst title
```

This means there are a couple elements of the form that we are going to ignore in the script, including which decade the film was released and some of the more obscure conditional parameters. Still, it's enough to keep us busy.

Parsing Parameters with getopts

I've talked about the splendid getopts within shell scripts before, without which parsing the six parameters—two of which have arguments, four of which don't—would be a huge hassle. Instead, this

is straightforward. Here are the first few lines to give you the idea:

```
while getopts "g:k:nrst" arg
do
  case "$arg" in
    g) params="${params:+$params&}gen=$OPTARG" ;;
```

There's a lot to talk about here, but we have covered getopts before, and you can <cough> check the man page too, right? In a nutshell though, a letter with a trailing colon means it has a required parameter, so g and k have arguments (g:k:), while n, r, s and t do not (nrst).

The params expansion is a nifty little shell trick that's worth a special mention too. The notation ${params:+$params } expands to the value of the $params variable, plus a trailing space, if the variable already has a value. Otherwise, it's the null string. The point? To avoid leading ampersands in the URL that we're building.

Let's have a quick peek:

```
$ findmovie.sh -g war -k peace -r
finished. params = gen=war&syn=peace&revs=1
```

As we'd hope, the params variable has been expanded to reflect the specific values that the user has specified on the command line—in this case, War films that have reviews and contain the word "peace" in the synopsis.

Building the Full URL

There's a hiccup waiting to bite us with the code in its current state though. The problem is, what if the user specifies two words in the keywords value field or, worse, does so in the title field (remember, the last word or words are the title pattern, the core search for the Yahoo Movies system)?

The answer is that we need to convert spaces into symbols that are acceptable by the http system. That's easily done, fortunately:

```
params="$(echo $params | sed 's/ /+/g')"
```

It's not the most elegant solution, but it's certainly functional!

The bigger problem here is that Yahoo requires certain parameters actually be present to do a search. Choose a genre on the Web interface and click search, and you'll see that's not sufficient for it to proceed.

As a result, our base URL for searches is going to be a bit more complicated:

```
baseurl="http://movies.yahoo.com/mv/search"
baseurl="${baseurl}?yr=all&syn_match=all&"
```

Try that, and you'll find it doesn't work. Why? Because there are some hidden parameters that Yahoo has slipped into the form that are required to send to the search program. Without them, it just stops.

In fact, here's the baseurl value we need:

```
baseurl="http://movies.yahoo.com/mv/search"
baseurl="${baseurl}?yr=all&syn_match=all&adv=y&type=feature&"
```

Now, how do we put this all together? It's not so easy, because we still need to grab whatever's on the end of the invocation (the title pattern), then mask the spaces:

```
shift $(( $OPTIND - 1 ))
```

Hang on, let me explain this line before we go further. OPTIND contains the index into the positional parameters of the script, indicating the first parameter that wasn't absorbed by the getopts processing. Unfortunately, it's indexed from 1, and the options array is indexed starting at zero. The result? We have to subtract one from the value to be able to get the actual value with the $* notation:

```
params="$(echo $params | sed 's/ /+/g')"

pattern="$(echo $* | sed 's/ /+/g')"
echo URL: $baseurl${params}\&p=$pattern
```

Now, finally, armed with that, we can search for films that contain the word "love" and have reviews:

```
$ findmovie.sh -r love

URL: ...BASEURL...revs=1&p=love
```

Type that in, and you'll find it works fine, showing 80 films where "love" appears in the title and Yahoo Movies is aware of on-line reviews of the films.

Most Linuxes and other flavors of UNIX have a way that you can launch a Web browser from the command line, with the specified URL as its home. That's what we'll do:

```
echo $baseurl${params}\&p=$pattern
open -a safari "$baseurl${params}\&p=$pattern"
```

There are other things we can do now that we've converted the Yahoo advanced search form into a shell script, but we'll leave those for next month!■

Dave Taylor has been hacking shell scripts for a really long time, 30 years. He's the author of the popular *Wicked Cool Shell Scripts* and can be found on Twitter as @DaveTaylor and more generally at www.DaveTaylorOnline.com.

Linux VPNs with OpenVPN, Part IV

MICK BAUER

Use dangerous local-area networks without fear with OpenVPN.

For the past few months, I've been describing how to build a Virtual Private Network (VPN) server using OpenVPN, a free, multiplatform, TLS/SSL-based VPN dæmon. My example usage scenario involves the common "road warrior" setup where remote users connect back to a VPN server on their "home network" by establishing an encrypted VPN tunnel over the Internet.

Last month, in Part III, I finished a line-by-line walk-through of an example OpenVPN server configuration file (server.ovpn) shown here for your reference (Listing 1).

I then talked about running OpenVPN as a server process (the same executable can be run either as a

dæmon/listener or as a client process), either running in the foreground, with all log messages printed to the console:

```
bash-$ sudo openvpn --config ./server.ovpn
```

or in the background, with all log messages being written to /var/log/daemon.log:

```
bash-$ sudo openvpn --daemon --config ./server.ovpn
```

While in the early stages of getting OpenVPN working on both server and clients, you'll definitely want to run the server dæmon in the foreground, because you'll probably have to stop and restart it

---

Listing 1. Server's server.ovpn File

```
port 1194
proto udp
dev tun

ca 2.0/keys/ca.crt
cert 2.0/keys/server.crt
key 2.0/keys/server.key  # Keep this file secret
dh 2.0/keys/dh1024.pem
tls-auth 2.0/keys/ta.key 0

server 10.31.33.0 255.255.255.0
ifconfig-pool-persist ipp.txt
push "redirect-gateway def1 bypass-dhcp"

keepalive 10 120

cipher BF-CBC         # Blowfish (default)
comp-lzo
max-clients 2

user nobody
group nogroup
persist-key
persist-tun

status openvpn-status.log
verb 3
mute 20
```

Listing 2. Client's client.ovpn File

```
client
proto udp
dev tun

remote 1.2.3.4 1194

nobind

ca ca.crt
cert minion.crt
key minion.key

ns-cert-type server
tls-auth ta.key 1

cipher BF-CBC
comp-lzo

user nobody
group nogroup
persist-key
persist-tun

mute-replay-warnings

verb 3
mute 20
```

through configuration tweaks anyhow. Once every-thing's working, you can put an init-script into your server's /etc/init.d directory that starts OpenVPN in dæmon mode automatically at startup time.

OpenVPN Client Configuration

This brings us to client configuration. Listing 2 shows a sample client configuration file, client.ovpn. Let's dissect it!

First is the `client` directive. Like `server`, which we covered last time, client is actually a helper directive that, when read by the openvpn command, expands to two other directives: pull, which instructs OpenVPN to accept options pushed to it by the OpenVPN server it connects to, and tls-client, which enables TLS (SSL) encryption and tells OpenVPN to assume the role of client any time it initiates a TLS transaction.

Next comes `proto udp`, which tells OpenVPN to use UDP packets to build its VPN tunnel. This setting needs to be the same as what's specified on the server to which you wish to connect.

Next comes `dev tun`, which tells OpenVPN to encapsulate IP packets via a /dev/tun interface, rather than Ethernet frames via a /dev/tap device. I'm sticking to IP encapsulation in my examples, and besides, this setting has to be the same as on the

server to which you wish to connect.

And, to which server do you wish to connect? The one specified in the `remote` directive, which has two parameters, IP address (or hostname) and port. In Listing 2, these are set to 1.2.3.4 1194, specifically UDP port 1194. (If earlier I had set `proto` to `tcp-client`, OpenVPN would assume you mean TCP port 1194 here.)

The IP address of my example server is 1.2.3.4, which may strike you as improbable, but this address is, at least, Internet-routable. If you're going to connect to your OpenVPN server from across the Internet, you'll need to target an Internet-routable IP address. In my home setup, this is actually the address of my DSL router, which I've configured to redirect UDP 1194 connections to the same port on my OpenVPN server, whose real IP address is a non-Internet-routable 192.168.0.0 address.

After `remote` comes `nobind`, which tells OpenVPN to allow the local IP stack (your Linux kernel's TCP/IP modules) to assign a local port from which to send and receive OpenVPN packets dynamically, rather than have the OpenVPN dæmon "bind" to (listen on) a specific port like a server process would. This setting, therefore, is suitable only for VPN client systems.

Certificate/Key-Related Directives

Next, there are five directives related to helper files that OpenVPN will need to read in order to build a tunnel. `ca` specifies a file containing at least one Certificate Authority certificate, specifically, the certificate of whatever CA will have signed the OpenVPN's server certificate. If you try to connect to a server whose server certificate was *not* signed by a CA key/certificate specified here, your OpenVPN client process will terminate the connection.

`cert` specifies a file containing a client certificate for your OpenVPN client process to present to the OpenVPN server. This certificate needs to have been signed by a CA whose certificate resides in the *server's* ca file, or the server will reject connections from *you*.

In many if not most cases, the simplest way to handle these certificates is to use the same CA to create and sign both server and client certificates. In fact, if you remember Part II in this series (*LJ*, March 2010), I *already* created a client certificate and key, right after I created the server credentials.

Because this process is both important and simple, let's take a minute to review it. I'm skipping the process of setting up and creating the Certificate Authority itself, as that applies only to server setup (you should do that only once, on the server). So, assuming you've got a working CA set up on your OpenVPN server as described in Part II of this article, follow these steps to use OpenVPN's pkitool script to create a new client certificate:

1) su to root:

```
bash-$ su
```

2) Change your working directory to /etc/openvpn/2.0:

```
bash-# cd /etc/openvpn/2.0
```

3) Declare some PKI-related environment variables stored in the file vars:

```
bash-# source ./vars
```

4) Create the new certificate:

```
bash-# ./pkitool --pass minion
```

In step 4, the string `minion` is the name (the "common name", in x.509 parlance) of the host or user whose certificate you're creating. After issuing this command, you'll be prompted twice to type the certificate's passphrase.

The output of this command takes the form of three files: ./keys/minion.csr, ./keys/minion.crt and ./keys/minion.key. Only the last two are important for the purposes of this article: the new client

certificate and client key, respectively.

Of the helper files related to crypto that were created when you set up the OpenVPN server, your client certificate and key are the only two unique to the client; ca.crt and ta.key are used on the server and on all clients that connect to it. Note also that although the client certificate (minion.crt) contains no private data, the client key minion.key and the TLS authentication key ta.key both should be kept secret through local file permissions and by handling them carefully.

For example, you should never e-mail any client key or TA key in clear text (note that using an https:// URL for Webmail access doesn't count as message encryption). You should use S/MIME or PGP e-mail encryption if you need to mail keys to users.

If you use a USB drive or other physical media to distribute keys, you should either deliver it in person or use a trusted courier to deliver it, and users should be instructed either to destroy or erase the media after installing their keys, or keep the media under lock and key. Although having a passphrase-protected client key *should* make it hard for an attacker to use an intercepted key file, it's no guarantee! Under no circumstances should you issue blank-passphrase client certificates for any VPN scenario.

Speaking of the client key's passphrase, you also should take care in how you transmit that passphrase to the key's user. Because it isn't good policy for any system administrator to know users' passphrases in any context, users may afterward want to change the key's passphrase. The simplest way for users to do so is via the openssl command, like so:

```
bash-$ openssl rsa -in minion.key -out minion.key -aes192
```

The user does *not* need to be root to do this, provided the proper file permissions are set on minion.key (users should have read/write access on their own keys). After entering this command, users will be prompted for the key file's old passphrase and then twice for the new passphrase.

Once users have copied the files ca.crt, client.crt, client.key and ta.key over to their client system's /etc/openvpn/ directory, they should make sure they have the correct file permissions set. The two .crt files should be world-readable, but only owner-writable (that is, `-rw-r--r--`). The two .key files, however, should be only owner-readable/writable (that is, `-rw-------`).

All four files should be owned by root, assuming your users have root on their own Linux systems. (Setting up OpenVPN for nonroot users and the security challenges of doing so are beyond this article's scope.)

Now that you're clear on how to generate and manage client certificate/key pairs, let's continue working our way down Listing 2. The `ca`, `cert` and

`key` directives specify the paths of your CA key file, client certificate file and client key file, respectively. In Listing 2 the values for these parameters are all just filenames, without their full paths specified. This implies that those three files are in the same directory as the client configuration file itself.

So, unlike on the server, where I left all the certificates and keys in /etc/openvpn/2.0/keys and, therefore, specified a CA certificate path of 2.0/keys/ca.crt, on the client system, you can get by with simply `ca.crt` if the file ca.crt, like the configuration file client.ovpn, is kept in the directory /etc/openvpn/.

The `ns-cert-type server` directive says what type of certificate your client should accept. Because in this example I'm dealing with multiple clients connecting back to a server, in Listing 2, it's set to `server`. This will prevent some other client from impersonating the server in a man-in-the-middle attack; the server's certificate, while signed by the same CA as its clients, has attributes that identify it as a server certificate, not another client certificate.

The last directive in the certificate/key-file portion of Listing 2 is `tls-auth ta.key 1`, which tells OpenVPN to use the file ta.key to add an extra layer of authentication on your VPN tunnel by requiring all packets in the TLS handshake phase at the beginning of each tunnel session to be signed with the specified TLS Authentication key. After the name of the TLS Authentication key (ta.key), specify a number telling in which "direction" to use this file: "0" for the server and "1" for clients.

Other Client Settings

Nearly all of the rest of the directives in Listing 2 are ones I already covered in Parts II and III of this series when dissecting Listing 1, the server configuration file. The client-side settings for those directives are even the same as specified on the server.

In the case of `user nobody` and `group nogroup`, which tell the openvpn command to run with unprivileged user and group identities after initialization, make sure the user account nobody and the group nogroup exist on your client system. Because both typically exist on most Linux systems, they probably do already. If not, you either can create them or change the directives to point to some other, existing dæmon account or group name. In no event should you change either to any user or group name used

for actual human-use login accounts!

The only other directive worth mentioning here is `mute-replay-warnings`, which I didn't include in the server.ovpn file. Declaring this directive (without any argument) tells OpenVPN not to log events associated with its anti-packet-replay mechanism, which tends to trigger false alarms if you connect to a wireless network. It won't turn off the actual anti-replay protection; it just suppresses associated local log entries.

Initiating a Tunnel

You've created a client configuration file and put it into your client system's /etc/openvpn directory. You've copied over your CA certificate file, client certificate and key files, and your TA key file. It's time to connect to the server.

Assuming your client system is running Linux (specifically Ubuntu), and assuming that, as in Listing 2, your client configuration file is called client.ovpn, follow these steps the first time you connect to your server:

1) Change your working directory to /etc/openvpn:

```
bash-$ cd /etc/openvpn
```

2) Run OpenVPN like this:

```
bash-$ sudo openvpn --config ./client.ovpn
```

3) When prompted, enter your client key's passphrase.

```
Enter Private Key Password: Your passphrase here
```

Note that in step 2 you started OpenVPN *without* the `--daemon` directive, leaving it running in the foreground. If everything works without a hitch, the next time you start OpenVPN, you can use `sudo openvpn --daemon --config ./client.ovpn`, in which case OpenVPN will run silently after asking you for your client key passphrase. On my Ubuntu client system, OpenVPN logs to the file /var/log/syslog when running in dæmon mode.

Troubleshooting

Hopefully, at this point you've got a working VPN tunnel back to your server. If you don't though, two different mistakes have caused the majority of my own problems using OpenVPN.

First, your tunnel will fail if you fail to copy all necessary files into /etc/openvpn: client configuration file, CA certificate, client certificate, client key and TLS Authentication key. These files also must have appropriate permissions set, and you must run the openvpn command with the appropriate level of

privilege (that is, root privilege).

Second, firewall rules on both server and clients must be either disabled (left open) or reconfigured to allow OpenVPN traffic to pass. Telling you how to write such rules easily could occupy half or more of an entire article, especially if I were to cover the art of forcing certain types of traffic to use the tunnel. Suffice it to say, for now, check your iptables rules before you even bother running OpenVPN in server or client mode.

Cross-Platform Notes

In writing this article, I tested both an Ubuntu client and a Windows XP client. Getting the Windows XP client to connect properly was no more difficult than on Ubuntu. It was a simple matter of placing the correct files in the correct directory and tweaking my Windows firewall configuration a little.

On Windows clients, the `user` and `group` directives have no effect, as OpenVPN's self-demotion feature is not supported in Windows. Other than that, however, I found the Windows version of OpenVPN to work in a very similar manner as the Linux version.

Conclusion

And that, dear readers, is how you configure OpenVPN to allow yourself to connect back to your home network from an untrusted remote site. This process is, in practice, much easier than my taking four months to describe it implies. Hopefully, my line-by-line dissections of the two configuration files have given you a strong enough understanding of how OpenVPN works for you to explore other usage scenarios.

I may devote one more column to this topic, because Virtual Private Networks are such a powerful tool, and these four installments have covered only a small subset of its potential. Regardless, I hope you've found this series useful and that you have success in your own Virtual Private Linux endeavors. Until next time, be safe!∎

Mick Bauer (darth.elmo@wiremonkeys.org) is Network Security Architect for one of the US's largest banks. He is the author of the O'Reilly book *Linux Server Security*, 2nd edition (formerly called *Building Secure Servers With Linux*), an occasional presenter at information security conferences and composer of the "Network Engineering Polka".

Resources

Official OpenVPN Home Page: **www.openvpn.net**

Ubuntu Community OpenVPN Page: **https://help.ubuntu.com/community/OpenVPN**

BONUS: 12 issues of LJ included

LINUX JOURNAL™

101⁺ TECH TIPS

How-to videos featuring cool tips & tricks that make Linux more fun

DVD™

101 Tech Tips DVD by *Linux Journal* is a series of one-minute how-to videos with shortcuts, tips and cool tricks for Linux users of all levels of expertise. Hosted by *Linux Journal* editor Shawn Powers, this DVD also features fun video extras, DRM-free copies of each tech tip for playing on your computer, as well as an entire year (2009 PDFs) of *Linux Journal* itself!

Buy yours today for just $29.95.

Linux Troubleshooting, Part III: Remote Networks

KYLE RANKIN

The Internet is down. Or, is it? Read more to find out how to know for sure.

This column is the third in a series dedicated to one of my favorite subjects: troubleshooting. Because my column generally is aimed more at tips and tricks and less on philosophy and design, I don't talk much about overall approaches to problem solving. Instead, in this series, I describe some general classes of problems you might find on a Linux system, and then I discuss how to use common tools, most of which are probably already on your system, to isolate and resolve each class of problem.

In my previous column, I introduced some ways to troubleshoot network problems on your local network. Many network problems extend past your local network and either onto other

> **The scenario I use here to test trouble-shooting skills is one that everyone has run into at one point or another—you try to load a Web site, perhaps even a reliable site like Google, and it won't come up.**

local subnets or onto the Internet itself. In this column, I provide you with the tools and techniques for answering that immortal question: is the Internet down, or is it just me?

The Internet Is Down

The scenario I use here to test troubleshooting skills is one that everyone has run into at one point or another—you try to load a Web site, perhaps even a reliable site like Google, and it won't come up. Because I covered local network troubleshooting in my last column, I'm assuming you already have gone through those steps and are ready to proceed past the local network. Even though this example deals with testing access to the Internet, you can use the same steps to troubleshoot problems accessing any remote network.

Test Your Gateway

For your computer to communicate with any other computer outside your local network, you must have a gateway (router) configured on your local network, and you must be able to reach it. Without getting into heavy-duty network theory, a router connects two or more networks and knows how to route packets between those networks. Your Linux computer has a list of all of the routers it knows about for each network of which it is a member and when it should use those routers all stored in its routing table. You can use the route command to show your computer's current routing table:

```
$ route -n
Kernel IP routing table
Destination  Gateway   Genmask        Flags  Metric  Ref  Use  Iface
10.1.1.0     *         255.255.255.0  U      0       0    0    eth0
default      10.1.1.1  0.0.0.0        UG     100     0    0    eth0
```

In the above example, I have one gateway defined: 10.1.1.1. It is listed as my default gateway, which is the router it will use whenever it doesn't have any other routers defined for that network. In my case, it's also the only router in my routing table. That means any time my machine wants to communicate with a remote network (in my example, anything that's not within 10.1.1.0/255.255.255.0 or 10.1.1.1–10.1.1.254), it's going to send the packet to 10.1.1.1 to forward on.

So now that I know my default gateway, I use ping to test whether it's available:

```
$ ping -c 5 10.1.1.1
PING 10.1.1.1 (10.1.1.1) 56(84) bytes of data.
64 bytes from 10.1.1.1: icmp_seq=1 ttl=64 time=3.13 ms
64 bytes from 10.1.1.1: icmp_seq=2 ttl=64 time=1.43 ms
64 bytes from 10.1.1.1: icmp_seq=3 ttl=64 time=1.79 ms
64 bytes from 10.1.1.1: icmp_seq=5 ttl=64 time=1.50 ms

--- 10.1.1.1 ping statistics ---
5 packets transmitted, 4 received, 20% packet loss, time 4020ms
rtt min/avg/max/mdev = 1.436/1.966/3.132/0.686 ms
```

In this example, four out of five ping packets were received, so I can be reasonably sure my gateway works. If I couldn't ping the gateway, either my network admin is blocking ICMP packets (I hate when people do that), my switch port is set to the wrong VLAN, or my gateway is truly down. If the gateway is down, fixing the problem might mean rebooting your DSL or wireless router (if that's how you connect to the Internet) or moving your troubleshooting to whatever device is acting as your gateway.

Test DNS

In my case, I was able to ping the gateway, so I'm ready to move on to DNS. Because most of us don't browse the Web by IP address, we need DNS to resolve the hostnames we type into IP addresses. If DNS isn't working correctly, even if we technically can reach that remote IP address, we never will know what the IP address is.

A basic way to test DNS is via the nslookup command:

```
$ nslookup www.linuxjournal.com
Server:    10.2.2.2
```

Address: 10.2.2.2#53

Non-authoritative answer:
Name: www.linuxjournal.com
Address: 76.74.252.198

In this example, DNS is functioning correctly as far as I can tell. I say as far as I can tell, because I'm assuming that 76.74.252.198 is the correct IP address for www.linuxjournal.com. If it were the wrong address, that very well could be the cause of the problem! The DNS server in this case is 10.2.2.2, but in some environments, it could be the same IP address as your gateway.

Even though the DNS server worked, because I want to show how to troubleshoot DNS, I need some examples of how it can fail. To illustrate this, let me show a few different nslookup commands that have failed:

```
$ nslookup www.linuxjournal.com
;; connection timed out; no servers could be reached
```

This error tells me that nslookup couldn't

communicate with my DNS server. That could be because either I don't have any name servers configured on my system or I just can't reach them. To see whether I have any name servers configured, I would check my /etc/resolv.conf file. This file keeps track of what name servers I should use. In my case, it would look like this:

```
search example.net
nameserver 10.2.2.2
```

If your resolv.conf file doesn't have a name server entry, you have found the problem. You need to add the IP address of your name server here. Because I do have a name server defined in resolv.conf, the next step is to attempt to ping the name server's IP with the same ping command that I used for the gateway above. If you can't ping the name server, either a firewall is blocking ICMP (those pesky network administrators!) or there's a routing problem between you and the name server. To rule out the latter, use a tool called traceroute.

If DNS isn't working correctly, even if we technically can reach that remote IP address, we never will know what the IP address is.

Traceroute tests the route between you and a remote IP address. To use it, type `traceroute` followed by the IP address you want to reach. In my case, I would use 10.2.2.2:

```
$ traceroute 10.2.2.2
traceroute to 10.2.2.2 (10.2.2.2), 30 hops max, 40 byte packets
1  10.1.1.1 (10.1.1.1)  5.432 ms  5.206 ms  5.472 ms
2  10.2.2.2 (10.2.2.2)  8.039 ms  8.348 ms  8.643 ms
```

In this example, I can route to 10.2.2.2 successfully. To get there, my packets first go to 10.1.1.1 and then move straight to 10.2.2.2. This tells me that 10.1.1.1 is likely the gateway for both networks. If there are more routers between you and your remote server, you will have more hops in between. On the other hand, if you do have a routing problem, your output might look more like the following:

```
$ traceroute 10.2.2.2
traceroute to 10.2.2.2 (10.2.2.2), 30 hops max, 40 byte packets
1  10.1.1.1 (10.1.1.1)  5.432 ms  5.206 ms  5.472 ms
2  * * *
3  * * *
```

If you start seeing asterisks in the output, you know the problem likely begins on the last router

on the list, so you would need to start troubleshooting from that router. Instead, you might see output like this:

```
$ traceroute 10.1.2.5
traceroute to 10.1.2.5 (10.1.2.5), 30 hops max, 40 byte packets
1  10.1.1.1 (10.1.1.1)  5.432 ms  5.206 ms  5.472 ms
1  10.1.1.1 (10.1.1.1)  3006.477 ms !H  3006.779 ms !H  3007.072 ms
```

This means your ping timed out at the gateway, so the remote host could be down, unplugged or otherwise inaccessible, so you would need to troubleshoot its connection to the network.

Note: traceroute relies on ICMP, so if ICMP is blocked on your network, install a tool called tcptraceroute to perform a similar test over TCP (the syntax is the same, you just type `tcptraceroute` instead of `traceroute`).

If you can ping the name server but it isn't responding to you, go back to my previous column and perform all the troubleshooting steps to test whether the remote port is open and accessible on the remote host. Keep in mind though that DNS servers use port 53 on TCP and UDP. Again, if you aren't sure what port a service uses, check the /etc/services file on your system. It lists most of the common services you will use.

Other Name Server Problems

Another common nslookup error you might run into is this:

```
$ nslookup web1
Server:         10.2.2.2
Address:        10.2.2.2#53

** server can't find web1: NXDOMAIN
```

Here my name server at 10.2.2.2 responded to me but told me it couldn't find the record for server web1. This error could mean that I don't have web1's proper domain name in my DNS search path. If you don't specify a host's fully qualified domain name (for instance, web1.mysite.com) but instead use the shorthand form of the hostname, your system will check /etc/resolv.conf for domains in your DNS search path. It then will add those domains one by one to the end of your hostname to see if it resolves. The DNS search path is the line in /etc/resolv.conf that starts with the word `search`:

```
search example.net example2.net
nameserver 10.2.2.2
```

In my case, when I search for web1's IP address, my system will first search for web1.example.net, and if that has no records, it will search for

web1.example2.net. If you want to test whether this is the problem, simply run nslookup again but with the fully qualified domain name (such as web1.mysite.com). If it resolves, either make sure you always use the fully qualified domain name when you access that server, or add that domain to the search path in /etc/resolv.conf.

If you try nslookup against the fully qualified domain name and you still get the same NXDOMAIN error above, your problem is with the name server itself. Troubleshooting the full range of DNS server problems is a bit beyond what I could reasonably fit in this column, but here are a few steps to get you started. If you know your DNS server is configured to have the record you are looking for itself, you need to examine its zone records to make sure that particular hostname exists. If, on the other hand, you are searching for a domain for which you know it doesn't have a record (say, www.linuxjournal.com), it's possible your DNS server isn't allowing recursive queries from your host or at all. You can test that by trying to resolve some other remote host on the Internet. If it doesn't resolve, it's probably a recursion setting. If it does resolve, the problem might very well be with that remote site's DNS server.

Test General Internet Routing

If after all these tests you find that your DNS servers are working fine, but you still can't access the remote server, the final step is to perform another traceroute like above, only directly against the remote server. So for instance, if you wanted to test your route to www.linuxjournal.com, the traceroute might look like the following:

```
$ traceroute www.linuxjournal.com
traceroute to www.linuxjournal.com (76.74.252.198), 30 hops max,
➥60 byte packets
1  10.1.1.1 (10.1.1.1)  1.016 ms  2.222 ms  2.308 ms
2  75-101-46-1.dsl.static.sonic.net (75.101.46.1)  6.916 ms
➥7.389 ms  8.386 ms
3  921.gig0-3.gw.sjc2.sonic.net (75.101.33.221)  11.265 ms
➥12.435 ms  13.050 ms
4  108.ae0.gw.equinix-sj.sonic.net (64.142.0.73)  13.846 ms
➥15.233 ms  15.390 ms
5  GIG2-0.sea-dis-2.peer1.net (206.81.80.38)  35.149 ms
➥36.272 ms  36.944 ms
6  oc48.so-2-1-0.sea-coloc-dis-1.peer1.net (216.187.89.190)
➥37.340 ms  27.884 ms  27.266 ms
7  10ge.ten1-2.sj-mkp16-dis-1.peer1.net (216.187.88.202)
➥28.421 ms  29.014 ms 29.688 ms
8  10ge.ten1-2.sj-mkp2-dis-1.peer1.net (216.187.88.134)
➥30.903 ms  31.015 ms 31.804 ms
9  10ge.ten1-3.la-600w-cor-1.peer1.net (216.187.88.130)
➥40.840 ms  41.279 ms 42.069 ms
10  10ge.ten1-1.la-600w-cor-2.peer1.net (216.187.88.146)
➥42.587 ms  43.710 ms 44.921 ms
11  10ge-ten1-2.dal-eqx-cor-1.peer1.net (216.187.124.122)
➥81.702 ms  82.959 ms 83.934 ms
12  10ge-ten1-1.dal-eqx-cor-2.peer1.net (216.187.124.134)
➥74.876 ms  72.454 ms 72.798 ms
13  10ge-ten1-3.sat-8500v-cor-2.peer1.net (216.187.124.178)
➥80.224 ms  81.872 ms  82.569 ms
14  216.187.124.110 (216.187.124.110)  83.499 ms  84.162 ms
➥85.048 ms
15  www.linuxjournal.com (76.74.252.198)  85.484 ms  86.461 ms
➥87.153 ms
```

In this example, I'm 15 hops (or routers) away from the www.linuxjournal.com server. This is an example of a successful query, but if you ran the same query and noticed a number of rows of asterisks that never made it to your destination *and* you couldn't ping www.linuxjournal.com directly, the problem could be an Internet routing issue between you and the remote network. Unfortunately, it's probably something outside your control, but fortunately, these sorts of problems tend to resolve themselves pretty quickly, so just keep trying.

If you try nslookup against the fully qualified domain name and you still get the same NXDOMAIN error above, your problem is with the name server itself.

If, on the other hand, your traceroute command was successful, but the remote site still didn't work, go back to the steps I discussed in my previous column on how to use telnet and nmap to test whether a remote port is open. It actually could be that the remote server is down (hey, it happens to the best of us) or that someone has configured a firewall to block you from that remote server.

I hope this series has kindled (or rekindled) your interest in troubleshooting under Linux. One of the things I love about Linux is how little it hides from you about how it works and how many troubleshooting tools it provides when things do go wrong. If this has piqued your interest, there are many more troubleshooting avenues for you to explore—from DNS servers like I mentioned above, to troubleshooting just about any type of service. Also, if you have any other great tools or techniques you use to track down these problems, drop me a line. I'm always on the lookout for tools to solve problems faster.■

Kyle Rankin is a Systems Architect 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.**

Remix the Internet and Your Television with the Roku DVP

DIRK ELMENDORF

Developing custom channels for the Roku.

This month, the challenge is to write about Linux and entertainment. MythTV is cool, but it's been covered. Boxee looks great, but someone beat me to that punch. I was getting worried I wouldn't find something appropriate to cover until my wife reminded me that the device on which we were watching *Farscape* runs Linux, and she suggested I see if there was something worth discussing beyond the great selection of documentaries. As luck would have it, there is quite a bit worth writing about.

The device is the Roku Digital Video Player, which launched in 2008. This tiny box uses Linux to stream Netflix video to your television. Until recently, that was the only reason to get one, but now, Roku has started announcing other channels of content (Major League Baseball, Facebook, Pandora and more). It is interesting to see the tiny little box expand its capability. Even more interesting is that Roku has opened a developer program that allows anyone to create custom channels for it.

Before I dive in, here's a little background. My wife is from Seattle, Washington, and is a huge fan and supporter of the KEXP radio station. I have done several projects in the past to bring KEXP to our home in San Antonio, Texas, but looking at the Roku's capabilities, it seems like the perfect platform for bringing KEXP on-demand to our living room.

The first step in becoming a developer is to go to the Roku.com Web site and create an account in its portal. After that, click the link for Developer. There is a small sign-up form where you agree to abide by the rules of the developer program. Once completed, you can download the SDK and get started.

Roku tries to honor its obligation to the GPL code involved in allowing the Roku to work. If you visit **www.roku.com/support/gpl_rdvp.aspx**, you can download tarballs of software. One thing that struck me in reading the developer agreement is the section that says "Subject to the Grace Period, Your Channel Application must at all times....iii. not contain any open-source code or other restricted code that could require Roku to publicly post or display any third-party notices or any modifications to such code." I understand what the Roku folks are trying

to protect themselves from. Apparently, parts of the Roku platform are licensed from third parties (like the support for WMA), and Roku wants to keep parts proprietary. I assume the MIT license is fine, but it sounds like this environment might not be very friendly to the GPL. I posted a question to the developer forum but did not get any response by the time I had submitted this article for publication.

Assuming you still are interested in the platform, you have three different options for deploying a channel: push it to a local Roku, create it as a private channel or get it approved as a public channel.

The public channel requires Roku's approval. Once it's approved, Roku offers it in the channel store on the Web site. This is the path to take if you are building a channel for public consumption. The private channel allows you to roll out your channel to people. The upside is you don't need Roku's approval to roll it out. The downside is that more steps are involved in getting the channel onto a Roku. The local push is designed for the development phase of your channel. Once you enable development on a Roku (use a remote to enter Home 3x, Up 2x, Right, Left, Right, Left, Right), you can just upload your channel. The SDK includes a handy Makefile to make this incredibly easy. This is how I have done all of my development, as I'm developing channels for myself (and other developers). The main limitation of this last form of development is that you can have only one "development" channel installed at a time. That did not pose much of a problem for me, but if you're sharing code, it could make things more complicated.

Roku development is all done in BrightScript, which appears to be a custom language that combines ideas from Visual Basic and JavaScript. The main goal is to allow you to write code in a dynamic language that can easily compile down to efficient code for the Roku's embedded environment. This is accomplished by adding in BrightScript Components, such as default screens and media tools. You use BrightScript to customize the screens and build up your channel. There is even

a PowerPoint presentation template to allow you to mock up your application screens easily.

Getting Down to Business

All the code for this article is being shared at **github.com/economysizegeek/linux_journal_roku**. This will make it a lot easier to see the different channels. Simply `git clone` that repository, and you will have the latest version of all the code. I assume you have turned on developer mode for your Roku. You also need to set an environment variable in your shell so that the Makefile will know where to push the code. In my case, I added `export ROKU_DEV_TARGET=192.168.210.244` to my .bashrc. Make sure you put in the IP address of your Roku.

Let's start by doing a simple Hello World! example. This will get you familiar with the tools and also confirm that you have everything working. In the git repository is a directory called hello_world. Inside that directory, you should be able to type `make install`, and it will push a new channel onto your Roku automatically. If that doesn't work, confirm that you have make, curl and zip installed (the Makefile depends on them). Also make sure your Roku is in Dev Mode (enter the key combo from this article). Additionally, ensure that you didn't set the environment variable (`echo $ROKU_DEV_TARGET` should print an IP), and check that you didn't set it to the wrong IP (the Roku's IP can be confirmed in Settings→Player Info).

Once you have sorted that out, you should be able to go to the Roku and see a new channel called "Hello World". Clicking on it with your remote will start the channel. It doesn't actually do anything other than say "Hello World!" The point is to confirm your environment and give you a quick-and-dirty tour of what you need to start making your own channel.

In the hello_world directory, you will see two directories and a few files. The images directory has a variety of images needed for this small application. You will notice that some have HD and some have SD in their names. This represents High Definition and Standard Definition. Because the Roku is connected to a television, you have to make sure any art you display is sized correctly for the TV screen. The Makefile is there to make it easy to compile, install and remove the channel. The manifest file is required by the Roku to build the package on the other side. That just leaves the source directory and the HelloWorld.brs file (Listing 1). The brs file is the BrightScript that actually sets up the channel.

Let's do a walk-through of the code, but if the version you get from the git repository is different, it means I fixed a bug after this went to press.

The first code listing is a crash course in BrightScript. All it does is create a Main that builds a single screen

Listing 1. HelloWorld.brs

```
'Copyright 2010 Dirk Elmendorf

Function Main() as Integer
  print "Hello World! to the console"
  print ' you will not see this '
  p = CreateObject("roMessagePort")
  screen = CreateObject("roSpringboardScreen")
  screen.SetMessagePort(p)

  o = CreateObject("roAssociativeArray")
  o.ContentType = "episode"
  o.Title = "Sample App"
  o.Description = "Hello World!"
  o.SDPosterURL = "pkg:/images/episode_icon_sd.png"
  o.HDPosterURL = "pkg:/images/episode_icon_sd.png"
  screen.SetContent(o)
  screen.AddButton(1, "Exit")
  screen.Show()
  while true
    msg = wait(0, p)
    if msg.isButtonPressed()
      print "Goodbye World!"
      return 0
    endif
  end while
end Function
```

and exits on any button push. The print statements are spit out to the Roku console. The console also allows you to enter a debugger and see any syntax errors. You can access the Roku console by entering `telnet $ROKU_DEV_TARGET 8085`. I had to use it a couple times to find out why the channel wasn't showing up (which often means there is a syntax error).

As you can see, I created a function called Main. This is the entry point for all BrightScript applications. I included two different print statements. The first one works, but the second one only prints out a blank line, because a single quote is considered a comment in BrightScript. Then, I created two objects. The roMessagePort is where messages (events) are sent. The roSpringboardScreen is a BrightScript component. This is basically a screen you can reconfigure by providing information. Roku ships with a number of these screens for your use. I chose this one because it was the easiest to work with. I connect the screen and the port. This tells the system to send all events from the screen to the message port I created. There probably are cases where you need to have different message ports for different events, but so far, I have used only a single message port shared with every screen.

Then, I create a roAssociativeArray. This is very much like a hash (or a plain JavaScript object). I use it to set up up a number of predefined fields. This gets passed into the screen. Next,

I create a single button on the screen and tell the system to draw the screen. The last step is to set up an infinite loop. The wait allows the system to sit until an event is triggered. Then, I can have the result of that event processed. In this case, it means the program exits.

Listing 2. KEXP.brs

```
'Copyright 2010 Dirk Elmendorf

Function Init() as Object
  obj = {
    port: CreateObject("roMessagePort")
    screen: CreateObject("roSpringboardScreen")
    player: CreateObject("roAudioPlayer")
    screen_options: CreateObject("roAssociativeArray")
    song: CreateObject("roAssociativeArray")
    status: ""
    drawScreen: function(description)
        m.screen_options.Description = description
        m.screen.SetContent(m.screen_options)
        m.screen.Show()
    end function
    playingNow: function()
        m.screen.ClearButtons()
        m.screen.AddButton(1, "Pause Stream")
        m.screen.AddButton(3, "Exit")
        m.DrawScreen("Live MP3 Stream from KEXP.org")
    end function
    play: function()
        if m.status = "" then
          m.player.AddContent(m.song)
          m.player.SetLoop(true)
          m.player.play()
          m.status = "playing"
        else if m.status = "paused"
          m.player.resume()
          m.status = "playing"
        endif

        m.screen.ClearButtons()
        m.screen.AddButton(3, "Exit")

        m.drawScreen("Buffering....")
    end function
    pause: function()
        if m.status = "playing" then
          m.player.pause()
          m.status = "paused"
        endif

        m.screen.ClearButtons()
        m.screen.AddButton(2, "Resume Stream")
        m.screen.AddButton(3, "Exit")

        m.drawScreen("Stream Paused")
    end function

    exit: function()
        print "Goodbye World!"
        m.player.stop()
        return 0
    end function
  }
  obj.screen.SetMessagePort(obj.port)
  obj.player.SetMessagePort(obj.port)


  obj.screen_options.ContentType = "episode"
  obj.screen_options.Title = "KEXP"
  obj.screen_options.SDPosterURL = "pkg:/images/episode_icon_sd.png"
  obj.screen_options.HDPosterURL = "pkg:/images/episode_icon_sd.png"
  obj.screen.SetStaticRatingEnabled(false)

  obj.song.Url = "http://kexp-mp3-2.cac.washington.edu:8000/"
  obj.song.StreamFormat = "mp3"
  obj.status = ""
  return(obj)
end Function


Function Main() as Integer

  app = Init()
  app.play()

  while true
    msg = wait(0, app.port)
    if type(msg) = "roAudioPlayerEvent"
      if msg.isStatusMessage() then
        print "Audio Player Event:"; msg.getmessage()
        if msg.GetMessage() = "start of play" then
          app.playingNow()
        endif
      endif
    else if msg.isButtonPressed()  then
        if msg.GetIndex() = 1 then
          app.pause()
        else if msg.GetIndex() = 2 then
          app.play()
        else if msg.GetIndex() = 3 then
          return app.exit()
        endif
    else if msg.isScreenClosed() then
      return app.exit()
    endif
  end while
end Function
```

Rocking Out

The next step was to add in another BrightScript component—namely roAudioPlayer. This component handles playing MP3 streams. I wrote a new script from scratch to do this. The channel to handle streaming is in the kexp directory. The main file (Listing 2) is in the source/KEXP.brs (at **github.com/economysizegeek/linux_journal_roku**).

Listing 2 is for this channel. It is different in two main ways. First, this time, the event loop actually looks at the button being pushed. The loop is now able to trigger a different behavior depending on the index value of the button that was pushed. The other change is that I created an application object. I wanted to demonstrate that Roku has provided an incredibly dynamic language, so I built it up much the same way you would in JavaScript. The one odd thing to notice is the use of "m." inside the methods for the app object. The "m." is a pointer to the object. You can read it as "this.", because that is how Roku is resolving it when it gets compiled.

The event code is also tracking the state of the stream itself. The roAudioPlayer considers it an error if you try to pause a stream that is not playing or try to resume a stream that is not paused. The code handles those cases. It also handles shutting down the stream when you exit.

With a working streaming application in hand, I started trying to get the "on-demand" portion of the KEXP Web site integrated into it. The first step was to navigate to a show and find a URL for a single show. This took some work, because the on-demand programs are served using a different protocol and URL from the live feed. Eventually, I was able to get a URL by downloading and opening files that the KEXP server provides.
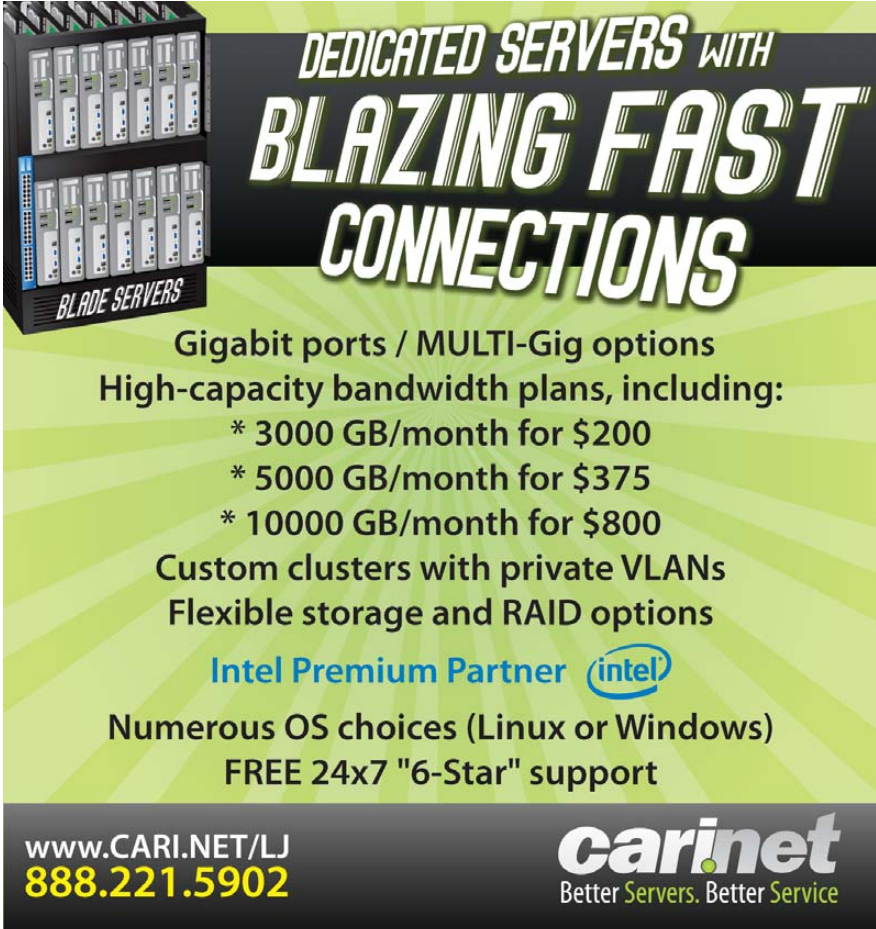
Unfortunately, I ran into a Roku platform limitation. It does not support the codec that KEXP is using. Currently, there is no way for a third-party developer to add additional codec support. Because the live stream is available in MP3, the Roku handles it fine, which has been the case for a number of the radio feeds I looked at. So, if the stream you want to use streams via Flash or some other codec, you may be out of luck as well. Roku also supports WMA, but that seems to be more for files than streams.

For this article, I wrote everything from scratch. When you are working on your own projects, you won't need to do that. Roku includes a number of different sample applications in the SDK. The code gives a great tour of the components and how they can be used, including audio, video, XML parsing and handling a registration process. Those examples also are a lot more polished than the code I have provided here (they include examples of how to create your own theme for the channel).

The next thing to play with is video streaming. I chose to focus on audio for this article, because it's very straightforward. Streaming video means obtaining a video source, as well as encoding it properly, and then you have to generate an XML document to describe it. You will need a Web server that can serve both the video and the XML. Once you have everything working, you should be able to run your own little Netflix-style video-streaming service. Now that you have your feet wet with Roku development, head over to Brian Lane's blog. He has documented some hacking he's done to get all of this working (**blog.brianlane.com/2009/12/20/streaming-local-video-with-your-roku**).

My goal with this article was to get streaming up for my wife (which I mostly achieved—I'm still working on getting the *Shake the Shack* show on demand). I was incredibly surprised at how easy it was to get up and hacking. Originally, I was turned off by having to deal with a new language, but BrightScript is close enough to JavaScript that it wasn't the hurdle I thought it would be. For me, the best part is that the Roku already had earned its place in my AV rack thanks to Netflix. Adding the ability to hack on it and the opportunity to try channels other people have built makes it that much more awesome.■

Dirk Elmendorf is cofounder of Rackspace, some-time home-brewer, longtime Linux advocate and even longer-time programmer.

Safentrix's Cloud-Based Hosted E-mail Security Service

Safentrix bills its new and free cloud-based hosted e-mail security service as the world's first to offer out-bound e-mail security. The service provides spam and virus filtering for incoming e-mail as well. The firm says its product provides peace of mind to organizations worried about misused e-mail servers, given that the majority of spam results from this misuse. Other key features include compatibility with all common e-mail servers, a control panel secured with high-grade encryption, a geographically dispersed network with multiple redundant servers, near-zero false positives, near-zero maintenance, up to 20,000 users per domain and premium and free service levels.

www.safentrix.com
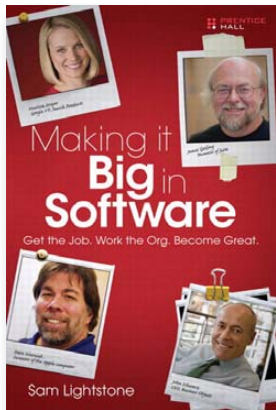
Sam Lightstone's *Making it Big in Software* (Prentice Hall)

Got the drive to be a software legend but lacking the road map to get there? You might want to pick up Sam Lightstone's new book *Making it Big in Software: Get the Job. Work the Org. Become Great.* from Prentice Hall. Lightstone covers the road to the top in a step-by-step manner, from landing a job with great potential, to moving your way up and then blazing your way to the pinnacle. He says that everything has changed radically in "the biz", and professors and career counselors never tell you the complete story. Other topics include mastering nontechnical skills crucial for success; understanding the "sweet science" of software R&D; "working the org" to move up rapidly; managing time, projects and life; and even breaking out into entrepreneurship. Also included are 17 interviews with legends such as Steve Wozniak, Marc Benioff and our own personal fave, Linus Torvalds.

www.informit.com

VariCAD

Among the most veteran CAD apps on the Linux platform, VariCAD, now in version 2010 2.0, is a 3-D/2-D CAD system for mechanical engineering design. Core features include tools for 3-D modeling and 2-D drafting and dimensioning; libraries of standard mechanical parts (ANSI, DIN); calculations of standard mechanical components; and tools for working with bills of materials (BOM) and blocks. Version 2010 2.0 adds new features, such as geometric constraints among entire solids, advanced tools for working with Boolean trees, a significantly improved user interface and improved DWG import. A free 30-day trial version is available for download.

www.varicad.com
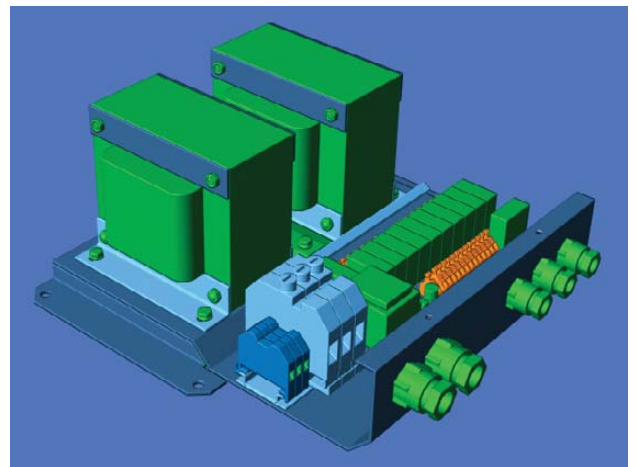
AdMarvel's Android Advertising Toolkit

AdMarvel has beefed up its Android Advertising Toolkit, an open platform and toolset that enables Android application developers to integrate rich-media advertising into their applications. The new version 1.5 also supports rich-media advertising on the mobile Web on Android phones. Advertising metrics and analytics are provided via an on-line Web console giving a near real-time view of the application's advertising performance. AdMarvel also provides a fully functional rich-media ad server, supporting the provisioning and management of direct-sourced or in-house advertising.

www.admarvel.com
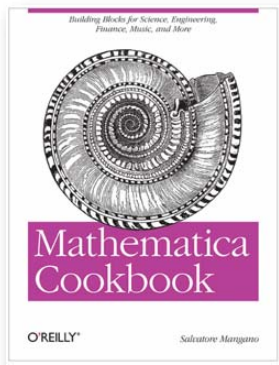
Salvatore Mangano's *Mathematica Cookbook* (O'Reilly)

If you want to get the most out of Wolfram Research's famous Mathematica application, pair it with Salvatore Mangano's *Mathematica Cookbook* for useful and informative "recipes" of all kinds. The book covers all levels of applying Mathematica for data visualization, covering everything from high-school algebra to sophisticated analyses in science, engineering, financial analysis, software development and more. Some specific topics include 3-D imaging, audio processing, calculations from mathematics and physics, image processing, distributed computing and many others.

www.oreilly.com

Wolfram Research's Workbench

Speaking of Mathematica, that application's maker, Wolfram Research, has released version 2 of Workbench, an Eclipse-based IDE that is specialized for Mathematica and other Wolfram technologies. The platform, says Wolfram, "combines Mathematica's ever-growing programming language and world-renowned computation, visualization and modeling features with one of the world's leading IDEs". Developers, thus, enjoy the luxury of working with any Eclipse-supported language. New in version 2 is the ability to create documentation for Mathematica applications, as well as a host of improvements to code editing and navigation. Workbench requires Mathematica 6 or higher and is compatible with Linux, Windows and Mac OS X. Workbench is also a key tool internally for Wolfram for the development of Mathematica, Wolfram|Alpha and other technologies.

www.wolfram.com

Wolfram Workbench 2

Ksplice Uptrack Service

Likening this service to changing out a car's engine while speeding down the highway, Ksplice Uptrack from Ksplice, Inc., eliminates the need to reboot Linux servers to reboot their kernels. Thus, crucial updates and security patches can be applied live without rebooting or disrupting running applications. Ksplice Uptrack subscriptions are currently available for six leading versions of Linux including Red Hat Enterprise Linux, Ubuntu, Debian GNU/Linux, CentOS, Parallels Virtuozzo Containers and OpenVZ.

www.ksplice.com

Fluffy Spider's FancyPants

FancyPants is a multimedia applications framework from Fluffy Spider Technologies, whose new version 3.0 now supports Google Android. The company says that FancyPants gives OEMs of Android-based devices and their channel partners new capabilities to differentiate their wares in an increasingly crowded marketplace. Besides mainstream Android mobile development, FancyPants supports digital video (STB, DVR, home-media center), office automation (printers and MFPs) and other consumer electronics and industrial automation areas. FancyPants further enables total separation of user interfaces from applications that use them, offering developers "a unique autonomous user-interface model, enabling comprehensive deep customization of Android device GUIs on a per-application, or suite of applications, basis".

www.fluffyspider.com

FLUFFY SPIDER TECHNOLOGIES
Idea to pixel, Pixel to screen

Fresh from the Labs

gbrainy—Brain Teaser
live.gnome.org/gbrainy

People interested in sharpening their mental faculties in an environment that is genuinely challenging would do well to give this a go. According to gbrainy's Web site:

> gbrainy is a brain-teaser game and trainer to have fun and to keep your brain trained. It provides the following types of games:

■ Logic puzzles: games designed to challenge your reasoning and thinking skills.

■ Mental calculation: games based on arithmetical operations designed to prove your mental calculation skills.

■ Memory trainers: games designed to challenge your short-term memory.

■ Verbal analogies: games that challenge your verbal aptitude.

> gbrainy provides different difficulty levels, making gbrainy enjoyable for kids, adults or senior citizens. It also features player's game history, player's personal records, tips for the player, or fullscreen mode. gbrainy also can be extended easily with new games developed by third parties.

Installation For those chasing binaries, packages are available for Debian, Ubuntu, Fedora, SUSE, Mandriva, One Laptop Per Child and NetBSD. Also available is the obligatory source tarball, and for those who want to run the bleeding-edge version, you can grab it using git. I won't be covering how to run git here, but nevertheless, I will incorporate git users into a line of my instructions. As usual, I'm going with the source tarball.

Here are the library requirements,
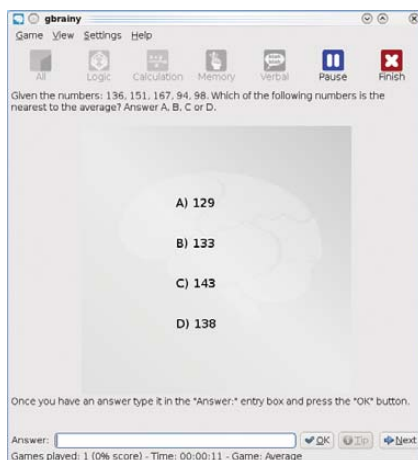
gbrainy presents you with challenging mathematical problems that often test your spatial abilities alongside arithmetic.
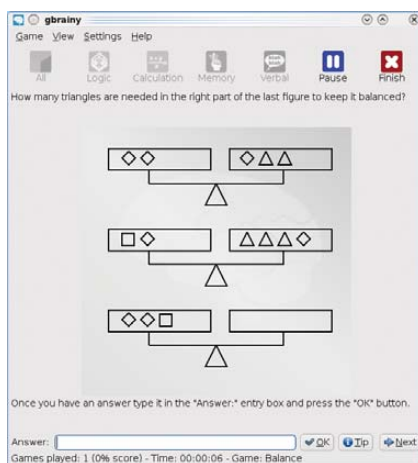
Interesting sequences and lateral thinking often play a part in gbrainy's Logic section.

according to the gbrainy Web site:

■ intltool 0.35 or higher.

■ Mono 1.1.7 or higher.

■ GTK and GTK Sharp 2.8 or higher.

■ librsvg 2.2 or higher.

■ Cairo 1.2 or higher.

■ Mono.Addins 0.3 or higher.

And, also according to the Web
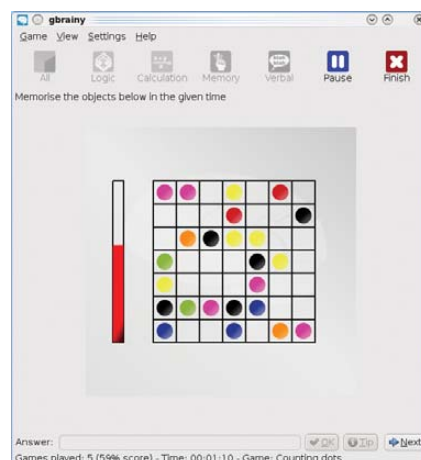
In the Memory tests, you'll find yourself using different ways to memorize things from what you may have done previously, especially when you're given such a short time limit.

site, for a standard Ubuntu installation, the packages required for compiling gbrainy are intltool, mono-gmcs, mono-devel, libmono-dev, libgnome2-dev, libgnomeui-dev and libmono-cairo2.0-cil.

Download the latest tarball, extract it, and open a terminal in the new folder. If you've downloaded the development git version, before carrying on, enter:

```
$ ./autogen.sh
```

For everyone else, simply enter:

```
$ ./configure
```

If your distro uses sudo:

```
$ sudo make install
```

If your distro doesn't use sudo:

```
$ su
# make install
```

Note that compilation doesn't seem to require the use of make. Once the installation is finished, run the program with the command:

```
$ gbrainy
```

Usage Once you're inside, using

gbrainy is pretty straightforward. The GUI layout and design are excellent, along with many other functions, such as text input, difficulty settings and so on. gbrainy even has a special setting to adjust for the color blind. Starting out in gbrainy is obvious. There are buttons for testing the following subjects: Logic, Calculation, Memory, Verbal or All (for taking every subject at once).

As each test starts, bear in mind that most questions aren't in the realm of "mildly taxing". Even on the default medium setting, most of these questions are quite tricky. It quickly will become apparent which subjects you have a knack for and which you'll struggle with.

Logic uses puzzles, such as number and block sequences, grid layouts and so on, generally for guessing the next number or arrangement in a sequence.

Calculation typically has you guessing for the closest number to something, using very thought-intensive divisions, multiplications, averages and so on, that seem to move beyond basic arithmetic skills.

Memory introduces a time factor, so pay close attention. Things such as a series of directions, a grid containing different objects or words spelling out colors that are actually colored with something different (such as the word yellow colored blue) are shown on-screen for a very short amount of time, and you'll be asked about an individual element that's usually pretty hard to recall.

And finally, there's Verbal, which plays games like finding the odd word out, identifying the word most suitable for a given situation, choosing what word can be used in relation to a certain word pairing and so on.

Thankfully, the text input is very dynamic in that the developers have thought ahead about what possible combinations people may enter, so chances are if you enter something in a different case or some other kind of textual variation, the input accepts it or instructs you as to how to answer the question.

All in all, this is a very challenging package. I wouldn't be surprised if it becomes a mainstay of educational distros, but hopefully, it catches on in standard distributions as well.

Step Into Chinese— Language-Mining and Flashcard System

new.asymptopia.org/staticpages/
index.php?page=StepIntoChinese

If you've been playing around with other flashcard systems trying to learn Chinese or perhaps running into problems with switching between Traditional Chinese, Simplified Chinese and Pinyin, this may be the project for you. To quote the Freshmeat page and parts of the Web site:

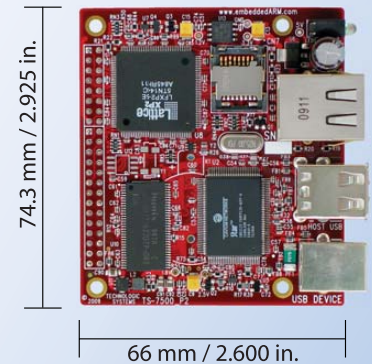Step Into Chinese is a flexible language-mining and flashcard system to assist English speakers seeking to understand the Chinese language. It was designed to address the lack of a one-to-one correspondence between Chinese characters and corresponding Pinyin morphemes.

The lack of a one-to-one correspondence between Chinese characters and the corresponding Pinyin is often regarded as the greatest difficulty facing learners of Chinese. Step Into Chinese has been designed to address exactly this difficulty.

Inside is an extensively cross-referenced data structure that allows the user to pursue deeper understanding of contexts, for example, by "locking on" to a particular Pinyin context and viewing successive instances of the same morpheme used in similar contexts.

The user can also "lock on" to English words in the Pinyin translations, English words in the collective phrase translations, and even Unicode strings used to represent Chinese characters in either traditional or simplified representations. Frequency information relative to the number of occurrences of each Pinyin morpheme, in each context throughout the data structure, is displayed for each entry.

The application data structure contains over 26,000 modern

Chinese words and concepts, corresponding to over 8,300 separate Chinese characters. Colors are used consistently throughout the application for rapid location and absorption of information.

Installation Apologies for the long project URL above—you're probably better off just visiting the main page at **new.asymptopia.org**. Why this confusion? Well, Step Into Chinese (SIC) is part of a collection of software developed by Asymtopia Software, a grass-roots open-source development company. In order to download SIC, you first need to create an account at the Web site, but don't worry, it's "free and only to prevent excessive download abuse by non-humans". It also allows you to participate on the Web site. I'm interested in several projects from Asymtopia and probably will be highlight them in the coming months, so you might want to go ahead and make an account now.

Before I go further, in terms of library requirements, you need to install the pygame libraries, which on my Kubuntu system were named python-pygame. As far as package options go, there seems to be only a .tar.gz file (no binaries), but don't fret, the installation is easy. Download the newest tarball, extract it, and open a terminal in the new folder.

If your distro uses sudo, enter:

```
$ sudo setup.py
```

If your distro doesn't use sudo, enter:

```
$ su
# setup.py
```

From here, you can run the program with:

```
$ stepintochinese
```

Usage First, a screen displays the status of some postprocessing, which has to do with Pinyin morphemes and their usage contexts. This may take a minute, but let it be. Soon you'll be inside the main screen. To give you an idea how it works, SIC already will have a word entered (that word being hello) and in the process of being displayed in "screensaver mode", where each English word is shown along with its Chinese translation in both Pinyin (Romanized

Step Into Chinese allows you to look up English words in both Traditional and Simplified Chinese, as well as Pinyin.

You also can look up words in Chinese using Pinyin (Romanized letters); here I'm displaying a word in Chinese Simplified.

letters) and Chinese characters, which can be set to either Traditional or Simplified.

Now all this might be a bit confusing to take in at first, so let me break it down for you. The word hello will be on-screen, but you'll want to enter your own words to translate, so let's do that now. First,

press backspace once, and screensaver mode (which I'll explain in a minute) will stop, allowing you to enter your own text. Now, keep pressing backspace until hello is gone, enter your own text, and press Enter. Your word now will appear in translated form above or at least a part of
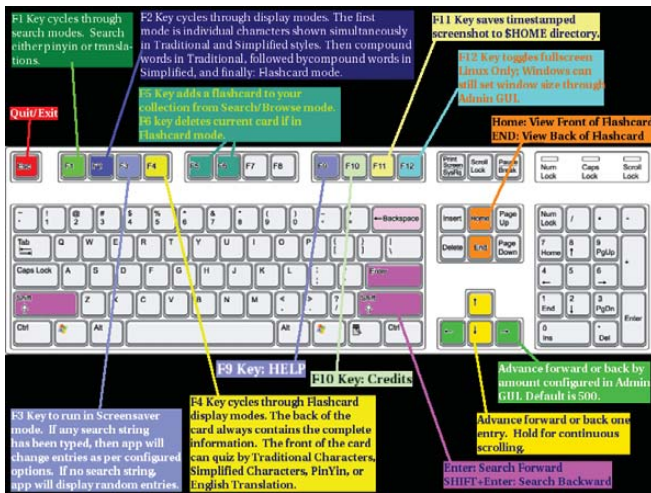
Taken from the Web site, this keyboard reference is an invaluable tool.

it in another English word (for instance, "car" may be inside "careless" or "carriage" and so on).

If the displayed translation isn't the one you want, press Enter again, and the next translation will be displayed. Keep pressing Enter, and you can scroll through all translated possibilities. Note that SIC accepts only American English. As an Australian and like most of the English-speaking world, I actually speak British English (this is an American publication; I don't normally spell this way), so entering something like "mum" or "colour" turned up with nothing. So if your entry doesn't produce any results, it might be that you're not using American English.

Let's move on to the screensaver mode. This is where the traditional flashcard system comes into place. Press F3, and all of the possible translation options will cycle through the display at a default rate of ten-second intervals. If you want to use SIC as a screensaver of sorts, pressing F12 puts SIC in full-screen mode.

As for Chinese characters, you probably will want to switch between Chinese Traditional and Chinese Simplified, so press F2 to do so for the standard display mode, and then press F4 for screensaver mode.

So far, I've dealt only with English text input, and SIC isn't restricted to that. You also can enter words in Chinese to display their meaning, albeit in the Romanized Pinyin form (actual Chinese text input is a much more complicated process in a non-Chinese OS setup, and this is aimed at foreign international users after all). Pressing F1 switches the search mode's parameters between English, Pinyin Chinese or Unicode searches.

That's about all I have space for this month, but ultimately, my time with Step Into Chinese has been a pleasant one. Although the interface is initially confusing, once you've been introduced to the basics, operating SIC actually is quite a simple process. I also like how the Asymtopia folks have shown some artistic flair in using fonts that are funky and colorful to make a pleasant working environment. SIC is highly recommended.∎

John Knight is a 25-year-old, drumming- and climbing-obsessed maniac from the world's most isolated city—Perth, Western Australia. He can usually be found either buried in an Audacity screen or thrashing a kick-drum beyond recognition.

Nokia N900

In a world full of powerful Android devices, is the Nokia N900 the best Linux computer you can fit in your pocket? KYLE RANKIN

The first time I thought about the N900 was about a year and a half before it was ever released. I was at the Penguicon conference, and some friends and I had settled down at a table to continue the time-honored tradition of chatting with each other on IRC even though we were a few feet away from each other. I noticed a friend take out a Nokia N800 and a Bluetooth keyboard and proceed to chat and browse the Web as though it were a laptop. I was amazed someone had managed to put together a smaller computer than I had, which made me start to reconsider what an ultraportable computer really was.

I always have preferred my portable computers to be, well, portable. My very first laptop was a Toshiba Libretto 50CT, and it was about the size of a VHS tape if you are old enough to know what those are. Since then, all of my laptops have been in the ultraportable category. As I looked at my friend's N800, I concluded that the main things missing were a bit more speed and cellular connectivity. After all, when the wireless was spotty at the conference, many of us were tethered to our cell phones or air cards. I decided if Nokia's next version of the Nseries was fast enough and had a cellular connection, it would be the device for me.

It should come as no surprise that when Nokia announced the N900, I was immediately intrigued. Would this be the new portable Linux computer I was looking for? Also, as a sysadmin who shares on-call duties, did I still need to drag my laptop around when I was on call, or could I do everything from this device? I have been fortunate to be able to spend a few months with an N900 Nokia provided me, and in this article, I review the N900 from the perspective of a longtime Linux geek who wants to know whether it can give me a small and open Linux device that can replace a Netbook or even a laptop for portable computing.

Hardware
One of the things that could make or break the N900 as a Netbook/laptop replacement is its hardware. As you'll see, the N900's hardware is a hybrid between a Netbook and a smartphone. Here are some of the main technical specs:

- 600MHz ARM Cortex-A8 (like in the iPhone 3GS, Palm Pre, Droid, Beagleboard and Open Pandora).

- 256Mb RAM plus 768Mb swap on Flash.

- 3.5" 800x480 resistive touchscreen.

- 32Gb included storage + empty MicroSD slot for up to 16Gb more.

- 802.11b/g, quad-band unlocked GSM cellular connection and FM transmitter.

- Bluetooth and microUSB port with tethering support over both.

- 5 megapixel camera, 3x digital zoom.

- Integrated GPS with AGPS.

If you are in the US, although the N900 is unlocked and can be used on any GSM network, if you want 3G speeds, you will have to use T-Mobile, as the N900 doesn't support AT&T's 3G frequencies.

The box also includes headphones, power adapters and a TV-out cable, but unfortunately, it's only a composite cable so you won't get the full 800x480 resolution of the native screen. Although it tends to work okay for a movie or a video game, when I used it to show slides during a presentation, the text was a bit difficult to read.

Finally, like the N810 before it, the N900 includes a slide-out hardware keyboard suitable for thumb typing. The keyboard itself is a bit small and has only three rows of keys and an offset spacebar that can throw you until you get used to it. But, the keyboard feels pretty solid, and after using it for a few weeks, I can type fast enough to keep up in IRC. If you want to do extended typing on the N900, I recommend investing in a larger keyboard. Unlike the prior Nseries, there is no USB-host mode, so you'll need a Bluetooth keyboard.

The Maemo Linux Distribution
A number of portable devices have similar hardware specs to the N900, but what makes it different is the software. Like the Nseries devices that preceded it, the N900 runs Maemo—Nokia's Debian-based Linux distribution. Maemo really feels like any other Debian-based distribution only optimized for a small touchscreen. Although you have custom desktop widgets, panels and application management, behind the scenes, you'll find standard Linux systems like apt-get, X, pulseaudio, upstart, dbus, gconf and sudo. Also, it's simple to get root on Maemo without hacking firmware or voiding warranties. Simply install the rootsh package that's available in the default repositories, and type `sudo gainroot` to get a root shell.

Maemo's desktop environment feels a lot like a slimmed-down, touchscreen-friendly GNOME desktop. The desktop is one screen high and up to four screens wide, and you can touch the screen and drag left or right to switch between desktops. Figure 1 shows a somewhat tweaked version of a Maemo desktop featuring shortcuts to some favorite applications, but

you also can put shortcuts to contacts, bookmarks or various widgets on the desktop as well. Widgets act much like plasmoids or other desktop widgets and let you manage the media player, check the weather, check news or Twitter feeds, or do other similar tasks.

Figure 1. A Maemo Desktop with the Matrix Theme

Along the top of the desktop is a bar that acts much like a panel on GNOME or KDE. On the top left-hand corner is a button you use to switch between desktops, an application launcher screen and an Expose-like application switcher that shows all of your currently running applications (Figure 2). This last feature is particularly well done and really makes the N900 stand out for its multitasking abilities. You also can press the Ctrl-Backspace shortcut keys to present the application switcher. It is very easy to switch between different programs.
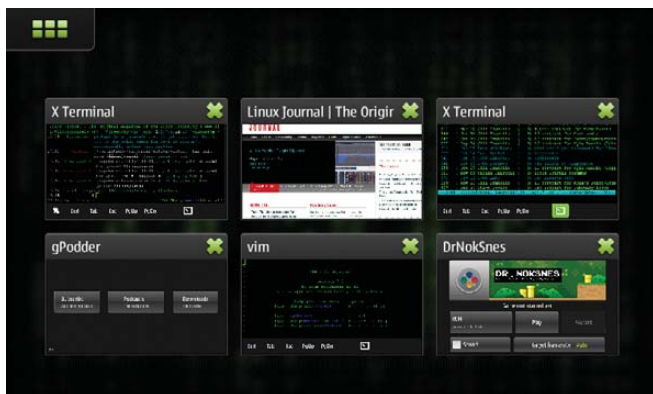
Figure 2. The Application Switcher with a Few Favorite Programs Opened

Contact Integration
The contact list integration on the N900 is one of its most compelling features. From a user-interface perspective, there is little difference between sending an SMS, sending an instant message, making a cell-phone call or calling someone via Skype or VoIP. All of these communication methods can be integrated into each contact in your contact list and are managed from a central settings page. There also are packages to extend IM support to all the major IM protocols. Once you sign into an account for the first time, those contacts are added to your contact list, so you can merge

them with any existing contacts.

For instance, if I set my Skype account to be on-line, people in my contact list who were currently on-line would have a little green dot next to their contact, and if I had a contact shortcut on my desktop, it would be there as well. If I want to IM or call people via the Skype network, all I have to do is open their contact and click that particular option. If any of my friends send me an SMS or an IM, it shows up in a standard IM window. If they call my cell phone or Gizmo number or make a Skype call, the N900 rings the same way in each case.

I've not mentioned the phone functionality much because, honestly, I rarely use a cell phone for actual phone calls. I hesitate to refer to the N900 as a phone at all, because although it certainly can make phone calls, it's really more of a portable computer than a phone. The phone feature seems to work fine for me, but if you come to the N900 primarily to make phone calls, you will discover that it's a portable computer first and a phone second or third.

Figure 3. An Example Contact Screen with a Few of My Contact Options

Browser
A good deal of my time on a computer involves a Web browser. Compared to many other portable devices, the browser on the N900 is among the best I've seen. Its MicroB browser is essentially a slimmed-down version of Firefox and renders pages like you'd expect in a regular full-sized browser without the need to load a "mobile" version of a page.

Of course, most non-optimized Web sites do appear a little small on the 3.5" screen. If you double-click on a section of a page, the browser zooms in so that section fills the width of the browser. I really liked this automatic zooming, especially for two- or three-column Web sites with content in one column and navigation bars on the sides. When you zoom in on an article, the rest of the navigation zooms out past the edges of the screen, so you can focus on what you want to read. You also can zoom with the hardware volume keys at the top of the device or make a clockwise or counter-clockwise gesture on the screen with your finger.

The browser currently supports Flash 9.4. I've been able to view regular streaming video fine, and full-screen mode works like you'd expect. Certain Firefox add-ons, such as Adblock and Greasemonkey, also have been ported to the MicroB browser already, but because of the differences between MicroB and standard Firefox, your favorite extension may or may not work out of the box.

For average sites, the CPU seems to provide adequate performance and even will render sites like Facebook, Google Calendar and Google Wave fine. I've noticed that some JavaScript-heavy sites, like the non-mobile version of Google Reader, do lag on the browser, and sites that use AJAX to provide their own scrollbars (like Google Reader) can be a bit of a pain to navigate. Overall though, I found I could browse the Web like on any other browser. To me, this is a great advantage to the N900 versus other mobile devices, as there has been a trend to write custom applications for various sites (like Facebook) to overcome poor performance on mobile browsers. On the N900, you don't really need a custom Facebook application, as the regular site works fine.

Terminal

When I'm not using a Web browser on a Linux machine, I'm in a terminal, so the N900's terminal capabilities were important to me. The terminal is installed by default, so that's one mark in Nokia's favor. Also, there's a somewhat hidden shortcut of Ctrl-Shift-x that launches a terminal automatically. Granted, it's a bit tough to press that keyboard combo, but the fact that one exists at all shows that this environment doesn't ignore people who like to get their hands dirty on the command line.

As you start to use the terminal, you'll notice that many of the common keys you might like to type in a terminal are not on the hardware keyboard. For instance, to get |, < or >, you have to press a special function key to bring up a list of special characters. The terminal does provide a shortcut bar along the bottom of the window with some common keys, so you can tap those on the touchscreen, but I just couldn't believe that | was left out. As with most things on the N900, you can tweak this shortcut bar. All of the settings for the terminal and many other applications are in gconf, so to add a pipe, I just needed to run two commands:

```
gconftool-2 --set --type list --list-type=string
 ➥/apps/osso/xterm/keys '[Tab,Escape,Page_Up,Page_Down,bar]'
gconftool-2 --set --type list --list-type=string
 ➥/apps/osso/xterm/key_labels '[Tab,Esc,PgUp,PgDn,|]'
```

If you want to switch out or add other keys, it's just a matter of changing those two gconf entries.
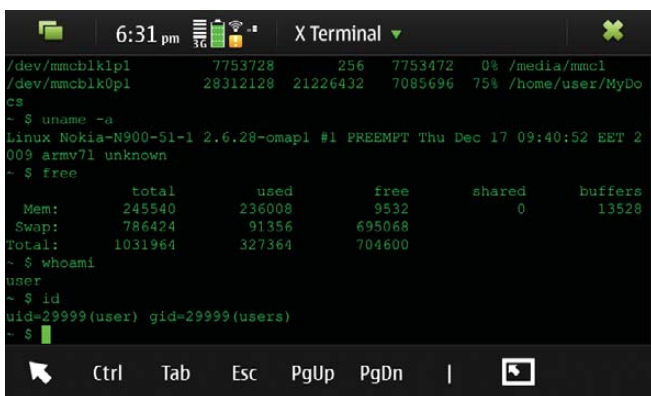

Figure 4. Default Terminal with Font and Shortcut Bar Tweaks

By default, the N900 terminal uses a BusyBox shell and vi,

but if you want bash and vim, they are only an apt-get away. After the initial tweaks, I've been pretty satisfied with the terminal. I can ssh easily into a remote server and run mutt and connect to my irssi screen session. If you do this though, you'll find yourself looking for the missing Alt key to switch between irssi windows. It turns out that Esc works like Alt in standard terminals, so to switch to window 2 in irssi, just press Esc 2. In fact, that should work in your regular desktop terminal as well.

Other Default Applications

The N900 comes with quite a few applications out of the box, and although I don't cover them all here, a few are worth mentioning. One of the most important of these is probably the package manager. This program acts much like any other graphical package manager you might be used to on Linux. Maemo packages everything in debs and provides its own APT repositories. In fact, if you are comfortable with apt-get, you can install the same programs via the command line. You also can add additional repositories, including ones you've made yourself if you want, but out of the box, you have access to Nokia's own repositories along with the stable maemo.org repository.

Maemo.org is the central site that organizes the Open Source community behind Maemo and is loaded with documentation, news and the main community forum. Many of the most popular applications from prior versions of Maemo have been ported to the N900, although you might not see them all out of the box. Although the main maemo.org repository is on by default, only applications that have been throughly tested show up there. If you are willing to risk some instability, you can add Testing and Devel repositories that somewhat mirror the Debian testing and unstable system. Applications that are shown to be stable in Devel move up to Testing and eventually are promoted to the main stable maemo.org repository.

The N900 also includes a media player, photo management, calendar and e-mail program. None of these applications stand out, but they all seem to do an adequate job. The media player can take advantage of hardware acceleration, and with the right third-party packages, you can get support for most media formats you'd want to play. Although the calendar application can't natively sync to Google Calendar, you can somewhat work around that with its native Exchange sync feature, even though you still can sync only your first Google Calendar. The good news is that most of the limitations you might find in these default programs can be solved with third-party software available from the maemo.org repositories.

GPS navigation is the one main exception to being able to solve application shortcomings with third-party programs. I found the Ovi Maps application included by default to be a bit sluggish and unintuitive and overall wasn't too impressed. Unfortunately, although a few other third-party GPS programs are available for the N900, none available at the time of this writing seem to be able to provide a better set of features than the default.

Interesting Third-Party Applications

Maemo's software selection may not have the numbers that Android or the iPhone have yet, however, in the short time that the N900 has been available, quite a large number of useful programs are already in the repositories. Also, the

Maemo platform provides you with more options as a developer, and you can write programs in C, C++, Python or even bash and with either the GTK or Qt graphics toolkits. The maemo.org repositories are where the most interesting and useful applications are, and it's where the N900 really shows some of its strengths as a platform. Here are a few third-party apps I've personally found useful:

- OpenVPN: open-source VPN to tunnel into my home network from anywhere.

- gPodder: excellent podcast program.

- BlueMaemo: turns the N900 into a Bluetooth keyboard, mouse and gaming pad.

- VNC and rdesktop: connect to and control remote desktop sessions.

- fbreader: great ebook reader with Project Gutenberg integration.

- DrNokSnes, iNES and SDLMAME: SNES, NES and MAME emulators.

- qtirreco: uses the IR port to turn the N900 into a universal remote.

- *Duke Nukem*, *Doom* and *Quake III*: enough said.

- Panucci: media player optimized for podcasts and audio books.

Beyond these programs, many interesting small packages extend the functionality of the N900, whether it's adding extra IM protocol support, remote control via wiimote or applications like MPlayer, nmap or even AbiWord. In fact, one of the most interesting programs is easy-deb-chroot. It sets up a complete Debian ARM chroot image on your filesystem and gives you access to a regular Debian ARM install. This means that even though GIMP, for instance, hasn't been ported to Maemo yet, it, OpenOffice.org, Konqueror, Wireshark and just about any other app from the Debian ARM repositories are available on the N900 and can run from within the chroot environment.

Day-to-Day Use
So after all this, what is it like to use the N900 every day as a portable computer? No matter what, a laptop with a faster processor, bigger screen, more RAM and larger keyboard probably is going to be nicer and easier to use than any portable counterpart. Of course, the majority of the things I use a computer for don't really *need* a large screen, fast processor or full keyboard. Ultimately, I'm talking about trade-offs and whether the limitations in the N900's size and hardware are made up by its features and portability.

The first thing you have to keep in mind is that even though hardware acceleration takes care of some things, CPU-intensive programs still are going to perform as though they are on a 600MHz processor. The N900 still handles multitasking in this circumstance better than other devices I've seen, but when the CPU is hammered (like when I update my

podcasts or applications), GUI transitions stutter, and sometimes it takes a second or two to switch programs.

The browser itself works well, so if you spend a majority of your time on the Web, you'll probably find the N900 does a good job. The display still is a 3.5" 800x480 screen, so even though it's crisp and bright, it's not as nice as a 12" or 15" laptop display. Because each program shows up maximized and it's easy to switch between open windows, this is a manageable problem, but if you squint on smaller screens, you might want to check out an N900 in person first to make sure you can see the screen fine.

The keyboard is not too bad, but it does take some getting used to. I would have liked another row of keys, but honestly, when you are thumb typing, no matter how things are arranged, you aren't going to hit your touch-typing speeds. I can chat at decent speeds, and it works fine for other short-term typing, but I'm not going to write full articles on the N900 without a Bluetooth keyboard. Also, because you have to press Fn key combos to get to most symbols, working with vim or programming is a good deal slower.

In many ways, the N900 is like a laptop in that its battery life can vary widely depending on how you set it and what you do with it. If I take basic common-sense steps for power management, such as adjusting the brightness and turning off any vibration or sound notifications I don't want, I can get a full day's moderate use out of the N900 on a charge. That includes listening to a few hours' worth of podcasts, browsing the Web on and off, playing some games, connecting to a remote screen session over SSH and chatting with irssi, and other regular use. Obviously, if I play a lot of *Quake III* or do other tasks that peg the CPU or network for long periods, the battery takes a hit. As long as a portable device can last through the day with normal use, so I can charge it at night when I'm asleep, that's good enough for me.

One warning about battery life though. I've noticed that some IM plugins can have a dramatic effect on battery life. Also, one reason that some software is in the Testing or Devel repositories is that they haven't been optimized for the N900 yet and might cause significant drain on your battery.

Conclusion
So, does the N900 live up to my expectations? Before this device, I took my laptop to and from work every day, and it was with me wherever I went—especially if I was on call. Since I've been evaluating the N900, my laptop has stayed at home so far unless I'm giving a presentation. Even when I'm on call, I've found between the VPN support, SSH, VNC and rdesktop, I can manage all of my servers from anywhere. Even when I'm at home, half the time I just want to do basic tasks like browse the Web, check e-mail and chat, so I don't bother to open my laptop—I just use the N900. When I telecommute or write an article, the laptop is more comfortable, but I've found I use it much less otherwise. I also should note that when the time came to send back the review unit, I bought an N900 of my own. Having a real Linux computer in my pocket with always-on Internet access was just too hard to pass up.■

Kyle Rankin is a Systems Architect 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.

Introducing
L2Ork:
THE LINUX LAPTOP
ORCHESTRA

Using Linux and a few extra parts, the Linux Laptop Orchestra
from Virginia Tech is music to our ears, in more ways than one.

DAVE PHILLIPS

When I searched Google for "define: orchestra", I received some interesting responses. The first definition I found says "a musical organization consisting of a group of instrumentalists including string players", which describes a rock band as well as the Boston Pops. Another defines an orchestra as "an instrumental ensemble, usually fairly large with string, brass, woodwind sections and possibly a percussion section as well". That one is closer to the popular conception of an orchestra, but my favorite definition is "a group of musicians led by the conductor who accompany the singers", a clarification sure to ruffle the feathers of many an orchestral instrumentalist.

The definition of a laptop orchestra requires similar flexibility. Modern computers make excellent music machines, whether for common media playback or for the specialized purposes

of music composition and audio synthesis. Even inexpensive laptops, notebooks and Netbooks are capable of high-quality real-time synthesis, giving them an apparently limitless variety of sounds. The machine is well described as a meta-instrument, an instrument that subsumes the sound palette of traditional instruments and adds new sonic possibilities of its own. An ensemble of such machines might be described as a meta-orchestra, an orchestra of orchestras or simply a laptop orchestra.

In this article, I combine a project profile, an interview with the team leader and some musings on the significance of the world's first Linux Laptop Orchestra, aka L2Ork (pronounced "lork"). The L2Ork is not the first group of its kind, but it is the first such group based on Linux system and application software. The laptop orchestra has arrived.

History

Today's laptop performance groups have evolved from aesthetic concepts and performance practices that date from the early days of the personal computer. In the late 1970s, musicians and hackers around San Francisco formed the League Of Automatic Music Composers. The League's many fascinating projects included one of the first "network bands" based on the newly available KIM-1 microcomputer. Eventually, the League disbanded, but members John Bischoff and Tim Perkis formed another network band to perform what they called Hub music. The Hub's performances were successful enough to keep the performing group active for ten years.

In 1985, the arrival of MIDI revolutionized the entire music world. For the first time, music could be composed, performed and recorded with an inexpensive single computer, without the need for instrumentalists and at comparatively little expense. Unfortunately, MIDI has certain drawbacks for networked ensembles, including a relatively slow transmission rate bound to a serial data flow. Nevertheless, ensembles of computers running MIDI software could network within the limitations of the specification, and even The Hub experimented with MIDI connectivity.

By the late 1990s, desktop computers were powerful enough to perform real-time audio synthesis while simultaneously processing MIDI data. At the start of the 21st century, the same measure of raw power could be found in lightweight low-cost portable machines, and the development of OSC (Open Sound Control) removed some of the restrictions imposed by MIDI. Among its virtues, OSC accommodates a much wider variety of data types and requires only common networking hardware, such as Ethernet cabling and connectors, giving it a natural fit to designs for computer-based orchestras.

The OLPC (One Laptop Per Child) Project deserves mention in this context. A wireless LAN of OLPC laptops can be bound to one machine's clock for synchronizing the group, giving any group of OLPC machines the capabilities of a laptop orchestra. However, those capabilities are restricted by the relatively low power of the machine itself.

In spring 2008, the prestigious *Computer Music Journal* published two papers describing the activities of a laptop orchestra led by Ge Wang and Perry Cook at Princeton University. The descriptions and analyses of the Princeton Laptop Orchestra (aka PLOrk) provided the starting points for subsequent laptop-based orchestras, such as the Stanford Laptop Orchestra (SLOrk), and eventually found their way into the able hands and mind of Ivica Ico Bukvic, a professor of music at Virginia Tech and the founder and current director of the Linux Laptop Orchestra there.

The Princeton Laptop Orchestra defined the form and basic potential of the laptop ensemble, but it relied upon the combination of a powerful open-source programming language (Ge Wang's ChucK) with a powerful closed-source application (Cycling 74's Max/MSP) running on proprietary hardware (Apple PowerBooks and MacBooks). Alas, those closed and proprietary aspects can drive the costs of such an ensemble beyond the budgets of many music departments and other interested organizations. Fortunately, the next logical step was taken by the Linux laptop orchestra. Max/MSP can be replaced by Pd and

other free software, the restriction of vendor-specific hardware can be removed by the use of inexpensive commodity machines and input devices, and even the cost of the speaker arrays can be reduced significantly. During our interview, L2Ork leader Ico Bukvic stated that the current cost per seat was about $750 and that the cost included the computer and the speaker system.

And Then There Was L2Ork

The success of the Princeton and Stanford groups inspired Ico Bukvic to form a similar group at Virginia Tech. However, his vision of the laptop orchestra would take the franchise into a new direction. Ico wisely adopted the Princeton group's general design and organization, but his Linux background compelled him to improve the ensemble's affordability and accessibility.

By 2007, Ico had founded the Digital Interactive Sound and Intermedia Studio (DISIS) at Virginia Tech with the intention of "fostering cutting-edge research as well as innovative approaches to interactive multimedia art genre", to quote from the DISIS Web site. With such an infrastructure in place, Ico concluded that DISIS would be a natural fit for the world's first Linux-based laptop orchestra.

Early in 2009, Ico received an internal grant from the Institute for Society, Culture and Environment at Virginia Tech, in collaboration with his colleague Dr Tom Martin, Associate Professor of Electrical and Computer Engineering. By the spring, the project had acquired greater support from Virginia Tech as well as from external sponsors that include the Sweetwater company, MSI Computer and the Roland Corporation. At that time, the DISIS developed its own design for a hemispherical speaker and began assembling the various other hardware and software components needed for the debut of the Linux Laptop Orchestra in the fall—all told, a fair amount of achievement for a year's work. Since then, the group has expanded to include Dr Eric Standley (Professor of Art) and seven undergraduates.

In December 2009, the Linux Laptop Orchestra made its official world debut in the Squires Studio Theater at Virginia Tech (Figure 1). According to eye-and-ear witnesses, the house was jammed, and the performance was a great success. Critical coverage was positive, and the event even made it to the hallowed pages of Slashdot. Clearly, fame had come to the L2Ork.

Figure 1. **L2Ork Debut Performance**

The Hardware

Let's take a closer look at L2Ork's physical plant. The orchestra is currently configured for up to 15 players. Each player operates a basic system that includes an MSI Wind Netbook (Figure 2) with a Cakewalk UA-1G USB audio interface, Nintendo Wiimote and Nunchuk controllers, and a custom "hemipod" speaker array. The hemipods are connected to three shared subwoofer systems.

Figure 2. MSI Wind Netbook with Custom Software

The L2Ork FAQ page notes that the decision to employ the Nintendo controllers resulted from a need to replace the Apple machines. The Princeton group exploits accelerometer hardware built in to the MacBook and PowerBook, but the Wiimote and Nunchuk controllers have proven to be a happy choice. They are well supported by Linux, are sufficiently sensitive for deployment as musical controllers, and their gestural geometry includes a Z-axis. Also, everyone has fun playing with them.

The integrated audio hardware on most laptops is insufficiently powerful for the typical needs of a desktop audio production system, much less for the needs of a member of a laptop orchestra. The L2Ork's Wind Netbooks are connected to the UA-1G, a USB audio interface with

Figure 3. L2Ork Hemipod

Figure 4. The Hemipod Disassembled
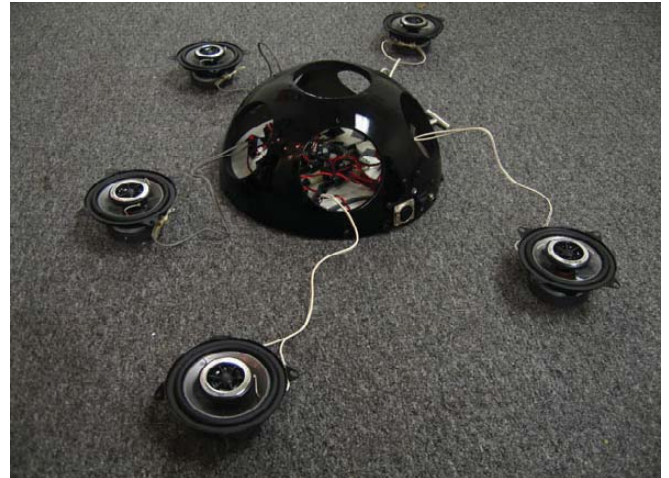
stereo I/O supporting sample rates up to 96kHz and bit depth to 24 bits. The unit includes decent analog-to-digital and digital-to-analog converters and provides RCA and optical connections for S/PDIF digital I/O. I was surprised to learn that MIDI is not a feature of the UA-1G, and I suspect that at some future time, the L2Ork crew will want to add MIDI connectivity to their network arsenal.

An Interview with Ivica Ico Bukvic

I spoke with Ico Bukvic on the hard-line and later asked him for a question-and-answer session to clarify some of the points we discussed. I must note that we had a hard time finishing our chats, the subject matter is simply too exciting and provocative.

DP: Are the Netbooks networked with each other or are they standalone units?

IB: Networking is a critical component of the ensemble, as this is something truly unique to the laptop orchestra genre. Hence, all machines are networked at all times, allowing composers to take advantage of this great opportunity to build monitoring tools, coordinate the ensemble and even alter the ensemble's properties in real time, based on performers' actions.

DP: What Linux distribution are you using?

IB: Currently, we are using a modified version of Ubuntu 9.04 and have just begun migrating slowly to 9.10 with the ultimate short-term goal of eventually moving to the 10.04 LTS release.

DP: Are you exploiting the capabilities of the real-time kernel, low latency and JACK?

IB: Absolutely! Our current 9.04 setup runs the 2.26.29-rc6-rt3 real-time kernel, which we compiled in-house. With our upcoming migration to 9.10, we've been testing the one provided by Ubuntu Studio and have been quite pleased by its performance in our tests so far.

DP: Besides Pd, what other application software is used?

IB: Currently, all works written specifically for L2Ork rely upon Pd or our customized version of Pd-extended. As things settle a bit, we are definitely looking forward to incorporating ChucK and SuperCollider, which will enable us also to perform works written for other *Orks, and other DSP software composers may end up utilizing works written specifically for our ensemble.

DP: Is being a musician or programmer helpful/necessary/preferable/required for members of the band?

IB: While having musical background, performance experience, and being proficient with technology certainly helps, this in my opinion is only a fraction of the overall L2Ork experience. If anything, I would argue that L2Ork offers a uniquely level playing field, engaging students regardless of their background and/or current area of study.

Faculty Professors Eric Standley, Ivica Ico Bukvic (with pod) and Tom Martin.

Perhaps this is one of its greatest creative strengths.

DP: Has L2Ork performed music written for other laptop ensembles?

IB: Not yet, mainly because we have been so busy exploring our own aesthetic identity that we've simply had no time to do so. This is, however, definitely on top of our to-do list, and I am hoping we will be performing other *Ork repertoire before the end of this year.

DP: What notable new possibilities present themselves to the composer writing for a laptop orchestra?

IB: I think L2Ork's versatility is at this point probably more limited by our imagination than its infrastructure, and I suspect this will remain true for some time to come. After all, at the heart of L2Ork infrastructure is a laptop, which by its very nature can take on so many different roles. This is something I find to be tremendously exciting, and this is what attracted me to the laptop orchestra concept in the first place.

Imagine having an ensemble where each performer's action, apart from producing sound, also alters someone else's instrument. What if these interdependencies were evolving, changing over time in a way one could not even begin to fathom, cascading complexities that may arise from such treatment of a musical material. Or, what about having a group of L2Orkists *not* making sound collaboratively sculpt a 3-D sculpture whose shape is also affected by the sound produced by the

As you can see in Figure 3, the hemipods are definitely eye-catchers. An amplifier and six speakers are mounted within a hemisphere enclosure for a radiant 6-channel sound dispersal (Figure 4). Each seat in the group has its own pod, and each pod is controlled by its single user. The pods are potentially the most expensive component to the orchestra, but the L2Ork group has published complete directions for users who want to cut costs by building their own arrays.

The Software

L2Ork's Netbooks run a mixture of off-the-shelf free software and some custom control programs (also free) written especially for the orchestra. Where the Princeton group deployed the Max/MSP software, the L2Ork group uses its free-software doppelganger, Miller Puckette's Pure Data, more affectionately known as Pd. Both Max/MSP and Pd present a graphic interface to their myriad functions, which include numerous synthesis

rest of the ensemble and whose resulting properties somehow feed back into other domains, affecting instruments, lighting, or theatre, or something else. On a more conservative side, just having full awareness of what everyone else is doing within the ensemble can be used in all kinds of cool and experimental ways to shape a piece's structure and consequently its drama. For instance, a composer may request that a certain group of performers maintain a certain amount of activity, which is generated by averaging actions from everyone within that group. As material becomes more abstract, this quickly becomes an invaluable tool to monitor the overall output. And the list goes on as far as one's imagination can muster.

DP: What notable challenges—administrative, technical, musical, personal—have you encountered while directing L2Ork?

IB: Let's just say that working with L2Ork is as trying as it is exciting. I think to be able to truly attain understanding of and control over L2Ork, one has to be comfortable with juggling many different roles: being an engineer, computer scientist, user-interface designer, composer, performer and so on. Often, these roles tend to overlap, and I found this to be particularly exhausting.

Last semester must have been one of the busiest semesters I've had since I joined Virginia Tech, if not in my entire academic career. I probably would never have bothered if I did not think that the potential benefits weren't worth it. In this respect, I feel L2Ork did not disappoint.

DP: What are some of your long-range goals for L2Ork?

IB: I am not sure if we have any big long-term plans, in part because we are still so overwhelmed with the existing possibilities that we've had little time to think five or ten years ahead. Another consideration is the pace at which technology changes. Who knows, perhaps in five years the entire ensemble will have technology embedded in their clothing.

Near-term, however, my focus is on growth both in numbers and in repertoire, touring and strengthening ties with our peer *Ork ensembles, encouraging Linux community participation, forming new Linux-based *Orks and, of course, ensuring that we have adequate support to ensure the initiative's long-term well being.

DP: Anything or anyone else you'd like to mention?

IB: I would like to hereby extend my deepest gratitude to those who have made L2Ork possible. This includes the entire Linux community and particularly the members of the Linuxaudio.org community, our stakeholders, sponsors, fans and supporters, as well as all the students for their efforts and dedication and for having been brave enough to invest their time into something so experimental.

and audio processing routines. Pd is powerful, easy to learn and extensible by its users. This last aspect is of particular importance when designing interfaces usable by performers who may not be literate in either Pd or Linux itself. Pd accommodates a wide variety of I/O devices (such as the Nintendo controllers), includes functions to provide easy access to network connectivity, and supports OSC and MIDI. A further list of Pd's capabilities is beyond the scope of this article, but it
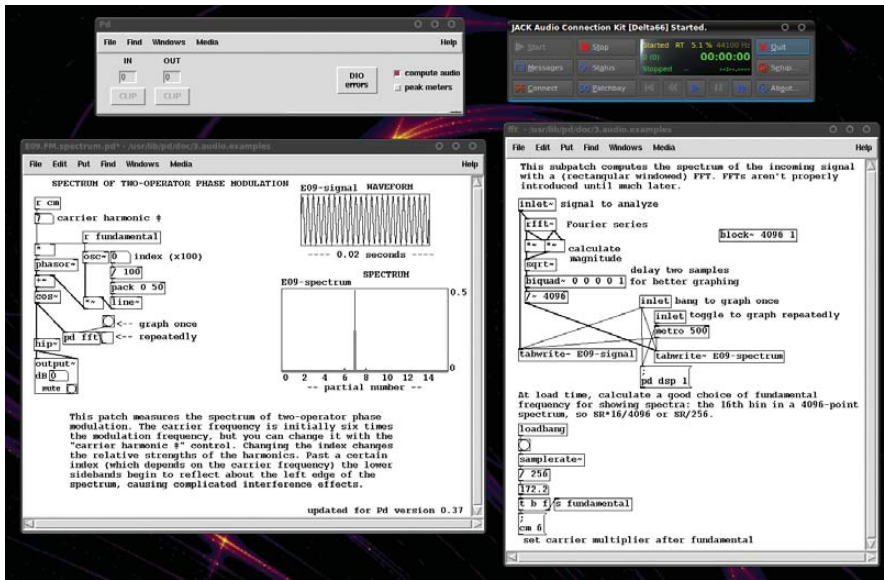
Figure 5. A Pd Patch for Audio Synthesis

must be said that Pd is a "perfect fit" for the laptop orchestra on a budget. The suggested retail price for the closed-source proprietary Max/MSP is a cool $495, while the total cost of Pd is $0.

Pd's integrated graphic components aren't bad (Figure 5), but they are less than shiny, especially when compared with the program's commercial counterpart. However, thanks to Pd's extensibility, the L2Ork group gets to look at some more sophisticated custom GUI components.

Ico Bukvic is no stranger to programming a useful and attractive GUI. While watching the L2Ork performance videos, I thought I recognized a variant of RTmix, a program he wrote when he taught at the University of Cincinnati. RTmix provides visual and audio cues for performers working with computers. It also can schedule other computer-generated activities and events—a handy tool indeed for the laptop ensemble.

Playing in the Band

I was curious to know what academic disciplines were represented within the group of players. I could assume the presence of at least some musicians, and I expected a few comp-sci types, but I was surprised to learn that in fall 2009 the group consisted of one music major, three majors in music technology and one representative each from biology, theater arts, engineering and political science.

Players without performance backgrounds may be reluctant to "let go" as they learn how to manipulate sound with the Nintendo controllers. As they progress, they lose their self-consciousness and begin to realize that they are exploring new performance techniques. Enthusiasm takes hold as the players learn how to create predictable (and unpredictable) effects by moving the controllers in defined arcs and to definite points in the gestural space. Given the early stage of the modern laptop orchestra, a standard repertoire of performance techniques is still being researched and cataloged. However, it is a goal of rehearsals to ensure that each member of the group is aware that his or her movements have specific outcomes for the

piece being played. In the same manner as a traditional instrumentalist, the L2Orkists learn how to play their instruments with the control of a rehearsed technique. In other words, they're not just waving their arms around. Well, actually they are waving their arms around, but every movement is defined and directed. And, that brings us to the role of the conductor.

In a traditional orchestra, the players are led through the score of the music by the conductor. To the untrained eye, conductors may seem to be doing little more than waving their arms around. In fact, conductors are responsible for many musical functions. They need to know the score in intimate detail, to know its parts for every member in the instrumental groups, and have a working knowledge of each instrument and the ability to instruct players in terms specific to their instruments. Conductors are human mixers, the one component of a group who plays no instrument yet commands them all. Conductors are responsible for timbral and amplitude balances, and the interpretation of an entire work is performed according to their conception of how the piece should sound. Every artist has a unique personal interpretive sensibility; thus, we have a multitude of recorded versions of Beethoven's symphonies.

In the laptop orchestra, the conductor's role subsumes traditional functions and adds new possibilities. With the appropriate software, the computers in the ensemble are perfectly capable of performing without a conductor. In point of fact, they can do without the performer too, but that scenario is not of interest at the moment. As you can see in the videos, the L2Ork behaves like a traditional orchestra—that is, the performers play their instruments under the guidance of the conductor. However, other scenarios easily present themselves, and those other possibilities may be of compelling interest to a composer.

The Composer's Corner

See the Resources for this article for some links to music performed by various laptop orchestras. It doesn't take long to realize that just about any style of music can be written for and performed by a laptop group, although Ico advised me that if the composer wants a screaming lead-guitar solo over a manic heavy-metal bass and drum pattern, that composer would be better off with a rock band than a laptop orchestra. Yes, a laptop ensemble is extremely flexible, but composers still must consider whether it is the appropriate vehicle for their music.

Ico Bukvic's "Citadel" is an excellent example of a piece written specifically for the L2Ork group. It is a beautiful work, and while listening to it, I wondered if it could be played by a traditional group. I concluded that it could be arranged for such a group but that it probably would lose something in the translation. By comparison with some other compositions for laptop groups, Ico's piece is a relatively conservative work, one that was likely intended to give

his players a gentle introduction to a piece written in a modern musical idiom. Other recordings show off the less traditional, more experimental possibilities. Of course, it is unnecessary to choose between extremes, and composers for the laptop orchestra have access to a vast palette of sounds and performance techniques that they can bend, fold, staple and mutilate to their hearts' content.

The Author Muses

The computer can be used to create new sounds from synthesis primitives (oscillators, envelopes and filters), from recordings of existing sounds (sampling, analysis/resynthesis and granulation) or from mathematical models of the physical characteristics of sounding materials (physical modeling). It can follow a set of instructions that tell it when to do these things, and with the appropriate control software, the machine can decide for itself where, when and what to play during a performance. But, exactly how does someone "play" a computer? The generality of the machine works in favor of multiple solutions to the problem of the performer's interface. The common input devices—keyboard, mouse, joystick, touchpad, tablet, touchscreen—can be used, although typically, they have no tactile sensitivity. MIDI-based solutions include keyboards, guitars, percussion pads and wind controllers, all of which have the advantage of being familiar designs for musicians trained to play traditional instruments. The USB port provides connectivity with pressure-sensitive devices and other sensors whose data streams can be mapped to synthesis parameters and score events. Composers can specify any or all of these devices as needed, or they may design a completely new interface.

Given the protean nature of the instrument and its networking capabilities, computer-based ensembles can take on many forms. The machine can be programmed to function musically as a single instrument or an ensemble of instruments, as a standalone performer or as a member of an ensemble, with or without an operator. These distinctions are not exclusive—a soloist may be accompanied by an ensemble in a concerto-like piece—and the composition of a group is subject to fluid redefinition over a network. Unlike an instrument in a traditional orchestra, the machine needn't have a single permanent identity.

A human instrumentalist performs many related tasks when reading a score or improvising. Ensemble players learn to master their individual parts by attending to technical issues and by responding musically to the dictates of the score. Within the group, there is a further exercise of sensitivities. Each player becomes responsible for the balance of his or her part within the group, and everyone's performance is understood to serve a particular interpretation. Modeling the complexity of the musical and social group dynamics of a performing ensemble presents formidable difficulties, particularly when we want the machine to interact with a musically convincing improvised response to another player's improvisation while following and interpreting the guidance of a conductor (human or otherwise).

Frankly, the musical mind boggles at the possibilities presented by the laptop orchestra. Ico and I joked about creating a game called *Orchestra Hero* for the group, but

why not? In the domain of education, the possible applications of the laptop orchestra could be most attractive, especially as costs rise and budgets shrink. I have no desire to see traditional instruments replaced by computers, but I'd love to see the computer become even more integrated into school music programs at all levels.

The Future Beckons

I hope you've enjoyed this introduction to the Linux Laptop Orchestra. Of course, you really need to watch and listen to the group in concert, so be sure to check out the videos and other recordings. Check out the whole *Ork scene while you're at it—you'll be watching the cutting edge in action. If you are sufficiently inspired, you even can put together your own Linux laptop orchestra. The plans are freely available, the costs are low, and the software solutions are free and open source. All that's needed is a little do-it-yourself spirit, a little hardware hacking and Linux on the inside.■

Dave Phillips has been using Linux for sound and music since 1995. He is one of the original founders of the Linux Audio Developers/Users groups and has been the maintainer of linux-sound.org for more than ten years. He is the author of *The Book Of Linux Music & Sound* and has written many sound-related articles for various Linux publications. His other activities include playing in a blues band, reading Latin literature, playing with his shar-pei Maximus and spending time with his beloved Ivy. You can hear Dave's music at linux-sound.org/ardour-music.html.

Resources

DISIS, Home to Ico Bukvic's Digital Interactive Sound and Intermedia Studio: **disis.music.vt.edu/main**

The L2Ork home, the main site for information regarding the L2Ork Project. Links to recordings, performance schedules, history and FAQ: **l2ork.music.vt.edu/main**

The L2Ork YouTube site. Connects to a rehearsal video and an in-depth series of instructional videos for building the hemipods and other L2Ork hardware components: **www.youtube.com/user/VTDISIS**

The League Of Automated Music Composers, an excellent history, complete with images and recordings (highly recommended): **crossfade.walkerart.org/brownbischoff**

One Laptop Per Child (OLPC): **laptop.org/en**

Princeton Laptop Orchestra: **plork.cs.princeton.edu**

Pure Data (Pd), an excellent introduction to the awesome capabilities of Pd (highly recommended): **puredata.info**

Slashdot on L2Ork: **entertainment.slashdot.org/story/09/12/03/2018253/Introducing-L2Ork-Worlds-First-Linux-Laptop-Orchestra**

Stanford Laptop Orchestra: **slork.stanford.edu**

Comparing
MythTV
and XBMC

MythTV and XBMC can turn your Linux system into a video playback device with all the options you've ever dreamed of.

MICHAEL J. HAMMEL

The Linux desktop is a place to work and a place to play. People fortunate enough to have time to play have many distractions to amuse them. Audio fun abounds with programs like Amarok and Rhythmbox. Video enthusiasts will find a plethora of options, from standalone video players (like MPlayer, Xine and VLC) to comprehensive multimedia environments (like Freevo and Moovida). Being a video enthusiast, I find myself drawn to two of the most popular open-source entertainment tools: MythTV and XBMC.

MythTV (**www.mythtv.org**) is a full-featured digital video recorder built for the Linux desktop yet perfectly suited to running as a home-theater system. It includes support for live TV and recordings, video file and DVD playback, photo galleries, music collections, weather forecasts, movie rentals via Netflix and even home video surveillance.

XBMC (**xbmc.org**) is a media player that also supports DVD and video file playback, music collections, photo galleries and weather forecasts. Unlike MythTV, XBMC was not originally

Figure 1. MythTV can be run as a distributed system, with front and back ends on different computers.

developed for the Linux desktop. Instead, it was built to run on modified Xbox hardware, and later it was modified to be a general-purpose media tool and ported to the Linux desktop. Because of this, some of XBMC's features are geared toward the use of Xbox features, such as running games and dashboards. Even so, XBMC has evolved into an excellent desktop Linux media player in its own right.

MythTV and XBMC are similar tools but have different designs and

Figure 2. XBMC is a media player on steroids.

target audiences. This article doesn't attempt to compare them side by side, apples to apples. Instead, the intention is to examine each from a user perspective and discover what features have meaning to different types of users. Because these applications are so feature-rich, the primary focus here is limited to video services—playing movies and watching TV. Although both systems have been designed to work well when displayed on a TV, this article is written from the point of view of using the applications on a desktop.

Installation and Requirements

MythTV is a well established project and, as such, is easily installed on most major Linux distributions using desktop package management tools. Ready-made packages are available for Fedora/Red Hat/CentOS, Ubuntu, Debian, OpenSUSE, Mandriva and Arch Linux. There also are live CD distributions, such as MythBuntu, MythDora and LinHES (formerly KnoppMyth), which allow you to run MythTV without installing it. Building from source is possible, but it can be complex, and there are many prerequisites. Using prepackaged versions is the preferred installation mechanism.

MythTV can be used without TV capture cards. The MythVideo plugin can be used to view and manage video files that have been ripped from DVDs or other sources. However, MythTV requires a supported TV capture card to allow watching live TV. It currently does not provide direct support for external TV sources, such as Hulu or Veoh.

MythTV back ends start automatically when installed from distribution packages. To start front ends, use the `mythfrontend` command. You can run other command-line tools, such as `mythfilldatabase`, although these often run automatically once MythTV is configured.

XBMC is built specifically for the Ubuntu distribution, and packages are available from the Ubuntu repositories. Other Linux distributions must build from source. Fortunately, doing so isn't hard, at least for Fedora. As with MythTV, you need to install many prerequisite packages. The XBMC Wiki provides copy-and-paste commands for installing them. Some distributions may require setting some environment variables, but building the package is the same for all listed distributions:

```
./bootstrap
./configure
make
make install
```

To install to a program-specific directory, specify a prefix. A prefix allows installation to a directory that can be removed easily later:

```
./configure --prefix=/opt/xbmc
```

To start the program, run `bin/xbmc` from the installation directory, such as /opt/xbmc/bin/xbmc.

Design Overview

MythTV is designed as a personal video recorder (PVR) application. It is a network-based system with back ends used to acquire and serve TV and other media to front ends that handle display and playback. It supports a wide variety of hardware for both analog and video capture (Figure 3). The core recording and playback system is extended with plugins that provide additional features. The MythVideo plugin handles management and playback of video files, such as DVD rips. MythTV is best known for its wide support of TV tuner hardware and ability to play and record both analog and digital TV.

MythTV's core focuses on watching and recording TV as well as providing extensive program guide and scheduling support. Some plugins extend this feature, such as the Archive plugin, which allows saving recorded programs to DVDs or other files.

Because of XBMC's Xbox roots, it focuses on being a multifunction media player, capable of many types of media playback from local or remote sources. This focus also allows the development team to emphasize the user experience. XBMC is best known for its well designed and sophisticated user interface and skins (Figure 4).

XBMC's core focuses on video playback, but it also is extensible through the use of Python plugins. Currently, plugins are available that add support for TV and movie guides, instant messaging and front ends to other
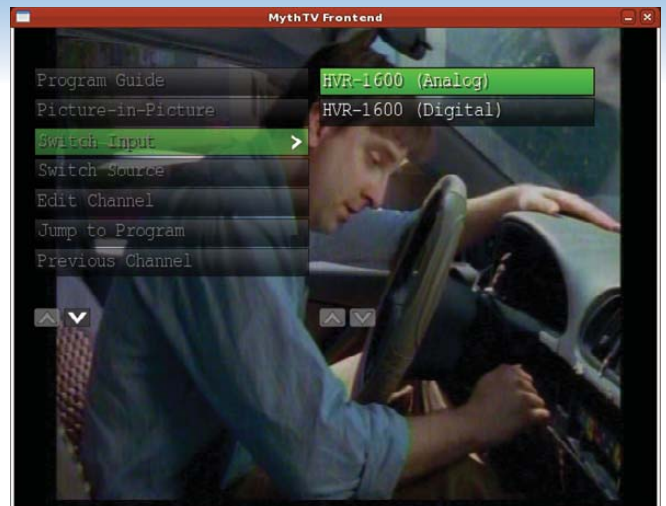

Figure 3. MythTV back ends support both analog and digital (HDTV) video capture cards.
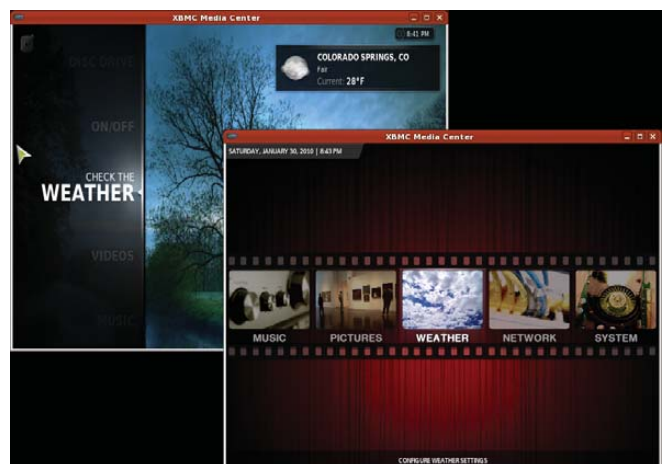

Figure 4. XBMC skins are plentiful and impressive.

hardware like a TiVo. And, XBMC forms the basis of at least one commercial venture, Boxee.

Video Management

Browsing and playback of video files, such as ripped DVDs, in MythTV is handled by the MythVideo plugin. MythVideo manages video files under a single configurable folder. Folders under the top-level folder can be mountpoints for multiple hard drives. Each folder can have its own icon (folder.png) that is displayed while browsing the collection (Figure 5). Videos can be streamed from the back end or played by the front end from an NFS mount. Over an 802.11g wireless connection, streamed files play with fewer pauses/skips.

Figure 5. The MythVideo gallery view shows folder icons and is the easiest way to browse a large video collection.

Using mountpoints under the top-level directory offers multiple options on how to manage your hard drives. RAID configurations can be used to create one large storage space. Alternatively, multiple drives can be used for different genres.

Unlike MythVideo, XBMC is based entirely on the concept of *sources*. A source can be a local hard drive or NFS mountpoint or a remote server providing streamed data (Figure 6). XBMC supports multiple streaming formats and actually can be used to connect to a MythTV back end to watch TV and stream video.

Figure 6. The movies source is an NFS drive, but the MythTV source is a network connection.

To add videos to a MythVideo collection, copy files to the appropriate folder. MythVideo can play AVI, MPEG and even DVD ISO files (including menus). To reduce disk usage, 7GB DVDs often are ripped to AVI files, which can be as small 2GB without loss of quality. However, ripping like this typically loses the DVD menu options and possibly the subtitles. If you have the disk space, a DVD ISO copy is faster to create, and you won't lose any DVD features.

Videos can be added to local or network sources for XBMC. However, XBMC cannot stream MythTV movies. Instead, the MythTV movie folders must be mounted via NFS on the XBMC system, and the local mountpoint added as a video source to XBMC. Like MythTV, XBMC can play a large number of video file formats.

Organization and Browsing

MythTV provides three methods for video collection browsing. The first is Browse mode, and it allows you to browse the entire collection front to back. The page up and down keys make jumping through the list easy enough, although this method doesn't remember your last location. The second method is List mode, and it uses a tree listing to display entries. This listing benefits from a well structured hierarchy under the top-level video directory. Gallery mode is the last mode, and it uses icons for folders and movie posters for individual entries. This is visually appealing and also benefits from a well structured hierarchy for the video collection, although the number of entries in each folder displayed over the icons is rather distracting (Figure 7).

Figure 7. MythTV can browse sequentially or through a hierarchy.

You can give custom icons to folders in MythTV by creating a JPEG or PNG image in each directory. The size doesn't matter, although making them large enough to be displayed in high quality on the front end with the highest resolution may offer the best overall results. MythTV scales the image as appropriate.

XBMC approaches video organization a little differently. First, it provides two browsing modes: file and library. File browsing starts by showing the

available configured sources. Choosing one of those then presents a listing of options relevant to that source. For example, if one source is a set of videos on a local hard drive and another is a remote MythTV back end, the former lists the movies in a typical tree hierarchy and the latter shows features available from the server, such as live TV or recordings.

In file mode, the listings are similar in structure to a directory listing. The typical ".." parent directory lets you back up through the hierarchy. Library mode allows you to browse only recordings. It won't show sources. Instead, it shows you the list of scanned recordings. Moving up through the hierarchy typically involves selecting an up-arrow icon (Figure 8).

Figure 8. XBMC's Library mode displays scanned recordings, but File mode can access any remote source.

Like MythTV, folders under XBMC can have custom icons. Place a folder.jpg file in each directory that requires a custom icon, and it will be displayed in either mode. If you share

Figure 9. Transparent PNG folder icons are supported by MythTV, while XBMC uses JPEG format icons.

the directories between MythTV and XBMC, using JPEG will mean only one icon is needed. However, JPEG images don't support transparency, which means converting a PNG folder icon with transparency from MythTV to a JPEG for XBMC will lose the transparent effect (Figure 9).

In either mode, the listings can have multiple views. Library mode offers List, Full List and Thumbnail views. File mode adds Wide Icons and DVD Thumbs.

Both applications allow browsing by genre. Genre information is set automatically when metadata is retrieved for a video. Both systems can stack files. Stacking associates multipart movie files as a single movie.

Metadata Editing

Information about videos is known as metadata. The metadata includes movie title, runtime, cast, synopsis and a variety of items similar to what is found on IMDB.com or TheMovieDatabase.org. Both applications can retrieve metadata

automatically when new files are added to folders, and both allow you to edit this metadata while browsing the collection (Figure 10). XBMC offers multiple configurable sources for metadata on a per-folder basis. MythTV's metadata sources are dependent on configurable scripts, but only one script can be configured, and it applies to all files.

XBMC sets metadata via the context menu. Right-click on any folder to get the menu, and then select Set Content. Choose a source for the metadata, and then run an automated scan. XBMC will let you add a movie that is not found manually, but once you've done this, there is no

Figure 10. MythTV (left) can edit individual movies. XBMC (right) works only on folders of movies.

easy way to remove the entry if it's wrong. In MythVideo, pressing I over a movie in any browse mode will bring up a context menu. From there, you can edit the metadata data, reset it, play the movie or just view more details about the movie.

Video Playback

Once you're past the setup and management, you finally can play your videos. And, this is where the two systems shine. Both offer high-quality playback of files in AVI, MPEG4 and even ISO formats. ISO offers the best viewing, because all of the DVD features are available, including menus and subtitles. Of course, both programs also will play a DVD directly from a local DVD drive. MythVideo plays videos with no embellishments—the full window is the video. XBMC actually can wrap its menus around a playing video and will play a video in a small window while the user browses other features (Figure 12). XBMC also offers the usual media controls on-screen. MythTV uses an on-screen menu, but media controls are supported by keyboard keys without the aid of on-screen controls (Figure 11).

Figure 11. MythTV's on-screen display handles most functions except media control while playing.
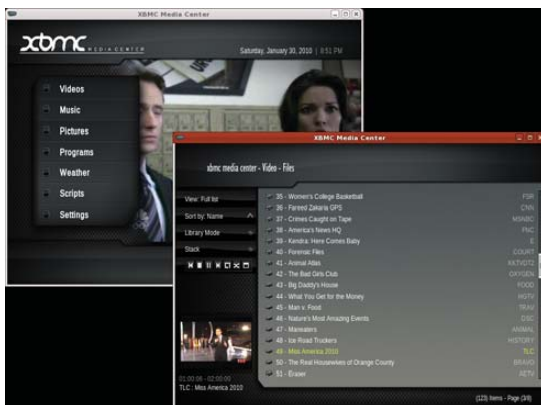
Figure 12. XBMC can wrap menus around playing video (left) or place it in a small window (right).

TV Guide, Playback and Recordings

From the user perspective, TV playback under both systems is very similar to video playback. XBMC places TV as another source under its Video feature. MythTV separates videos from TV, although TV recordings still are under the TV function. The main difference between MythTV and XBMC is the way the guide is displayed. MythTV has a built-in guide that is updated by a back-end server (Figure 13). The guide is accessed from the OSD menu while watching live TV or when scheduling a recording.

MythTV can record live TV from any configured hardware. The more TV tuners installed on the back ends, the more sources for recordings. When one tuner is busy recording, another can be used to watch live TV. You can schedule recordings using any of the numerous guide search mechanisms, and you can start recordings while watching a show. You also can view recordings even while it is still recording, as long as some recorded data is available. Recordings are given priorities so that if they conflict and not enough video sources are available, the higher priority gets access to the device.

There is only one way to watch recordings, and MythTV uses a tree structure for finding the recordings. Migrating recordings from MythTV's TV feature to MythVideo requires archiving them first via MythArchive. This is not particularly user-friendly, so keep your expectations low for hanging onto recordings.
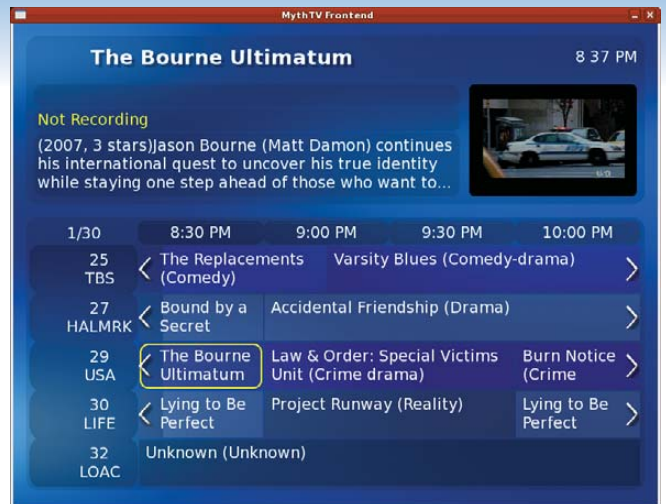
XBMC uses a MythTV back

Figure 13. The MythTV guide is color-coded by genre.

end to show live TV. It also uses MythTV's guide information. However, it shows the guide information using the same view types as videos in File mode. XBMC, however, cannot record data. It can play only live TV or view existing video files. It cannot create new video files from live TV.

Appearance and Intangibles

Both systems support the VDPAU extensions for NVIDIA, which provides high-quality playback. Because both support OpenGL, they also provide video controls, such as brightness, contrast and color controls (Figure 14). The actual features available under MythTV depend on various configuration options for both video and TV features. XBMC always has the same options available from any video playback, be it video files or live TV. These controls were available for an NVIDIA-based display. Other video hardware may present different options.

MythTV uses upfront configuration of window sizes and doesn't allow scaling the display while you watch. This isn't a problem once you're set up to display on a real TV, but if you watch on a desktop while you work, it's more of an issue. XBMC can display full screen, as does MythTV, but it also allows playing in a window that can be scaled while video is playing.

Neither system will stream Internet video, so you can't get to Hulu or Veoh from them. There are hacks for doing this, but they mostly just launch the Hulu player as an external application.

Figure 14. Video controls are available in both programs.

Both systems are fairly stable. XBMC crashed a few times during my research, and I experienced a few lockups, although the latter may have been due to some interaction with the MythTV back end. MythTV crashes are very seldom. During my research, it crashed once while experimenting with the use of subtitles while playing a DVD ISO file.

Summary

The XBMC experience with its polished skins is very entertaining. But, without a TV source, you still need a MythTV back end to watch TV. If you watch TV while you work, using XBMC as a TV front end to a MythTV back end is ideal, because you can scale the XBMC window interactively. MythTV may be your only choice if you need to watch, schedule and record multiple sources at once.

Both tools do an admirable job with video file collections. MythTV provides more flexibility in setting metadata on a per-movie basis, while XBMC has attractive browsing options and supports more metadata sources. But, don't expect to play Blu-ray disks any time soon with either tool. For one thing, they produce huge files (50GB typically), if you can rip them at all.

This is far from a comprehensive review of MythTV and XBMC. Both offer far more features and capabilities than can be given justice in a single article, especially if your goal is to hook them up to a real TV and use them as home-theater systems. If you enjoy using your Linux system for media playback, you owe it to yourself to investigate both of these excellent applications.∎

Michael J. Hammel is a Principal Software Engineer for Colorado Engineering, Inc. (CEI), in Colorado Springs, Colorado, with more than 20 years of software development and management experience. He has written more than 100 articles for numerous on-line and print magazines and is the author of three books on The GIMP, the premier open-source graphics editing package.

Full Speed Ahead with
Handbrake

See how Handbrake, a powerful and easy-to-use video conversion program, can help you enjoy your favorite TV shows or films while at your computer, whether you're at home or away from home. **ANTHONY DEAN**

As an animation and comic-book fan, I like to watch various TV shows and movies featuring my favorite characters, ranging from the *Simpsons* and *Superman* to *Looney Tunes* and the *X-Men* films. Like most such fans, I've also bought the various DVD releases of these shows. However, since I'm not always at home, I also like watching TV programs and movies on my laptop or Palm Pre smartphone. It's great to be able to watch films while on vacation and other trips, or cartoons while commuting on the bus to work. Instead of repurchasing digital download versions of these programs, however, I find it's much cheaper (and easier, when factoring in the lack of Linux-friendliness of some on-line digital video stores) to convert my DVDs into more versatile, computer-ready video files. Doing this gives me such advantages as being able to store my video collection in an easily accessible format (without having to swap DVDs from my bookshelves), being able to convert the files for playback on various devices (such as my Pre or my old iPod), and keeping a backup of my DVD collection. Although various video conversion programs are available for Linux, the most user-friendly one I've found is Handbrake.

Handbrake, a cross-platform program (with versions available for Windows and OS X along with Linux), comes in both a command-line version and a GUI version for Linux. An open-source program, Handbrake offers the ability to convert from unencrypted DVD video sources (including VIDEO_TS folders, .VOB and .TS files, and .ISO images) and most multimedia files that FFmpeg can handle, including AVI (.avi) and MPEG (.mpg) files. Output formats include the Matroska (.mkv) and MP4 (.mp4) video containers; H.264, MPEG-4 and Theora video codecs; and AAC, MP3 and Vorbis audio codecs (along with AC3 passthrough).

Installation

Installation of Handbrake itself is straightforward. If Handbrake isn't offered in your system's package manager, the Handbrake Web site's downloads section offers you the choice of installing either a command-line version or a GUI version, for either 32-bit or 64-bit systems. For the purpose of this article, I cover the GUI version, which keeps with Handbrake's emphasis on user-friendliness. If the .deb or .rpm files offered (geared toward Ubuntu and Fedora users, respectively) aren't sufficient, another option is to download the source code and compile Handbrake.

Note that Handbrake does not unencrypt encrypted DVDs on its own, which includes most commercial DVDs available. Before installing Handbrake, make sure the libdvdcss library is installed first. The legality of converting DVD video for one's personal use may vary by locality, so check with the proper authorities before proceeding.

From DVD to Laptop

To cover the basics of Handbrake, let's use it to convert to a more laptop-friendly format DVD of the hit 2000 superhero film *X-Men* about Marvel Comics' classic mutant-powered superhero team. After inserting the DVD into the computer (and launching Handbrake), select the Source button (Figure 1).
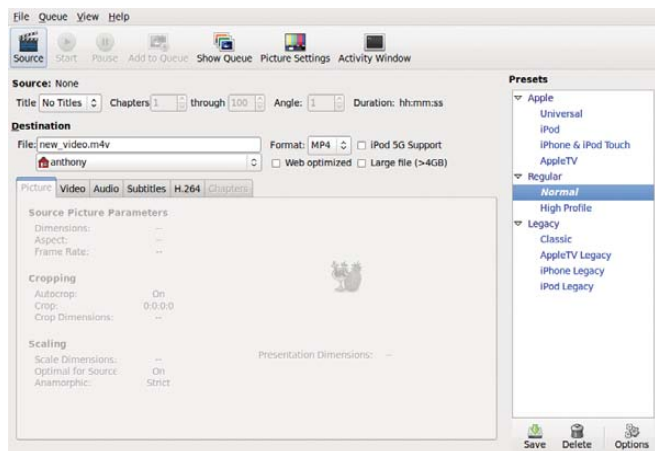
Figure 1. Handbrake's Default Interface, before Loading a DVD

If the DVD's directory isn't selected automatically, navigate to it; afterward, select OK, and Handbrake will read the DVD's tracks. The movie's available tracks will be displayed in the drop-down Title menu, next to the Chapters beginning and ending boxes and an Angle box. The movie itself generally will be the track with the longest time listed. Select the movie's track, then under the File destination box, type what you want to name the movie. If you don't want to convert all the chapters from the original source (say, if you just want the opening title sequence or a particular scene), type in the Chapters boxes the desired range of chapters. The Angle box can be ignored by most users (including for our purposes here), given its infrequent use on DVDs.

The next step is to decide whether you want to use Handbrake's presets or your own customized settings. One of Handbrake's strengths is its various presets, which allow
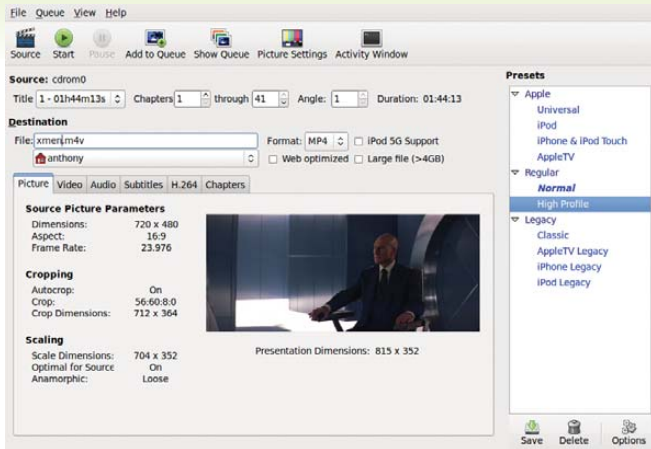
Figure 2. Handbrake after Loading a DVD

you to choose a convenient high-quality setting for various conversion purposes. Previous versions of Handbrake offered myriad presets, but as shown in Figure 3, version 0.9.4 (the most-recent version, at the time of this writing) has simplified the offerings for convenience and now include Normal (the default) and High Profile (high-quality) presets; presets for the iPod, the iPhone/iPod Touch and the Apple TV; and several now deprecated legacy settings, aimed at users of previous versions of Handbrake.

For my purposes (keeping the file to play on a laptop), I
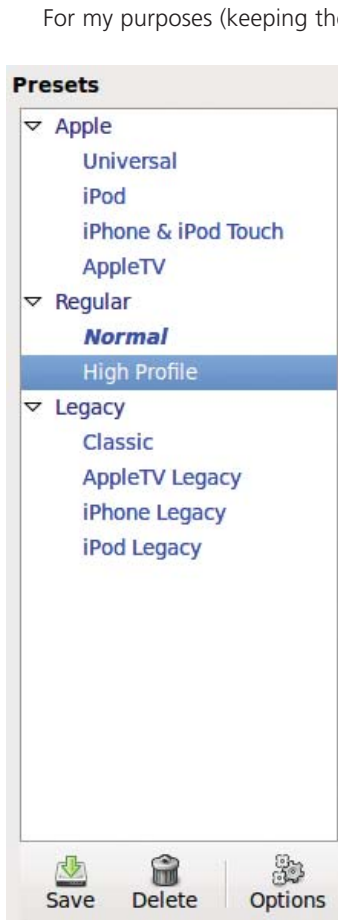
Figure 3. Handbrake's Presets

usually select High Profile, a high-quality setting, though I've also found the iPhone/iPod Touch preset helpful for making files specifically for watching on my Pre, which uses the same video settings as the iPod Touch and iPhone.

If you're fine with the presets settings as is, the next step is to select Start, and Handbrake will begin converting the video. However, if you want to convert multiple DVD tracks (say, the extras on a movie DVD or all the episodes from a TV show DVD), select Add to Queue first to add the current track, then select another track as previously described, and click Add to Queue. Repeat until you've chosen all the desired tracks from the DVD, then select Start.

If you're looking for finer control over the conversion settings, you need to do more than just

the above, of course. As shown in Figure 1, Handbrake offers several different tabbed panes to adjust various settings. The panes include Picture (the default displayed in Figures 1 and 2), Video, Audio, Subtitles, H.264 and Chapters. Let's look at what each pane offers.

Picture

This is the default pane shown when Handbrake is launched. When the desired video or DVD to convert from is loaded into Handbrake, a preview scene from the video is displayed, along with information on the video's dimensions, including cropping and scaling. Click on the Picture Settings button in the top bar to adjust the final video's cropping and scaling settings manually, as well as the detelecine settings (which are mostly used when converting TV programs from DVD).

I generally leave the dimensions settings alone, along with keeping decombing turned on when converting TV programs (and off when converting movies, although turning it off isn't necessary).

The Picture Settings window also can play a brief preview of what the final converted film will look like, based on the current settings.

Video

From the Video pane, two drop-down menus offer settings for the video codec and framerate. Video codec choices include H.264, MPEG-4 and Theora. There's also an option to turn on two-pass encoding (a setting that scans the video source twice to ensure a higher-quality result), as well as radio buttons offering a choice of converting based on either the bitrate, a specific targeted file size or a constant quality rate.
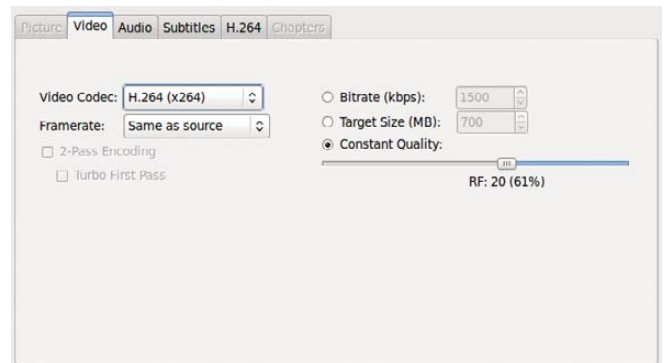

Figure 4. Video Pane

I generally keep H.264 selected (Handbrake's default setting) and leave the constant quality setting as is, unless I need a specific bitrate or file size. If so, I also turn on two-pass encoding and select Turbo First Pass, a setting that speeds up the rate of the first pass. If adjusting the bitrate manually, I usually choose a figure between 1,000–1,500kbps (though I lean toward the conservative side).

Audio

In the Audio pane, depending on the video source or the preset chosen, one or several audio tracks may be displayed. The number of tracks may be added to or subtracted from using

the + or - buttons shown. This is useful if you want to include, for instance, the director's commentary track from a movie or alternate languages (a potentially popular feature for, say, Japanese anime fans).
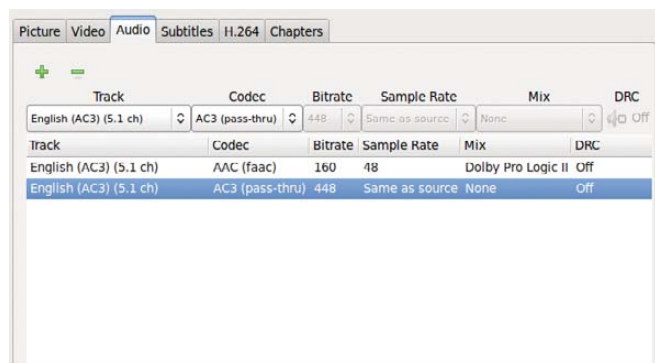

Figure 5. Audio Pane

To adjust the tracks even further, drop-down menus are displayed: Track, Codec, Bitrate, Sample Rate, Mix and a button for DRC (Dynamic Range Compression). Track provides the option of choosing which audio track(s) to encode, including foreign-language tracks or director's commentary. Codec allows you to choose to which audio codec to encode (depending on which video container is chosen): AAC, MP3, Ogg Vorbis or AC3 (the default audio on a DVD). Bitrate and Sample Rates allow you to set to which bitrate and sampling rates, respectively, to encode. Mix allows you to select output mixes: mono, stereo, Dolby Surround, Dolby Pro Logic II and 6-channel discrete. Finally, DRC allows (if activated) you to adjust the limits of compressing the dynamic range of the audio output. This setting would be used for audio with very loud or very quiet portions by altering the resulting audio to increase the sound on very soft portions and reducing the sound on very loud portions.

For the purposes of encoding my *X-Men* movie, I'd choose at least one track (in English, the only language I fluently speak or read), encoded to AAC with a bitrate of 160kbps, with a mix of either stereo (my very basic audio setup at home consists of just a pair of speakers) or one of the surround sound settings. I leave DRC and Sample Rate alone. If I opt for a second track (say, to include a separate track for surround sound if I ever upgrade my home audio setup), I'd choose AC3, which would encode the audio as is from the video source. I also might decide to include a third audio track for the director's commentary (so I can hear Bryan Singer expand upon some of the film's plot points). Keep in mind that adding extra audio tracks, depending on the settings, may make the resulting file larger.

Subtitles

The Subtitles pane allows you to include subtitles and closed captions in the output file, either from the original video source or from an external file. Although I very infrequently use closed captioning (not having hearing difficulties and very rarely watching non-English-language material), this option would be highly useful for someone converting a subtitled
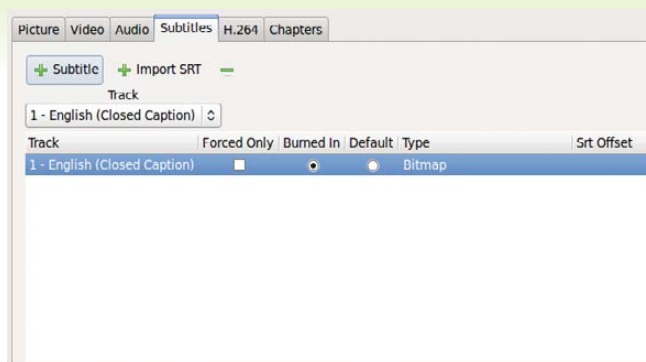

Figure 6. Subtitles Pane

video source.

The options in this pane include selecting one or more tracks (via the + Subtitle, + Import SRT and - buttons), then choosing a language for each track. Forced Only should be checked if you want the subtitles displayed only when the video demands it (say, when a scene has someone speaking a foreign language that must be translated and displayed on-screen for plot purposes). For this option, choose Foreign Audio Search from the track drop-down menu. Burned In usually will be chosen automatically if it's a subtitle track, placing the subtitles on the movie with no

way of turning them off. Closed captioning, meanwhile, doesn't use this feature, and it can be turned on or off in the resulting file's video player. There's also the option of choosing which track is the default (via the Default radio button). If you've chosen + Import SRT, then a different set of options will be presented, allowing you to choose which external file to use to import subtitles.

H.264

The H.264 pane allows Handbrake users to tweak various advanced x264 settings to their specifications. For the average user (such as myself), this pane can be left alone. Those interested in the settings here can consult the Handbrake Web site's user guide page, although as the page notes, it (at the time of this writing) hasn't been updated yet for 0.9.4.
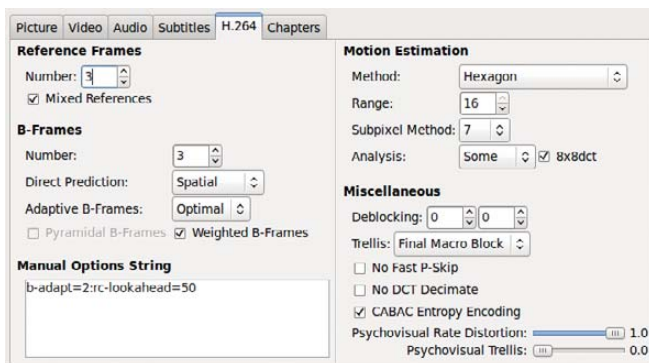
Figure 7. H.264 Pane

Chapters

The Chapters pane gives you the option of including chapter markers at the same places the chapter breaks exist on the original DVD. To activate this, select the Chapter Markers box. The length of each chapter is displayed, as well as generic titles reading "Chapter 1", "Chapter 2" and so forth. If you want to create specifically named chapter titles (such as "That cool scene where Magneto makes the guns hover" or "Statue of Liberty slugfest"), click on the chapter title of choice and type a new name.
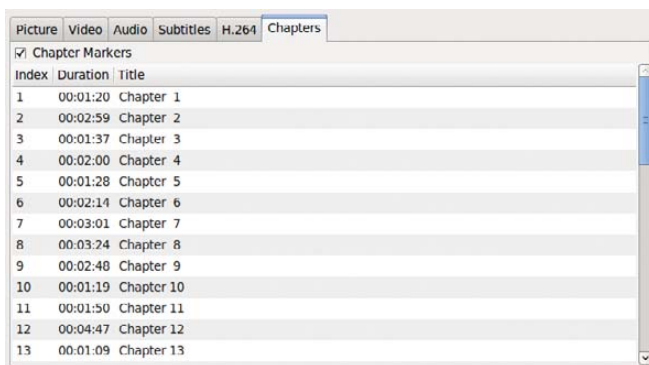

Figure 8. Chapters Pane

For my usage, I keep the Chapter Markers box checked, although I usually don't bother renaming the chapters.

Customized Presets

After performing all the previous customizing steps, you might want to save these settings for future reuse. To do this, click the Save button shown in Figure 3 at the bottom of the Presets pane, and give the desired settings a custom name in the box that appears. The new preset then will appear in the Presets pane, ready to use for future video conversions.

Length of DVD Video Conversion

After finishing all of the above, to begin conversion, click Start. For my computer, an HP laptop with 4GB of RAM and a 2.1GHz Intel Core 2 Duo T6500 processor, running Ubuntu 9.10 (64-bit), it took two hours and 43 minutes to convert the *X-Men* movie (which runs an hour and 44 minutes) using the High Profile preset. The end result of all this, of course, is that I now have a high-quality movie file that I can view (or use) as I wish.

Figure 9. Finished Video File Playing in Totem

Conclusion

As you can see, Handbrake is a great program for those wishing to convert their video media, particularly their DVDs, to a more versatile digital format. Thanks to Handbrake, I've converted much of my DVD collection for easy access and viewing on my laptop, allowing me to enjoy watching Hugh Jackman (or Bugs Bunny) in action while away from my TV set or away from my apartment entirely.■

Anthony Dean works as a file clerk for the city of Milwaukee, Wisconsin. He's been using Linux as a primary operating system since 2005. Anthony can be reached at adean33@gmail.com.

Resources

Handbrake: **handbrake.fr**

Handbrake Downloads: **handbrake.fr/downloads.php**

Handbrake's User Guide to x264 Options: **trac.handbrake.fr/wiki/x264Options**

Comparing Five Music Players

There are so many choices of music players for Linux, it's hard to know which one will work best for you.

BRUCE BYFIELD

On modern computers, music players are as standard as word processors and spreadsheets, but how do you choose one? You can take for granted that music players using the same soundcard and speakers will produce roughly the same quality of sound. You also can assume that any music player you try will be able to play any music format that your operating system supports, including, for GNU/Linux, the free Ogg Vorbis format. A modern music player will support various sources as well, ranging from the local collection of tracks on your hard drive to CDs, DVDs, external players and on-line sources, such as Internet radio and podcasts. So, how do you decide?
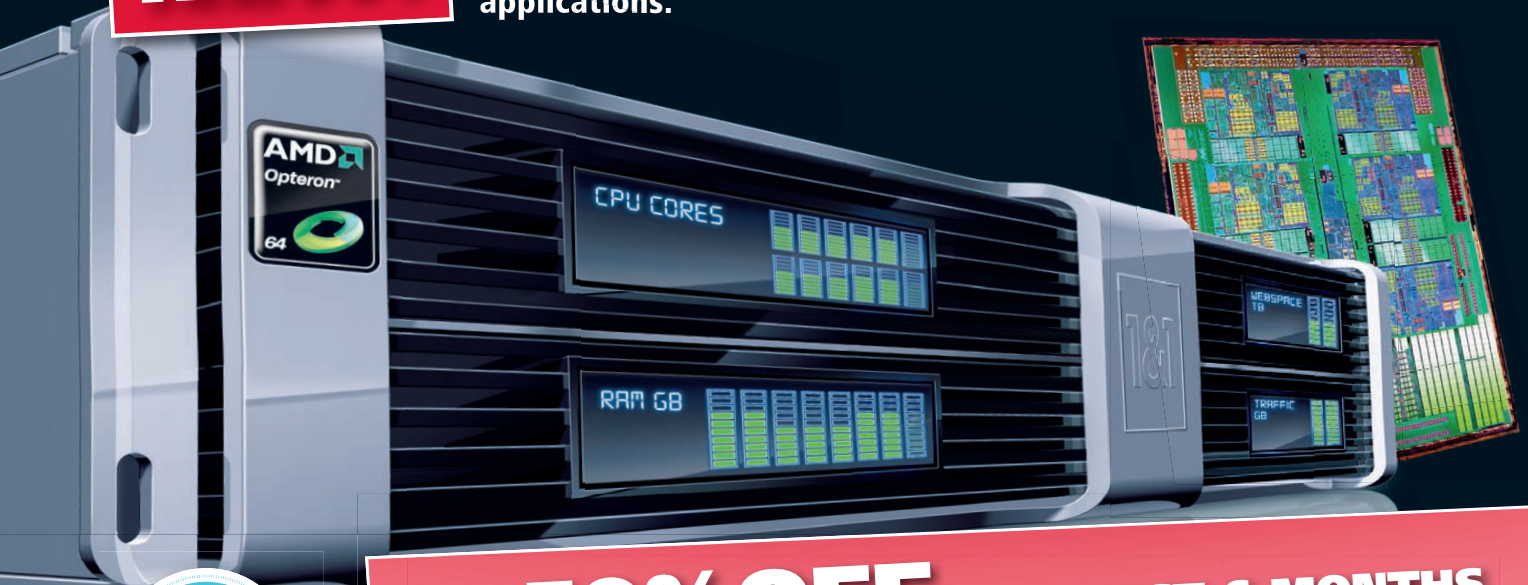
To suggest an answer, I looked at five of the most popular music players for GNOME and KDE—**Amarok**, **Banshee**, **Exaile**, **Rhythmbox** and **Songbird**—using the current versions available in the Debian unstable repository. After comparing them in each of six general usability categories, I ranked them and tallied the results.

Interface and Usability

Dragging and dropping tracks, desktop notifications and minimizing music players to the notification tray are all standard features these days. The differences in how each music player handles those features are usually minor, although Amarok, like most current KDE apps, provides the most customization for notifications. It also repurposes its middle context pane when you are moving tracks from the media source pane on the left to the playlist on the right by temporarily transforming it into live links that you can drop selections on to get different results.

However, the largest problem for all these music players is how they handle collections of local tracks and podcasts and music stores that easily can number in the thousands these days. Unfortunately, in four out of five cases, the handling of all this information is not well thought out.

The endless displays of tracks, albums, artists and playlists not only make for a cluttered window, but also can leave users with a feeling that they have too much information. At times, what controls actually do can be difficult to discover, as with the filters for Rhythmbox's search filter, which easily can be mistaken as controls for altering the panes displayed in the window. Too often, the space for each column in a pane is so limited and track or album names are so truncated in anything less than a full-screen view, they almost might not be listed at all.

The exception to this rule is Amarok, whose three main panes maximize the display space in the window while using every sort of trick—from expanded trees to hiding music sources not currently in use—to reduce the clutter and information overload. Exaile and Songbird manage some of the clutter in their default views with tabs, but Banshee and Rhythmbox both have a series of permanent panes that feel badly in need of cleanup. You can, of course, greatly improve the layout of all the music players via the View menu by selecting which panes or columns to display, but Amarok remains far ahead in general appearance. If you really want to remove the clutter, you can hide the middle context pane, reducing the information in Amarok to a functional minimum. Another possibility is to undock one or more of Amarok's panes to create a separate floating window that you can refer to only as needed.

Still, all five players do what they can to help users navigate. All can sort lists in ascending and descending order, and all include search filters, although Amarok gives you more control over both sorting and filters. In addition, Amarok and Banshee both offer bookmarks.

Amarok

After Amarok, the best-designed is Songbird, whose Web structure gives it an instant familiarity. Songbird also features skins, called feathers, and a zoom for changing the size at which information is displayed, but these features, although novel and convenient, are not enough to challenge Amarok seriously.

RANKING:

1. Amarok
2. Songbird
3. Banshee, Exaile and Rhythmbox (tie)

Editing Metatags

The metatags for tracks are essential to any music player, not only for display, but also for automating various features, such as CD ripping and suggesting music similar to the current track.

In all five music players, tags are lumped into statistics about how often each track is played, how the track is rated and background information, such as lyrics and cover art. Yet, despite the importance of tags, Rhythmbox and Exaile do not include a general Comment tag for miscellaneous information, and only Exaile and Amarok have any provision for custom tagging (and you easily can miss the Add button in Exaile).

An even larger omission is the fact that you can edit tags only by albums in Amarok. The reason may be that Amarok is designed to play music equally by track or album, while the other players are oriented toward tracks. Yet, even so, editing tags under albums means that you have to enter them only once instead of ten or more times.

Another useful feature for handling tags in Amarok is to guess them from the filenames. If you have ripped music from a source like K3B, which names files according to sequences of tags, this feature can eliminate drudgery even further. The feature is not always accurate, however, and it can cause trouble when illegal characters are used in a filename, yet, even so, anyone digitalizing masses of music will find it essential.

RANKING:

1. Amarok
2. Exaile
3. Banshee
2. Rhythmbox and Songbird (tie)

Automating Playlists

With all five players, you can randomize the order in which the tracks on a playlist are played. However, with Rhythmbox, that is as far as automation goes.

Banshee, Exaile and Songbird all have controls to create automated playlists based on criteria, such as the most played or the least played. In Banshee, two unique playlist controls are Recently Added and Unheard, while Songbird includes Artists on Tour, which selects from artists who are touring soon in your area. By contrast, Exaile's playlist controls are less useful unless you are planning for a party, as they include Top 100, Newest 100 and Random 100 and 500—a selection that sounds like an effort to re-create the Top 40 radio that many people originally turned to music players to escape.

Exaile is more to the point with its Smart Playlists, in which you can specify the criteria

Save the Date!

USENIX Federated Conferences Week
June 22–25, 2010, Boston, MA
www.usenix.org/events/#june10

USENIX ATC '10
2010 USENIX Annual Technical Conference
Wednesday–Friday, June 23–25
www.usenix.org/atc10

WOSN 2010
3rd Workshop on Online Social Networks
Tuesday, June 22
www.usenix.org/wosn10

HotStorage '10
2nd Workshop on Hot Topics in Storage and File Systems
Tuesday, June 22
www.usenix.org/hotstorage10

REGISTRATION DISCOUNTS AVAILABLE!

WebApps '10
USENIX Conference on Web Application Development
Wednesday, June 23
www.usenix.org/webapps10

HotCloud '10
2nd USENIX Workshop on Hot Topics in Cloud Computing
Tuesday, June 22
www.usenix.org/hotcloud10

And more!
Visit www.usenix.org/events/#june10 for new workshop announcements and registration information.
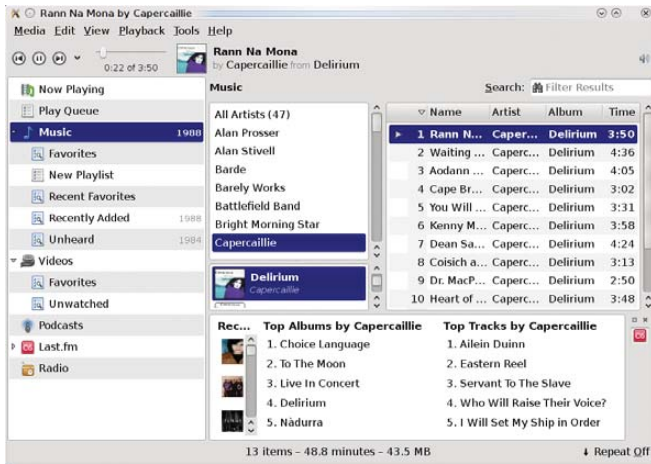
Stay Connected...

USENIX

www.usenix.org

http://www.usenix.org/facebook
http://www.usenix.org/linkedin
http://twitter.com/usenix
http://blogs.usenix.org

Banshee

for automated playlists. This feature is duplicated almost exactly in Songbird, but in my experience, it works only erratically in either Exaile or Songbird.

However, the most sophisticated automated playlists are found in Amarok. Amarok's Dynamic Playlists include three types of bias: Proportional Bias, in which playlists are forced to include set percentages of tracks that match the designated tags; Custom Bias, in which playlists include set percentages to match the tags; and Fuzzy Bias, which sets how much tags can vary from the tags specified. You can mix all of these biases to create playlists that match or exceed those in the other four players, and best of all, new lists are created with almost no delay when you click the Repopulate button.

RANKING:

1. Amarok, Exaile and Songbird (tie)

2. Banshee

3. Rhythmbox (lags far behind)

Context and Background

One disadvantage of digitalized music is that it lacks the background information you can get with a CD in the form of liner notes. Some tracks and albums use context tags to give information about producers and backup music, but this practice is relatively rare, and the information is not nearly as rich as with traditional media. Even the latest version of Amarok, in which you can add custom tags, you can't really re-create the information that comes with CDs via tags.

Instead, music players are starting to provide alternatives to liner notes using resources from the Internet. All five players can hunt the Internet for lyrics and covers, although with Rhythmbox, doing so requires enabling plugins. Exaile and Amarok also search for Artists and Albums, while Banshee bizarrely opts for a context pane that includes Top Albums and Top Tracks by the current artist—information that is minimally useful and more than a little imaginary the first few times you play tracks. However, taking a hard-core fan approach, Songbird outdoes the rest by searching for reviews, news, photos and videos.

The success of these efforts to provide extra context depends partly on the artist and album. In every case, the effort is more likely to be successful if artists are popular or at least have a cult following that increases their chances of being listed in Wikipedia or on fan sites. Disambiguation pages also can be a problem.

Even more important, searches can be long and may end with no results. Searches in Songbird for reviews and news often take longer than a single song, while in Amarok a long lyric search can freeze the track controls, although the track keeps playing. However, on the plus side, context features sometimes can produce unexpected benefits for the true fan, such as alternate record covers or (as I once found for The Pogues' "Fairytale in New York") unrecorded versions of the lyrics.

RANKING:

1. Songbird

2. Amarok

3. Exaile

4. Banshee

5. Rhythmbox

Scripts and Plugins

In free software, music players are second only to Firefox and OpenOffice.org in their efforts to extend functionality by creating a community of plugin writers. With each player, many of these plugins automate specific Internet sources. Other plugins alter the interface or add features.

Rhythmbox provides no means for easily browsing or installing new plugins. Because the functionality provided often is standard on the other players (such as finding cover art), the impression is either that few people are writing plugins for Rhythmbox or that the project conceives plugins as a way of providing a smaller standard code base. Banshee similarly has a limited number of plugins, including some that could be standard features, although it does distinguish itself with extensions for creating bookmarks and detecting BPM (beats per minute).

By contrast, the other three players have a much richer ecosystem of plugins. Songbird distinguishes itself by add-ons for creating a mash tape or for learning about upcoming local concerts. Amarok's plugins are focused largely on providing alternative sources for lyrics and for automated logon to Internet sources, many of them (such as those for CBC radio or on-line readings of the Koran) obviously intended for specific local audiences. Only a few of Amarok's scripts, such as the script for copying the current song information into the clipboard, are intended to extend functionality, perhaps in part because the Amarok 2.x series of releases is relatively new, and at least until recently, developers were busy providing

Exaile

7th International
Penguin Conference

August 30 - September 3, 2010
Boston, MA

Thirteen of the world's eighteen penguin species are listed on the International Union for Nature Conservation's Red List of Threatened Species.

The International Penguin Conference brings together the world's leading scientists, research managers and policy makers working on penguin biology, ecology, health and behavior who are at the forefront of ongoing penguin research, working to identify current and emerging conservation issues and create action plans.

Sponsorship opportunities now available.

By sponsoring the IPC you are not only doing your part to help protect penguins, but also gaining exposure among world conservation leaders.

www.PenguinConference.org

basic functionality that users missed from the 1.x series of releases.

The most varied set of plugins for functionality is found in Exaile. For example, you can add plugins to display cover art with an album or on the desktop, set an alarm, enable support for Metacity hot keys in GNOME, reduce the window to a minimal size and many other possibilities. Among this variety, Exaile's lack of a control for adding new plugins seems a strange oversight.

RANKING:

1. **Amarok and Exaile** (tie—both are varied but have different priorities)
2. Songbird
3. Banshee
4. Rhythmbox

Additional Features and Customization

All five players have features that do not fit neatly into other categories. Rhythmbox's are modest: a minimal window and a full-screen mode, and visualization—a random pattern that displays while music is playing. Banshee's are almost as modest, with options for how files and the local music collection are handled.

With Exaile, the other features start to get more interesting. Exaile includes a blacklist manager, so you can exclude tracks that you prefer not to hear but are not ready to delete, as well as commands within the context menu to burn selections to CD/DVD.

In comparison, Amarok's extras are largely practical. They include the ability to

reposition notifications (which can come in handy if they block your set of icons) and the ability to edit database settings. An especially useful feature is the storage of passwords to sites in KDE Wallet to keep them secure while making them easily accessible. However, if you do not use on-line sites often, you may find logging in to KDE Wallet each time you start Amarok irksome.

In keeping with its Web origins, Songbird has a strong emphasis on privacy and security. It has Firefox-like controls for ensuring privacy, including a Clear Private Data feature. Also like Firefox, it can store passwords for sites and use a master password to save you the effort of remembering the other ones. Songbird preferences also include a page in which you can choose what information you want to share with sites that you log in to. Still another sign of its Web orientation is a tab in the preferences for setting up how you use iTunes.

RANKING:

1. **Songbird**
2. Amarok
3. Exaile
4. Banshee and Rhythmbox (tie)

Choosing a Music Player

I awarded one point for a first-place finish, two for a second-place finish and so on, which means a low score is better. Tallying the results, Amarok gets first place with 8 points, with Exaile and Songbird tied for second, with 15 points apiece. They were
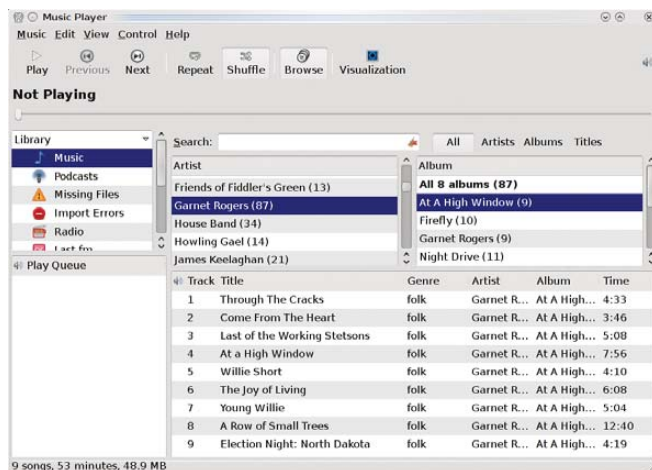

Songbird

Rhythmbox

followed by Banshee with 20 and Rhythmbox with 26.

These results don't tell the whole story. For instance, many uses will reject Banshee automatically because it depends on Mono. Others might prefer Banshee because it is GNOME-based or Rhythmbox because it includes a minimal feature set.

Just as important, I could have included at least five more players. When you consider Audacious, Listen, Quod Libet and Sonata, you realize just how broad the selection of free music players is. However, I selected these either because they are the default players with most distributions or because they have a substantial cult following. For better or worse, they are the most popular music players on the free desktop at the time of this writing.

However, even if you do not take the ranking as absolute, the reasons behind the rankings may help you decide which player is right for you. This exercise also suggests something about the current state of music players, with Amarok well in the lead and Rhythmbox fading, and the others sometimes sporting innovative features but failing to mount a general challenge to Amarok's dominance in the field.■

Bruce Byfield is a computer journalist who covers free and open–source software. He has been a contributing editor at *Maximum Linux* and Linux.com, and he currently is doing a column and a blog for *Linux Pro Magazine*. His articles appear regularly on such sites as Datamation, LinuxJournal.com and Linux Planet. His article, "11 Tips for Moving to OpenOffice.org" was the cover story for the March 2004 issue of *Linux Journal*.

Prisons vs. Horizons

How do closed content distribution devices pose threats to the Net's future? DOC SEARLS

The most important strategic moves in the tech world are more often orthogonal than oppositional. Put another way, capture is vertical while freedom is horizontal. If you want to escape from prison, you have to go sideways. And the outside is infinitely wider than the inside—higher too, but first you have to get out. If open is your strategy, ya gotta move laterally—to the outside, toward the horizons.

Last month, I wrote about how Apple's iPhone, the most cushy prison ever created for both developers and users, is challenged less by competing prisons than by open smartphones based on Google's Android. To review, the iPhone is a silo that stands on one company's closed OS and hardware. It is equipped with a slick SDK, rules galore about how products should run and developers behave, and a single retail sphincter—the iTunes "store"—through which all products, even ones that cost the customer nothing, are sold. Meanwhile, Android phones are restricted only in the sense that they have one (Linux-based) operating system, which is open to improvement and adaptation by anybody. There's no limit to its horizons.

So, while Apple has raised the bar for what smartphones can do, Google is now widening it. (So, for that matter, is Symbian, another open smartphone operating system that has more users than Apple's and Google's phones put together. Disclosure: I consult the Symbian Foundation.)

And now, with Apple's new iPad, the open OS folks have another category to widen. Fortunately, Apple has made it easier this time. That's because the iPad is a much narrower silo than the iPhone.

While smartphones are extensions of one's self (they are, primarily, phones), the iPad is mostly a consumption device. It is built to "deliver content" and to get money for it. All the other stuff the iPad does—e-mail, browsing, looking at picture albums and home videos—is gravy. The meat of the iPad is its system for pumping out content and getting money for it. Among the goods for sale through iPads are TV shows, movies, newspapers, magazines and books. In other words, the iPad is a better Kindle, with video and audio as well as print. Apple is also much better at playing this whole game—closed distribution through closed gear—than Amazon. Already (at the time this writing, in early February) Apple is winning "the e-book battle" by giving publishers what Amazon wouldn't give them: the ability to charge higher prices.

But content trapped in prisons is not the whole world, or even a majority of it. There's a limit to how big you can make a prison, and to the appeal of any prison to potential occupants. All these prisons still stand on the Net, which was built as a place where anybody can make and share (or sell, or both) anything. What we need now in that wide-open space are tablets that are more than real nice ways to "consume content".

We're sure to get them. Google is reportedly already working on one (or more) Android-based tablets. Maemo-based ones from Nokia have been coming out for half a decade and are bound to get better. I would love it if the Dells, Acers and HPs of the world would come through with open tablets based on open Linux OSes—and market them aggressively. I suppose one or more of them will, eventually. Meanwhile, we'll get what we want anyway. It's a big world, with lots of hardware makers.

What I'm worried about isn't the silos—or silos alone. The biggest dangers show up one layer down, where the Net's wide-open spaces are being carved up and fenced off while our leading blabbermouths are distracted, as usual, by vendor sports and other narrow concerns. Some of the carving is between silos. And while it's worth worrying about how much "content" gets locked up and how, there are problems just as big, if not bigger, at the national level. Writes Stephen Lewis, "By resting on a 'borrowed' infrastructure, the Internet has inherited the 'gatekeepers' that own and control, charge for, and regulate these legacy elements....Such organizations still carve up the world according to geopolitical entities and borders defined between the late-eighteenth century and the mid-twentieth and gerrymander services and access accordingly."

Nowhere is this a bigger deal than in China, where the Internet is replaced by a highly censored "Cinternet". The Open Net Initiative reports that more than 40 countries filter the Net, affecting more than a half-billion users.

Where the two threats—corporate silos and restrictive natures of nations—come together is around copyright. Here the frictions meet and lock. The Anti-Counterfeiting Trade Agreement (ACTA), currently being negotiated between the US, the European Union and other national entities, would, in the words of Aaron Shaw, "include sweeping provisions to criminalize information use practices currently allowed under US, European, and international law." Thus, to protect influential industries from "piracy" on the Net's high seas, ACTA would drain the oceans, replacing them with well-guarded canals.

All new regulations have the effect of protecting yesterday from last week. The irony here is that the Internet—even in its currently restricted places—is the best platform for tomorrow that humans have ever invented. If ACTA passes, it will find eager enforcers in the private prisons already being built.■

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.

Orion II iX-N4236

Powerful *4U* Orion II Storage Series

✔ **Outstanding Performance**
✔ **Excellent Cooling Efficiency**
✔ **Up to 24 Processing Cores with Hyper-Threading**
✔ **Up to 72TB in 4U, Unparalleled Storage Density**
✔ **Up to 432TB in 20U of Rack Space Utilizing Optional Orion II JBOD Expansion Units**

To order today call: **1-800-820-BSDi**

Orion II

Optional 45 Drive
JBOD Expansion Unit

Notable features include:

- Dual Intel® 64-Bit Socket 1366 Six-Core, Quad-Core, or Dual-Core, Intel® Xeon® Processor 5600/5500 Series
- 4U Storage Server Chassis with up to 72 TB storage capacity
- 36 x 3.5 Hot-Swap SAS/SATA HDDs (24 front side + 12 rear side)
- 1400 W (1+1) Redundant High Efficiency Power Supply (Gold level 93%+ power efficiency)

- Dual Intel® 5520 chipsets with QuickPath Interconnect (QPI) up to 6.4 GT/s
- Up to 192GB DDR3 1333/1066/800 MHz ECC Registered DIMM/24 GB Unbuffered DIMM
- 2 (x16) PCI-E 2.0, 4 (x8) PCI-E 2.0 (1 in x16 slot), 1 (x4) PCI-E (in x8 slot)
- Intel® 82576 Dual-port Gigabit Ethernet Controller

iXsystems Introduces the Orion II 4U Storage Solution

The iX-N4236 boasts energy efficient technology and maximum, high density storage capacity, creating a 4U powerhouse with superior cooling.

The Orion II has **thirty-six hot-swappable SAS/SATA drive bays**, providing 50% more storage density than its predecessor. By delivering high-end storage density within a single machine, iXsystems cuts operating costs and reduces energy requirements.

Storage sizes for the iX-N4236 are customizable, with 250GB, 500GB, 750GB, 1TB, and 2TB hard drives available. For environments requiring maximum storage capacity and efficiency, 2TB Enterprise-class drives are available from Western Digital®, Seagate®, and Hitachi. These drives feature technologies to prevent vibration damage and increase power savings, making them an excellent choice for storage-heavy deployment schemes.

The Intel® Xeon® Processor 5600 Series (Six/Quad-Core) and Intel® Xeon® Processor 5500 Series (Quad/Dual-Core) have a light energy footprint, while creating a perfect environment for intense virtualization, video streaming, and management of storage-hungry applications. Energy efficient DDR3 RAM complements the other power saving components while still providing 18 slots and up to 192GB of memory overall.

100% cooling redundancy, efficient airflow, and intelligent chassis design ensure that even under the heaviest of workloads, the Orion II remains at an optimal temperature, while still drawing less power than other servers in its class. With a 1400 W Gold Level (93%+ efficient) power supply, the entire system works together to efficiently manage power draw and heat loss.

For more information or to request a quote, visit:
http://www.iXsystems.com/Orion2

iX systems™

intel
Xeon
inside™

Powerful.
Intelligent.

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

More TFLOPS, Fewer WATTS

Microway delivers the fastest and greenest floating point throughput in history

2.5 TFLOPS

Enhanced GPU Computing with Tesla Fermi

▸ 480 Core NVIDIA® Tesla™ Fermi GPUs deliver 1.2 TFLOP single precision & 600 GFLOP double precision performance!

▸ New Tesla C2050 adds 3GB ECC protected memory

▸ New Tesla C2070 adds 6GB ECC protected memory

▸ Tesla Pre-Configured Clusters with S2070 4 GPU servers

▸ WhisperStation - PSC with up to 4 Fermi GPUs

▸ OctoPuter™ with up to 8 Fermi GPUs and 144GB memory

New Processors

▸ 12 Core AMD Opterons with quad channel DDR3 memory

▸ 8 Core Intel Xeons with quad channel DDR3 memory

▸ Superior bandwidth with faster, wider CPU memory busses

▸ Increased efficiency for memory-bound floating point algorithms

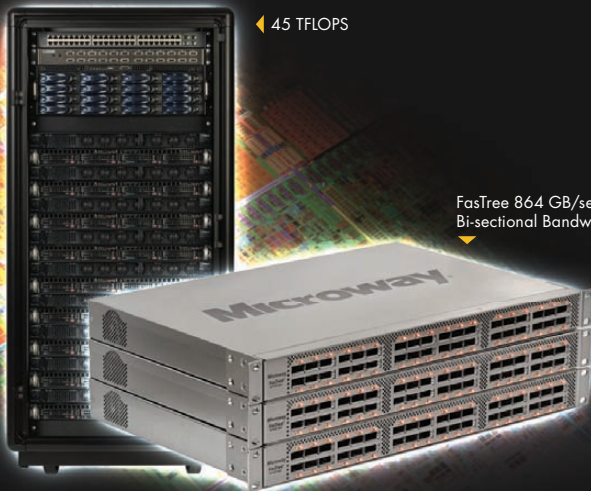### Configure your next Cluster today!

www.microway.com/quickquote
508-746-7341

10 TFLOPS   5 TFLOPS

45 TFLOPS

FasTree 864 GB/sec
Bi-sectional Bandwidth

FasTree™ QDR InfiniBand Switches and HCAs

▸ 36 Port, 40 Gb/s, Low Cost Fabrics

▸ Compact, Scalable, Modular Architecture

▸ Ideal for Building Expandable Clusters and Fabrics

▸ MPI Link-Checker™ and InfiniScope™ Network Diagnostics

Achieve the Optimal Fabric Design for your Specific MPI Application with ProSim™ Fabric Simulator

Now you can observe the real time communication coherency of your algorithms. Use this information to evaluate whether your codes have the potential to suffer from congestion. Feeding observed data into our IB fabric queuing-theory simulator lets you examine latency and bi-sectional bandwidth tradeoffs in fabric topologies.

NVIDIA TESLA CUDA

intel Channel Partner Premier 2010

AMD Fusion Partner Premier

GSA

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

Microway

Technology you can count on sm