LINUX™

JOURNAL

Since 1994: The Original Magazine of the Linux Community

AUGUST 2008 | ISSUE 172

REVIEWED:
Neuros OSD and
Cradlepoint PHS300

WE'VE GOT
BUGs

AND OTHER
COOL
PROJECTS
TOO

E-Ink +
Gumstix
Perfect
Match?

How To:
16 Terabytes in One Case

Billix
Kiss Install
CDs Goodbye

✚ Learn to Fake a
UFO Landing Video

Wiimote Linux
Interface **HOW-TO**

```
% ar x "a trouble free network"
ar: a trouble free network does not exist


% "How is Avocent geared to helping me create one?
Unmatched."


% ^What if we decide to do nothing?
Bad substitute


_
```

Even your ***nix** wants Avocent.

ACS 6000 Series Console Servers

At Avocent, we realize that as Unix and Linux® users, you like to do things your own way. You like to use CLI more than a GUI, you like to use Firefox® more than "the other one." Heck, some of you even like to set up a few servers at home to develop your own applications.

That's why our software solution allows you to use your CLI of choice, access your Unix or Linux® boxes from anywhere and run whatever browser you want. We wanted it to be flexible enough for you to do things **your way.**

• Simplify management with auto-discovery of target devices

• Optimize availability with multiple access paths

• Increase security through access levels

• Reduce set up time with software-controlled Cisco® pin-outs

Avocent®   To learn more, go to **www.avocent.com/acs**

CONTENTS AUGUST 2008
Issue 172

FEATURES

CONTENTS
AUGUST 2008
Issue 172

Next Month

THE UNDERDOG ISSUE

Everybody loves an underdog.
Some people even consider
Linux to be an underdog (we
prefer "Undiscovered
Champion"). Next month, we
focus on the little guy. Sure you
know all about Ubuntu, but
what about Gentoo? Granted,
Apache is the king of the Web
server world, but what about
the alternatives? (No, not IIS.)
Heck, even Inkscape has some
less-popular alternatives like
Xara Xtreme. So, whether
you're looking for a command-
line e-mail client, or you want
an alternative to BIND, next
month will be an issue you
don't want to miss.

EmperorLinux

...where Linux & laptops converge

Powerful Linux: The Rhino

- Based on the Dell Precision M6300/Latitude D830
- Highest performance NVidia 3-D graphics available on a laptop with WUXGA widescreen
- High performance Core 2 Extreme CPU plus 8 GB RAM
- Ultimate configurability — choose your laptop's features

Features include:

- 1.8-2.8 GHz Core 2 Duo/Extreme
- Up to 17" WUXGA LCD w/ X@1920x1200
- NVidia Quadro FX 3600M graphics
- 60-200 GB hard drive (7200 rpm SATA), up to 8 GB RAM
- DVD±RW or Blu-ray, Ethernet, 802.11a/b/g, Bluetooth
- One year Linux tech support - phone and email
- Three year manufacturer's on-site warranty
- Choice of pre-installed Linux distribution:

Rugged Linux

Tarantula CF-30

- Panasonic Toughbook CF-30
- Fully rugged MIL-SPEC-810F tested: drops, dust, moisture, & more
- 13.3" XGA TouchScreen
- 1.6 GHz Core 2 Duo
- Up to 4 GB RAM
- 80-200 GB hard drive
- Call for quote

Tablet Linux

Raven X61 Tablet

- ThinkPad X61 tablet by Lenovo
- 12.1" SXGA+ w/ X@1400x1050
- 1.6 GHz Core 2 Duo
- Up to 4 GB RAM
- 80-200 GB hard drive
- Pen/stylus input to screen
- Dynamic screen rotation
- Starts at $2150

Portable Linux

Toucan T61/T61p

- ThinkPad T61/T61p by Lenovo
- Up to 15.4" WUXGA w/ X@1920x1200
- NVidia Quadro FX 570M graphics
- 1.8-2.6 GHz Core 2 Duo
- Up to 4 GB RAM
- 80-250 GB hard drive
- 5.2-6.0 pounds
- Starts at $1530

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

LINUX JOURNAL

Since 1994: The Original Magazine of the Linux Community

Digital Edition Now Available!

Read it first
Get the latest issue before it hits the newsstand

Keyword searchable
Find a topic or name in seconds

Paperless archives
Download to your computer for convenient offline reading
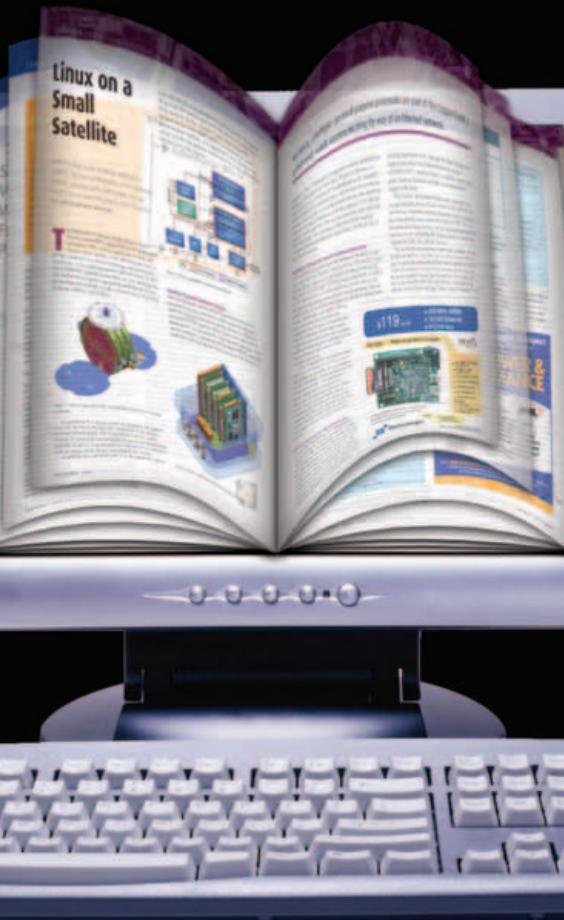
Same great magazine
Read each issue in high-quality PDF

Try a Sample Issue!
www.linuxjournal.com/digital

Linux: the Root of All Coolness

SHAWN POWERS

If you are a regular visitor to the LinuxJournal.com Web site, you might recognize me as the goofy video Gadget Guy, or possibly as the Web author with a penchant for controversy. While the latter is largely coincidental, the former is just the way I am (my wife can grudgingly attest to that). This month marks the first issue that I'm the Associate Editor of the print magazine as well. Whether adding me to the staff will be beneficial, or more like the spreading of Windows spyware, is yet to be determined.

The Cool Projects issue is significant to me for another reason as well. A year ago, in the August 2007 issue, my "How to Build Your Own Arcade" article marked the first time I was published in *Linux Journal* (**www.linuxjournal.com/article/9732**). It also appeals to my inner child that thinks life should revolve around stuff that's "cool". The 2008 Cool Projects issue (the one you're reading now) offers plenty of opportunity to have fun with our favorite operating system. Whether you're looking for a cool way to do your job, or whether you're trying to avoid doing your job altogether, we've got you covered.

If you subscribe to *Linux Journal* at work, and you're trying to justify the Cool Projects issue to your boss, fear not. We make it much easier than trying to explain the *Sports Illustrated* Swimsuit Edition to your significant other. Eric Pearce shows us how to make a 16-Terabyte file server out of bubble gum and popsicle sticks. Well, okay, maybe not with those ingredients, but he walks us through the process of creating a really big server.

Bill Childers shows us one of the coolest uses of a USB Flash drive I've ever seen. With an outdated 256MB drive, Bill shows us how to make a bootable device that will install many different Linux distributions and launch a handful of utilities too!

If you can't find something this month that directly ties to your job, feel free to play the "professional development" card, and have some fun while you're furthering your technological horizons. Michael Surran, for example, tells us all about his use of Linux in education.

As someone who has implemented Linux in schools before, I always find it cool when schools take the plunge.

Perhaps you are a programmer, and code all day, and code all night. Reuven M. Lerner shows us how to make sure our Rails are optimized, and Dave Taylor helps us extract really important data—from the Internet Movie Database. Along with some open-source mapping software, this issue is really full of easily justifiable diversions.

For me, however, the exciting thing about the Cool Projects issue is building cool stuff. Have you seen the Bug Labs' BUGs? All you have to do to build a cool project with them is snap together the pieces you want. The BUGs are amazingly versatile and are being developed every day. We show you how to make the little buggers bend to your will. Or, maybe you want to learn to use E-Ink technology and handcraft your own tiny PC. Jaya Kumar shows us how.

What if you don't subscribe to *Linux Journal* at work, and you're just looking for some cool things to do with Linux in your spare time? Kyle Rankin and Marcel Gagné felt the same way. Kyle shows us how to interface a Wii remote (Wiimote) to our Linux machines and use the controller as a joystick and mouse. Marcel, taking the word "cool" literally, shows us a handful of penguin and ice games bound to keep you busy for hours.

Finally, if reality isn't cool enough, we've got Zenoss, and we've got "How to Fake a UFO Landing". Granted, the two have nothing to do with each other, but if you name your network-monitoring system Zenoss (Zeen-ohss), you're just asking for some taunting. So, sit back, prop up your feet, and enjoy this issue of *Linux Journal*. If you get tired of reading, maybe catch a few flicks on TV with your Neuros OSD. We'll tell you about that little beauty as well.■

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.

servers ▶ DIRECT®

MORE PRODUCTS, BETTER SERVICE, GUARANTEED.

GET THE SERVER THAT TAKES YOUR BUSINESS AS SERIOUSLY AS YOU DO.

Intel® Xeon® Processor powering the ServersDirect® Systems provide the quality and dependability to keep up with your growing business.

GO STRAIGHT TO THE SOURCE!   1.877.727.7887 | www.ServersDirect.com

---

SDR-S1208-T00 STARTING AT $599

- Supermicro Mini 1U Rackmount Server with 260W Power Supply
- Supermicro Server Board w/Intel® 946GZ Chipset
- Support up to a Dual-Core Intel® Xeon® 3000 Series processor
- TPM Support
- 1x 3.5" Internal Drive Bay
- 2x Intel® 82573 PCI-e Gigabit LAN Port

SDR-S1312-S04 STARTING AT $1,499

- Supermicro 1U Rackmount w/650W Redundant High-efficiency Power Supply
- Supermicro Server Board w/Intel® 5000P Chipset
- Dual Intel® 64-bit Xeon® Quad-Core or Dual-Core
- Support up to 32GB DDR2 667 & 533 FB-DIMM
- 4 x 3.5" Hot-swap SATA Drive Bays
- Dual-port Gigabit Ethernet Controller

SDR-S1327-T04 STARTING AT $1,789

- Supermicro 1U Twin (2 System Nodes in 1U) with 780W Power Supply
- Each node supports the following:
  - Dual Intel® Xeon® Processor 5400 Support
  - 2x Intel® 82598EB 10 Gigabit Ethernet
  - Up to 48GB DDR2 667/533
  - 2 SATA Hot-swap Drive Support

SDR-C2205-T00 STARTING AT $699

- 2U Rackmount Server with 300Watt Power Supply
- Supermicro Server Board w/Intel® 3200 + ICH9R Chipset
- Support up to a Quad-Core Intel® Xeon® 3300/3200 series processor
- Up to 8GB unbuffered ECC / non-ECC DDR2 800/667 SDRAM
- 4x 3.5" Internal Drive Bay
- Dual-port Gigabit Ethernet Controller

SDR-S2301-T08 STARTING AT $1,599

- 2U Rackmount Server w/700W High-Efficiency Redundant Power Supply
- Supermicro Server Board w/Intel® 5000P chipset
- Dual Intel® 64-bit Xeon® Quad-Core or Dual-Core
- Support up to 64GB DDR2 667 & 533 FB-DIMM
- 8 x 3.5" Hot-swap Drives Trays
- Dual-port Gigabit Ethernet Controller

SDR-S3300-U16 STARTING AT $2,079

- 3U Rackmount Server with 800W Redundant Power Supply
- Supermicro Server Board w/Intel® 5000P chipset
- Dual Intel® 64-bit Xeon® Quad-Core or Dual-Core
- Support up to 64GB DDR2 667 & 533 FB-DIMM
- Dual-Channel U320 SCSI with 16 (8+8) Hot-swap Drive
- Dual-port Gigabit Ethernet Controller

SDP-CP200-T04 STARTING AT $899

- Pedestal Server with 600W Power Supply
- Intel® Server board with 3000 Server Chipset
- Support up to a Quad-Core Intel® Xeon® 3300/3200 series processor
- Up to 8GB unbuffered ECC / non-ECC DDR2 800/667 SDRAM
- 4x 3.5" Internal Drive Bay (Hot-swap Optional)
- Dual-port Gigabit Ethernet Controller

SDR-S4302-U08 STARTING AT $1,799

- 4U Rackmountable / Tower w/ 800W High Efficiency Red. Power Supply
- Supermicro Server board with Intel® 5000P Chipset
- Dual Intel® 64-bit Xeon® Quad-Core or Dual-Core
- Support up to 64GB DDR2 667 & 533 FB-DIMM
- Dual-Channel U320 SCSI with 8 (4+4) Hot-swap Drive
- Dual-port Gigabit Ethernet Controller

SDP-IP306-U10 STARTING AT $2,259
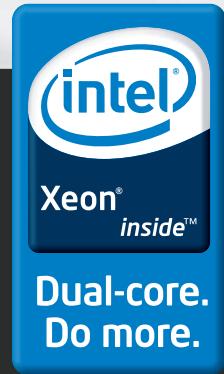
- Intel Pedestal Chassis with 830W Power Supply
- Intel® Server board with 5000 Server Chipset
- Dual Intel® 64-bit Xeon® Quad-Core or Dual-Core
- Support up to 64GB DDR2 667 & 533 FB-DIMM
- 10x Hot-swap SCSI Bays
- Dual-port Gigabit Ethernet Controller

SERVERS DIRECT CAN HELP YOU CONFIGURE YOUR NEXT HIGH PERFORMANCE SERVER SYSTEM - CALL US TODAY!

Our flexible on-line products configurator allows you to source a custom solution, or call and our product experts are standing by to help you assemble systems that require a little extra. Servers Direct - your direct source for scalable, cost effective server solutions.

1.877.727.7887 | www.ServersDirect.com

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, Pentium, and Pentium III Xeon are trademarks of Intel Corporation or it's subsidiaries in the United States and other countries.

(intel)
Xeon® inside™
Dual-core. Do more.

For powerful choose a web host!

✓ **Speedy Connection**
1&1 servers benefit from 100 MBit connection

✓ **Great Prices**
All dedicated servers are on sale!*

✓ **Cutting-edge Technology**
Hardware to meet your strong demands

AMD Athlon™ 64

AMD Opteron™ 64

HDD 0   HDD 1
IP: 212.227.126.71

World's #1 Web Host

With a wide variety of products and hosting packages, superior data center technology, unparalleled reliability, special offers, great prices and a 90-day-money-back guarantee, it's no wonder 1&1 is the world's biggest and fastest growing web host!

servers, powerful

Best Value: Compare for yourself.	1&1	The Planet
	BUSINESS I	**Intel Pentium 4 Series 2.4+**
CPU	AMD Athlon 64 3500+	Intel Pentium 4 2.4 GHz+
RAM	1 GB	512 MB included, add $15/month for 1 GB
Useable Disk Space	160 GB	80 GB
RAID	RAID 1 Included (2 X 160 GB HD)	Add $10/month for second 80 GB HD for RAID 1
BackUp	Full 160 GB Backup Included	Add $80/month for 80 GB Backup
Monthly Traffic (GB/month)	2000 GB/month	750 GB included, add $175/month for 1750 GB
Total Monthly Fee as Configured	~~$99~~ $49⁵⁰*	$389
	ENTERPRISE I	**Conroe 3060 Series - SATA**
CPU	Dual Core AMD Opteron 1218-2.6 GHz	Dual Core Intel Xeon 3060 Conroe Processor-2.4 GHz
RAM	4 GB	2 GB included, add $50/month for 4 GB
Useable Disk Space	400 GB	250 GB
RAID	RAID 1 Included (2 X 400 GB HD)	Add $40/month for RAID 1 + $20/month for 2nd 250 GB HD
BackUp	Full 400 GB Backup Included	Add $200/month for 200 GB Backup
Monthly Traffic (GB/month)	4000 GB/month	2500 GB included, add $175/month for 3500 GB
Total Monthly Fee as Configured	~~$299~~ $149⁵⁰*	$694
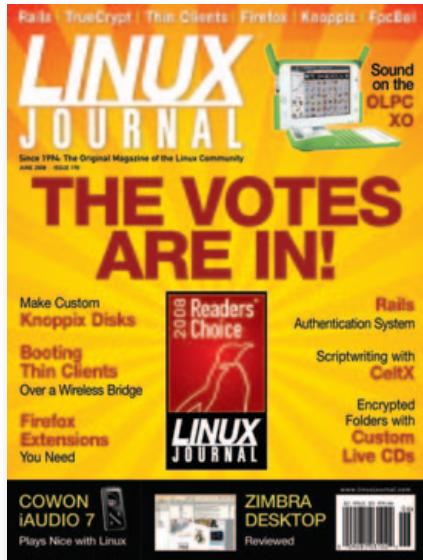
.US Domain only $2.99 first year!

For a limited time, America's internet address is on sale.
Create your own .us domain by visiting www.1and1.com.

Call **1.877.go1and1**

Visit us now **1and1.com**

1&1

letters

Try Puppy Linux

Wow, what a great article by Louis Iacona [*LJ*, April 2008, "Puppy Linux"]. I was pleasantly surprised to find it so in depth for a magazine article, which is usually no more than two pages. It definitely encouraged me to try Puppy Linux, which I will do. I hope to see more articles by this gentleman and hope he was well paid. Thank you for such a great magazine.

--
George Mulak

On ISOs

I have noticed mention of ISOs now and then in *LJ* articles.

While working on our own applications, we've often gone searching on the Internet for tools and applications that might already be found in the form of a ready-to-run ISO.

Because ISOs are relatively new to the public, we concluded that currently, it is difficult to find such works or list them if you are an author.

Therefore, we have created a new site, **www.isotogo.com**, to help the public and authors in working with ISOs. There is no cost to use it or to list your works, and because it is new, we are

working hard on moving up the search engine ladder.

--
Mike Paradis

Comments on the OLPC XO

Dave Phillips is effusive in his praise of the OLPC XO [*LJ*, June 2008], and most of it is well deserved, indeed. But it must be conceded that the keyboard is a piece of absolute trash. After just a few hours of use, mine developed keys that stick or fail to actuate or actuate with Alt applied unpredictably. This got worse until it was completely unusable. A Google search turned up many such complaints and detailed instructions for excising and replacing the keyboard (serious tinkerers only!). The USB keyboard is really a necessity, but that is not convenient in all circumstances.

The mouse does have two active buttons, as you can verify by copying the binary of gpm from a compatible system (I use Fedora Core 6 on a Dell Latitude). Run it as:

```
gpm -m /dev/input/mouse0 -t ps2 -r 5 -a 3
```

and play with the -r and -a settings to get it the way you like.

The display is a little bit strange. Plotting a bunch of random pixels in white on a black screen makes red, green and blue dots. A white-on-black line may show bands of color, depending on the point density and inclination. So your favorite graphics apps may need some tweaking.

If you want to run something that uses SVGALIB, you need my framebuffer version of that. It's not complete yet, but it does basic pixel, line and block functions. I'll put it on my SourceForge site soon, but meanwhile, if interested, send me a note at mcconnau@biochem.wustl.edu.

--
Bill McConnaughey

Yet Another One-Liner

The June 2008 issue of *LJ* published a letter from David Newall, which responded to a letter from Joao

Macedo published in the February 2008 issue, which was, in turn, a response to Dave Taylor's column in the December 2007 issue. Both Joao and David wrote in with one-liners using the echo and bc commands to do floating-point calculations in place of using Dave Taylor's solve.sh script. Joao's example embedded an actual newline character, whereas David Newall's used the escape code version of the same. There is yet another way to do this, and it is, in fact, my preference, as it is much more intuitive to folks accustomed to writing shell scripts. In bourne-shell scripts, a semicolon can be used to place commands together on a single line. It can be used for the same purpose with bc. Here is a third rewrite of the example to demonstrate:

```
echo 'scale=4;11/7' | bc
```

--
James Williams Zavada

Correction

First off, great magazine! While reading page 13, in the UpFront section [*LJ*, June 2008, "Eee PC Gets an Upgrade"], Doc talks about the upgrade that the Eee PC is getting, along the lines of the larger screen, larger SSD hard drive and more memory. He lists the new Eee PC 900 as having 1GB of RAM, and say that this is "up from 512KB". I dunno about you, but my Eee PC (from which I am sending this letter) has 512 *megabytes, not kilobytes*! If yours has 512KB of RAM, you should send it back! Great magazine, small typo, I forgive you!

--
Eric Jennings

Thanks

I just wanted to send a quick note to thank all of the contributors to *LJ*. You have inspired me over the past couple of years to migrate over to Linux as my OS of choice and motivated me to learn new projects. I am in the middle of setting up an LTSP project for our home-schooling community and using info gained from various *LJ* articles and book recommendations.

Continuous Data Protection
The Future of Data Centers

Can your backup software do this?	R1Soft CDP Server	Acronis® True Image	EMC Retrospect ®
Daily Backups	✔	✔	✔
Hourly Backups	✔	Not Supported	Not Supported
Open File Backups	✔	✔	Not Supported
Bare-Metal Restore	✔	✔	Not Supported
Continuous Data Protection	✔	Not Supported	Not Supported
Restore Linux LVM	✔	Not Supported	Not Supported
Restore Linux Software RAID	✔	Not Supported	Not Supported
Easy To Use Web Interface	✔	Not Supported	Not Supported
Manage Thousands of Servers	✔	Not Supported	Not Supported
Control Panel Integration	✔	Not Supported	Not Supported
	$99 /server †	$699 /server	You Can't Afford It

(NEW) - CDP for MySQL add-on Now available │ True-Granular Restore™ │ Store over 50 recovery points per-day
Bare-Metal Restore for MySQL Servers │ Restore tables or databases to original or alternate locations and more..

Data Centers serious about uptime and performance use R1Soft.

For more information visit: **www.r1soft.com** or call us at **800-956-6198**

†Price includes $600 Data Protection Server cost. Assumes minimum ratio of 25 protected servers per Data Protection Server

R1Soft
Continuous Data Protection®
For LINUX & WINDOWS

[LETTERS]

Keep up the good work. Hopefully I will send some converts your way soon!

--
Dean Anderson

Sony

I'm happy to see the coverage of Sony's use of Linux in the June 2008 issue. There is actually an even bigger list of Sony products running Linux at **www.sony.net/Products/Linux**. Myself, I was surprised to find my new digital camera on that list. Now if only we could turn this into some sort of quality label instead of a hidden feature. (Disclaimer: I am a UNIX sysadmin working for Sony.)

--
Nico De Ranter

More Must-Have Firefox Extensions

I'm surprised the article [Dan Sawyer's "Must-Have Firefox Extensions" in the June 2008 issue of *LJ*] didn't even mention AdBlock Plus. It's the first extension I put on any Firefox installation I come across. After I installed it on my girlfriend's laptop, she exclaimed, "now I understand why you actually *like* that one site!" Her laptop's running Kubuntu with VMware Workstation for those pesky Windows-only apps, by the way.

Another helpful extension is Cookie Button, which prevents all those cookie confirmation windows from popping up and still allows one to enable them easily for a specific site if required.

I also enjoyed last month's article regarding using virtualization on Mac OS X, because that's what I've been doing ever since I got my MacBook Pro [Dave Taylor's "Running Ubuntu as a Virtual OS in Mac OS X" in the May 2008 issue of *LJ*]. It's running Parallels with Kubuntu and Windows XP, and it allows me to develop and test software on all three operating systems with ease.

And, I can run Amarok to listen to music, because iTunes simply doesn't measure up.

--
U. Hertlein

GENSTOR

"Company With A Vision"

GNS 2450 *Green* Storage Appliance

Green
Scalable
Reliable
Performance
Plug & Play

Genstor Systems GNS 2450 is built on fully optimized Highly Efficient dual-core AMD Opteron™ processor model 2214 HE running Open-E® DSS™ (Data Storage Server).

GNS 2450 is an all-in-one IP-Storage with NAS and iSCSI (both target and initiator) functionality. Excellent price performance value, enhanced management and superior reliability is great for organizations of all sizes.

▶ Hardware & Software

- One or two Highly Efficient dual-core AMD Opteron™ processor-based platform

- 8GB RAM (Expandable to **64GB**)

- Dual GbE NICs (Optional 10GbE)

- 2U rackmount chasis with 12 SATA HDD bays with redundant power supplies

- Open-E® DSS™ (Data Storage Server) with NAS and iSCSI functionality

▶ Features & Benefits

- Administration:
  Powerful, secure and easy intuitive GUI for configuration and remote management

- Network Management:
  Along with standard Network Gateway support, the **GNS 2450** also supports DHCP Client, Multiple NIC, Adapter Fault Tolerance (AFT), Adaptive Load Balancing (ALB), Proxy Settings, IP-Sec, etc.

- Flexibility & Scalability:
  The **GNS 2450** storage appliance is highly scalable and flexible to take care of your future expandability needs

Price: $5200 (for 6TB raw storage)

GENSTOR SYSTEMS, INC
780 Montague Expressway # 604
San Jose, CA 95131
Phone: 408-383-0120
Fax: 408-383-0121

AMD Opteron 64

open-e

Please contact Genstor sales@genstor.com for your custom hardware needs
1-877-25-SERVER • www.genstor.com • sales@genstor.com
Open-E, Open-E DSS and the Open-E logo are registered tradedmarks of Open-E, Inc.
AMD, AMD Arrow logo, and AMD Opteron are trademarks or registered trademarks of Advanced Micro Devices, Inc.

diff -u

WHAT'S NEW IN KERNEL DEVELOPMENT

Having to reboot production systems to incorporate **security patches** is a pain. How much better would it be simply to graft the patch onto an already running kernel, and let it keep running? This is exactly what **Jeff Arnold** has been working on. He calls it **KSplice**, and at least for the moment, it supports any kernel that can load a module. The kernel itself doesn't have to support the feature explicitly. The way it works is that the user supplies the source tree for the running kernel and the patch to graft on. Then, KSplice compiles the patch and loads modules that apply the patch internally. Because of the interest it has generated, it's looking very likely that KSplice will be accepted into the kernel tree, at which point it might stop supporting kernels that don't know about it.

One obstacle standing in the way of Jeff's work is **Microsoft's legal department**. During the course of discussion, it came out that patent application 10/307,902 from 2002 seemed to cover Jeff's idea. And, although a number of folks, including **Gilles Espinasse** and **Willy Tarreau**, said they'd been "hot-patching" operating systems since the 1990s and earlier, **Bill Davidsen** felt that trying to launch a challenge against a Microsoft patent would be prohibitively expensive. However, according to Jeff, the patent application was rejected by the patent office. So, Microsoft may give up at this point, depending on its current internal weirdness level.

NFS is a lovely filesystem, but it has various problems that make people want to replace it. One of the latest attempts is **POHMELFS**, or Parallel Optimized Host Message Exchange Layered File System. POHMELFS is written by **Evgeniy Polyakov**, and it is a user-space layer that can be applied to any back-end filesystem, such as ReiserFS or XFS. It also seems to outperform NFS fairly significantly, at least according to the tests Evgeniy has performed so far. But, it's still only ready for playing around with. People shouldn't use it to store data they want to keep. Evgeniy

says it's too soon to talk about the filesystem being reliable for use. He's been able to do regular user activity on it without a problem, but he expects to find bugs, POSIX conformance issues and other issues.

ReiserFS has migrated its development from the **NameSys** servers to **kernel.org**, where work is continuing. **Edward Shishkin** and others continue to develop the filesystem in spite of **Hans Reiser**'s murder conviction.

There have been various other maintainer updates. **Greg Kroah-Hartman** is no longer the **PCI** maintainer; he's handed the whole kit and kaboodle off to **Jesse Barnes**. **Timur Tabi** has listed himself as the official maintainer of the **Cirrus Logic CS4270 sound driver**, the **Freescale QUICC engine library**, the **QUICC engine UCC UART driver** and the **Freescale SOC sound drivers**. And, **Zhang Wei** has abdicated maintainership of the **Freescale DMA driver** and handed that project over to **Li Yang**.

The politics of competing filesystems is never pretty. **LogFS** wants to support Flash drives, but its development has been slower than some people would like. So, **Artem Bityutskiy** and **Adrian Hunter** recently announced their own alternative, **UBIFS**, that is apparently quite a bit further along than LogFS. It's faster, more stable and more featureful, although it still has trouble with devices larger than 64G. LogFS, developed by **Jorn Engel**, also came out with a new release, perhaps partly in response to the UBIFS announcement. Of course, nothing says there can't be two co-existing Flash filesystems, but apparently one of the motivations for UBIFS was

the relatively slow pace of LogFS development. Artem in particular seemed bitter about this, especially since it appeared that various folks were suggesting waiting for LogFS instead of developing UBIFS, while UBIFS was already superior to LogFS.

Matthew Wilcox and others are trying to eliminate as many **semaphores** as they can from the kernel, and replace them with simpler locking structures, such as mutexes, spinlocks and completions. The problem is that semaphores provide additional features that are hard to mimic with those other types of locks. For example, semaphores can manage access to a cluster of resources, while other locks basically are just on/off switches. The benefit of eliminating semaphores is that it's possible to gain speed and, in the case of single-processor systems, to optimize the lock entirely out of the compiled binary, resulting in further speed gains. But to do this, key features of semaphores have to be mimicked near any lock that replaces one. Matthew, **Arjan van de Ven** and **Ingo Molnar** are addressing this by developing **kcounter**, which will provide ways of managing access to clusters of resources, similar to how semaphores do it.

Unfortunately, kcounter takes a cookie-based approach, similar to other things that have been seen in the kernel before, which have resulted in what **David Chinner** characterized as a very ugly interface. Hopefully, kcounter will avoid that pitfall, although it does seem as though a significant speedup might justify a little cookie ugliness. That question undoubtedly will spawn some lively debate.

—ZACK BROWN

LJ Index, August 2008

1. **Number of new toys the average child gets per year: 70**

2. **Size of the "baby industry", in trillions of dollars: 1.7**

3. **Number of computers donated to the World Computer Exchange: 26,695**

4. **Schools, orphanages and libraries served by the World Computer Exchange: 2,543**

5. **Youth connected per year by the World Computer Exchange: 1,079,110**

6. **Number of languages other than English among contributors to DistroWatch: 42**

7. **Number of countries covered by Linux-hosted Global Voices Online: 192**

8. **Number of authors for Global Voices: 101**

9. **Number of languages into which Global Voices is being translated: 11**

10. **Number of computers in the Windsor Unified School District (California): 2,500**

11. **Percentage of computers at Windsor Unified to be replaced by Linux thin clients: 50**

12. **Estimated thousands of dollars in energy saved annually by Windsor Unified School District, thanks to Linux servers and thin clients: 25**

13. **Estimated thousands of dollars in energy saved annually by Windsor Unified School District by switching to free software: 50**

14. **Estimated thousands of dollars in equipment costs saved annually by Windsor Unified School District by switching to Linux gear: 280**

15. **Number of dollars in four spent on energy in schools that is unnecessary: 1**

16. **Months children at Villa Cardal, Uruguay, had spent with beta XO OLPCs: 6**

17. **Average number of files created per XO by Villa Cardal kids: 1,200**

18. **Number of MB, +/–10, produced per machine by Villa Cardal kids: 40**

19. **Thousands of XOs due for Uruguay in 2008: 90**

20. **Thousands of XOs ordered by Peru in December 2007: 260**

Polynational Tux Curiosity

One could play for hours with Google Trends (**trends.google.com**). Not only does it show the spikes and slopes of search volume across time since the beginning of 2004, but it also lists the current top ten regions, cities and languages for each search. You can search for multiple keywords, comma-separated, and see colored lines for each. The results are usually more interesting than revealing. Such as:

■ Searches for Ronaldo and Beckham both spiked in 2005 at the last World Cup.

■ Searches for John Paul and Ratzinger peaked one after the other in early 2005 when the former died and the latter succeeded him as pope.

■ Searches for Linux and Microsoft have both gone down, the former slightly more than the latter.

■ Searches for Red Hat, SUSE and Ubuntu show declines for the first two and a steady rise for the third. Add Linux and find that Ubuntu has almost overtaken Linux in search volume. Does the rise in Ubuntu account for the decline in Linux searches? They seem somewhat reciprocal, but who knows?

What's more surprising are the top ten regions, cities and languages for each. Some results for Linux:

■ Russia is the top region, closely followed by India and the Czech Republic. The US is not among the top ten.

■ Beijing is the top city, followed by Tokyo and San Francisco, which is the only US city. The rest, in order, are Milan, Frankfurt, Augusta (Italy), Paris, Amsterdam, Madrid and London.

■ Russian is the top language, followed by German, Japanese, Italian, Chinese, Polish, Finnish, Portuguese, English and Swedish.

Some results for Ubuntu:

■ Norway is the top region, followed by Italy and Russia. The US is not on the list.

■ Milan is the top city, followed by San Francisco and Augusta (Italy). The rest are Helsinki, Madrid, Paris, Santiago (Chile), Frankfurt, Zurich and Mexico City.

■ Italian is the top language, followed by Finnish, Swedish, Russian, French, Spanish, German, Polish, English and Portuguese.

Google's qualification: "Google Trends aims to provide insights into broad search patterns. Several approximations are used when computing your results. Please keep this in mind when using it."

Also keep in mind that these were results on May 13, 2008. Try them when you read this to see how they change. Having used Google Trends for a while now, I can assure you the answer is: a lot.

—DOC SEARLS

On the Web, Articles Talk!

Every couple weeks over at LinuxJournal.com, our Gadget Guy Shawn Powers posts a video. They are fun, silly, quirky and sometimes even useful. So, whether he's reviewing a new product or showing how to use some Linux software, be sure to swing over to the Web site and check out the latest video: **www.linuxjournal.com/video**.

We'll see you there, or more precisely, vice versa!

One Tale of Two Scientific Distros

A few weeks ago, I was flying west past Chicago, watching the ground slide by below, when I spotted the signature figure eight of the Fermi National Accelerator Laboratory, better known as Fermilab. I shot some pictures, which I put up at the *Linux Journal* Flickr site (**www.flickr.com/**

groups/linuxjournal/pool, which runs on Linux too).

I figured Fermilab naturally would use Linux, and found that Fermilab has its own distro: Fermi Linux. Its public site provides a nice window into a highly professional and focused usage of Linux. Within Fermi Linux, specific generations are known as Scientific Linux Fermi, each with version numbers and the code names Charm, Strange, Top, Bottom, Up, Feynmann, Wilson and Lederman.

Some also have LTS in their names. LTS stands for Long Term Support. It has a FAQ. The first Q is, "What is Fermi Linux LTS?" The A goes:

Fermi Linux LTS (Long Term Support) is, in essence, Red Hat

Enterprise, recompiled.

What we have done is taken the source code from Red Hat Enterprise (in srpm form) and recompiled it. The resulting binaries (now in rpm form) are then ours to do with as we desire, as long as we follow the license from that original source code, which we are doing.

We are choosing to bundle all these binaries into a Linux distribution that is as close to Red Hat Enterprise as we can get it. The goal is to ensure that if a program runs and is certified on Red Hat Enterprise, then it will run on the corresponding Fermi Linux LTS release.

A follow-up Q goes, "I really don't

want to get into legal trouble, please convince me that this is legal." The A says:

> What we are doing is getting the source rpm of each Red Hat Enterprise package from a publicly available area. Each of these packages, except for a few, have the GPL license. This license states that we can freely distribute that package. We are recompiling those packages without any change. Hence, we can freely distribute those rpms that were built....And although these rpms are basically identical to Red Hat's Enterprise Linux, they were built by us and are freely distributable. We can do with them what we want....

> Although it is basically identical to Red Hat Enterprise Linux, it is, in essence, a completely

different release, just with the same programs, packaged the same way.

Fermilab supports its own users and directs others toward Scientific Linux, which was codeveloped by

Fermilab, CERN and other laboratories and universities. Troy Dawson is the primary contact for both Fermi Linux and Scientific Linux. On his own site, he explains, "Fermilab uses what is called Fermi Linux. It is now based on Scientific Linux. It is actually a site modification, so technically it is Scientific Linux Fermi. But we call all of the releases we have made Fermi Linux."

While Fermi Linux's version history

starts with 5.0x in 1998, Scientific Linux's history starts with 3.0.1 in 2004. Both sites' current distribution version pages have near-identical tables of releases, dates and notes. The latest version for both is 5.x.

In a comment to an on-line *Linux Journal* article (**www.linuxjournal.com/article/8253**), William Roddy wrote, "Scientific Linux will work in any environment Red Hat would, and even better. It's a work of art and genius, and in the field of high-energy physics, if this Linux didn't work, it wouldn't be used. Yet, it is useful to anyone. If you demand stability and security, you will not do better. It will always be there and it will always be free."—DOC SEARLS

Ubuntu Stays atop a Volatile Distro Market

By 1988, Larry Bird had already won the three-point shootout twice. In the locker room, while waiting to defend his title, he said to his opponents, "Who's finishing second?" Then he went out and won for a third time.

Ubuntu is in the same position. In DistroWatch.com's Page Hit Ranking, Ubuntu has been #1 for three years running. Mandrake (now Mandriva) was #1 in 2004. Ubuntu took over in 2005, then repeated in 2006 and 2007. Meanwhile, the "Who's finishing second?" question is always up for grabs, changing almost as constantly as all the other positions on down the list. Table 1 shows the results for three measures in 2008, taken on May 11.

Meanwhile, it's pretty clear that Ubuntu will hold its lead for at least the current year.

—DOC SEARLS

Table 1. Results

	2007	Last 3 months	Last 30 days	Last 7 days
1	Ubuntu	Ubuntu	Ubuntu	Ubuntu
2	PCLinuxOS	OpenSUSE	OpenSUSE	OpenSolaris
3	OpenSUSE	Fedora	Fedora	Puppy
4	Fedora	Mint	Mint	OpenSUSE
5	Sabayon	PCLinuxOS	Mandriva	Fedora
6	Mint	Mandriva	PCLinuxOS	PCLinuxOS
7	Debian	Debian	Debian	Slackware
8	MEPIS	Dreamlinux	Kubuntu	Mandriva
9	Mandriva	Sabayon	Slackware	Mint
10	Damn Small	Damn Small	Damn Small	Debian

Meet Mii at LinuxJournal.com

WiiLi (**www.wiili.org**) is a Linux port in the works for the Nintendo Wii. Drivers are being developed for Wii features, including the SD card slot, wireless 802.11b/g and Bluetooth hardware, and for the remote.

A proof-of-concept Wii distribution was rolled out earlier this year by the GameCube Linux Project. It relies on a Twilight Hack developed by Team Twiizers. The hack leverages an exploit in a Wii game title to launch a Linux bootloader (see a video of this in action at **www.linuxjournal.com/content/meet-mii**).

We're eager to see Linux on the Wii. In the meanwhile, settle for seeing *Linux Journal* on the Wii—we all have our own Miis here in the office. Add *Linux Journal* as a Wii friend, and we'll send you any *Linux Journal* staffers you'd like. Our console number is: 2924 2525 5556 2687 (post yours in the comments at **www.linuxjournal.com/content/meet-mii**, and we'll add you too).

—KATHERINE DRUCKMAN

What They're Using
Geoffrey Goodell

Geoff Goodell has a PhD in Computer Science from Harvard and is currently researching Internet surveillance and network neutrality. Recently, I found myself sitting next to Geoff, who was typing casually at a furious speed on his ThinkPad, while looking at a black screen divided into four quadrants, each apparently operating in command-line mode with colored type far too small for my old eyes to read, but obviously quite legible to Geoff. Because it was too hard for me to eaveswatch, even at close range, I asked him if he was an Emacs or a vi guy. He smiled and replied, "vi".

So, I asked him to let us know the rest of what he used, and here's his reply:

> I use a constellation of machines running Debian Linux. I store most of my working files (such as research, code, papers and personal Web pages) in AFS, which allows me to share data across machines seamlessly. For authentication, I use a combination of MIT Kerberos and OpenSSH public keys. I exchange multimedia files using rsync, and I use MPlayer for audio, video and streaming.

> For hardware, I use mainboards by Gigabyte, monitors by NEC and Samsung, IBM M13 keyboards and multihead video cards by NVIDIA. My laptop is an IBM ThinkPad T41, which also runs Debian Linux.

> For software, I try to keep a low profile. My window manager is ion2, which allows me to tile the screen with xterms and mostly avoid using the mouse. My favorite font is 6x10, and my color scheme is something that evolved over the years. For text editing, I use Vim. For e-mail, I use Mutt and GnuPG. For IRC, I use Irssi. For browsing, I use Firefox plus NoScript plus AdBlock Plus. For privacy, I also use Tor. I run Linuxnet IRC servers, Apache Web servers and an exim4 mail server. I coordinate non-IRC instant messaging through a set of gateways, so that my IRC client handles all the interaction. I use LaTeX to write papers, personal letters and presentations (occasionally I use OpenOffice.org for slides).

> My ISP is Speakeasy. My domain registrar is GANDI (in France).

> When not near a workstation or laptop, I have a Treo 750p (which runs a lovely SSH client called pssh) and a Motorola H12 noise-canceling Bluetooth headset.

—DOC SEARLS

Profiling Rails Applications

REUVEN M. LERNER

Wondering if your Rails application is running at peak efficiency?
Before optimizing, profile your application to see which parts are slow.

I am writing this article in mid-May 2008, several weeks after Twitter was rumored to be moving to a platform other than Ruby on Rails. Twitter, of course, is an extremely popular service that allows users to write updates and notes about their current status—and allows readers to follow any number of people's Twitter feeds. You can think of Twitter as a combination blogging and RSS platform, populated by people who express themselves with only 140 characters at a time.

Like many other runaway Internet successes, Twitter appears to have become too popular for its own good. This has led to some outages, most notably one at the beginning of 2008 that took more than a day to restore. Thus, it was seen as more than a mere coincidence when Twitter's main architect left the company, and that within a few days, the TechCrunch blog was quoting anonymous officials within Twitter about how the service would be transitioning away from Ruby on Rails.

This was followed by a great deal of discussion over whether Rails is a "scalable" architecture. Scalable used to mean that it was possible to scale up applications using a Web site, almost regardless of how many people are using it. But today, a scalable architecture is one that's lean and mean, handling as many users as possible with as few servers as possible. PHP, Java and .NET are pretty universally considered to be scalable in this sense. Although even the most efficient PHP application can handle only a finite number of simultaneous users, it's undeniable that Ruby is a slower language than PHP, and that the Rails framework adds some more overhead.

Of course, it's one thing to say that Rails doesn't scale as quickly as PHP, and another to say that it doesn't scale at all. And, there are other arguments to be made, including the fact that programmers cost more than servers, and that programmer productivity should be at least as significant a factor as scalability.

That said, it's easy for a Rails application to become slow. So, it is nice to know that a variety of utilities can be used to profile Rails applications—meaning, finding out exactly which portion of the program is taking a long time to execute. This

month, we look at some techniques for profiling Rails applications. Although such profiling doesn't make the software run any faster, it can help identify the slowest parts of an application.

Profiling Pages

If you aren't happy with the performance of your Web site—and quite frankly, you always should be concerned about the performance, trying to give it a boost wherever possible—the first question to ask is, "Where are people spending their time?" After all, if there are 100 different pages on your site, it doesn't really matter whether page 35 is really slow if no one ever visits it.

The first tool to examine, the production log analyzer, is designed to look at the Rails production log and produce some basic statistics about it. The production log, as well as the development and test logs, are typically stored in the log directory under the Rails project root. Thus, on a production server, the log is in log/production.log. This logfile is not rotated or modified automatically; there are clearly a number of ways to do that using cron and other UNIX command-line tools.

The thing is, there already is a facility on UNIX (and Linux) systems for handling logfiles, including their periodic rotation and disposal. This facility is known as syslog, which makes it possible to send logging information to a variety of different files based on priorities and source materials. The /var/log directory on my Ubuntu server is full of different logfiles, and nearly all of them were created and written to by syslog.

It turns out that we can use syslog for our Rails production logs. Once we have done that—and yes, we must use syslog for this to work—we then can analyze our production logs, learning exactly how much time people have spent in our Rails application.

To move your Rails production log to syslog, you need to do several things. First, you must install the Ruby gem that provides this behavior:

```
gem install --remote SyslogLogger
```

This installs the gem in the appropriate place

on your system; on mine, it was put into /usr/lib/ruby/gems/1.8/gem. Next, you need to add the following to one or more of your environment configuration files (either environment.rb or one or more files in environments/*.rb) for your Rails system:

```
require 'syslog_logger'
RAILS_DEFAULT_LOGGER = SyslogLogger.new
```

This, of course, loads the syslog_logger gem and sets the default logger to a new instance of SyslogLogger.

Now that you have told Rails to use syslog, you must tell syslog what to do with the files that come from Rails. I opened /etc/syslog.conf and added the following lines to the bottom:

```
*.info                    /var/log/production.log
```

And yes, the documentation system says that you can use a !rails tag before this line, or one like it, to restrict logging to messages coming from Rails.

Unfortunately, this syntax does not appear to be supported by Linux. So, this means production.log will include messages from other programs and facilities, not just Rails. That shouldn't concern us right now, although it might be an issue on a busy machine with many services in active use.

Once you have modified syslog.conf in this way, you can restart syslog.conf. Almost immediately, your production log should be stored to /var/log/production.log. You can check this, of course, with the following:

```
tail -f /var/log/production.log
```

Now, this logfile is similar in many ways to the logfile you just eliminated from the log directory in your application root. However, it is formatted in such a way that the production log analyzer will be able to find and perform calculations based on its output. To analyze the logfile, type:

```
pl_analyze /var/log/production_log
```

If you prefer to have the results sent to you via e-mail, rather than stored to a disk file, use the -e option:

```
pl_analyze /var/log/production_log -e
➥reuven@lerner.co.il
```

This option is particularly useful when you invoke pl_analyze from a cron job, for example.

The output file from pl_analyze is divided into three parts:

■ Time spent in each request.

■ Time spent in the database for each request.

■ Time spent rendering the output from each request.

For each controller action, pl_request lists how many times it was invoked, as well as the average time it took to execute. It also gives the min, max and standard deviation, allowing you to see how much the execution time varied over time.

Thus, the production log analyzer shows which actions take the greatest amount of time overall, which take the greatest amount of time in the database (or to render) and how many times each was invoked.

I have found pl_analyzer to be an indispensable tool when trying to determine whether a site is fast enough and where I should focus my attention to improve its speed.

Request Profiler

The production log profiler shows which actions require attention, but it doesn't tell why a particular action might be giving you trouble. For that, you need to dive into the application a bit more, profiling not a set of actions, but one particular action.

This is possible thanks to a built-in script that comes with Rails, script/performance/request. This script follows a set of instructions written in a (presumably short) Ruby program, using a similar set of commands and subroutines that are available for integration tests.

In other words, you use integration-test syntax to describe a short sequence of one or more actions and run this program via the request profiler. Then, the request profiler produces two output files that describe what was going on behind the scenes as those requests were serviced. This information can help you improve the performance of this particular action.

In order for this script to work, first install the ruby-prof gem:

```
gem install --remote ruby-prof
```

Once that is installed, you need to create a simple integration test script. This script doesn't need to be wrapped in the same object that the integration tests themselves use. Instead, simply create a file named test.rb, and put it somewhere on the filesystem. I created a directory named test/performance and put it there, with the one-line contents as follows:

```
get('/')
```

Notice that I'm using URLs here, rather than names of controllers and actions. Finally, with this in place, invoke the profiler:

```
script/performance/request -n 10 test/performance/test.rb
```

Now you should see the program telling you that it's warming up and then reporting as it goes through each of the iterations you specified. In the above example, the -n 10 option indicates the number of times the script should be invoked; by default, it's 100.

Note that the output files are put in the test directory (to which you might not have write access by default). And, indeed, the output files are quite useful, but they can be confusing the first time you look at them.

The first output file, profile-output.txt, is (as the suffix implies) a text file that shows how much time was spent in each method, both as a time measure and as a percentage of the total run time. Consider the following:

```
%self    total    self    wait    child    calls   name
13.74    58.35    38.13    0.00    20.22    608720  Buffer#read
```

Resources

The Rails Way, by Obie Fernandez, has become my favorite, because it includes so much useful information, as well as code examples. It doesn't try to teach you Rails, but it does provide a great deal of information that is useful for advanced users as well as newcomers.

Advanced Rails, by Brad Ediger, gives some greater depth to several topics, such as performance optimization, ActiveRecord features, RESTful sites and internationalization, among others.

Rails Analyzer Tools: this is a collection of tools that can help you better understand your Rails-based site. The production log profiler is part of the Rails Analysis Tools set; see **rails-analyzer.rubyforge.org** for more information.

This means there were 608,720 calls to Buffer#read during the test, which took a total of 38.13 seconds, or 13.75% of the execution time. Because this is a built-in method, you can't optimize it. However, you can try to reduce the number of times it is called, so that it will take even less time.

The question is, how do we know which functions are calling Buffer#read? Perhaps reading from buffers is an inevitable part of a Web application, and we just need to realize that?

If you look at the second file, profile-graph.html, you see a nicely linked description of which methods called which other methods, and how long it took. Each box represents the analysis of one method, and the method being analyzed is printed in bold.

All of the methods above the boldfaced method name are parent methods (that is, methods that called the one in question); whereas, methods below the current one are child methods (that is, methods that are called by the method being analyzed). By looking at who called Buffer#read, you can see which methods (if any) need optimizing or a smaller number of invocations. By going back and forth between methods, their parents and the source code, you can cut down on a great deal of waste, making your sites more efficient than before.
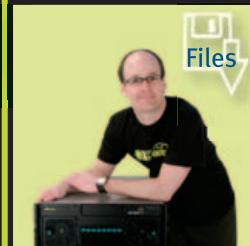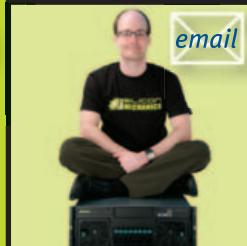
Conclusion

This month, we looked at two basic profiling tools that programmers can use to identify performance problems in Rails-based Web sites. There are, of course, other tools we can use, but the fact that these are so nicely integrated into Rails makes us all the more likely to use them. With constant monitoring and tweaking, we can make our sites run faster without having to resort to buying additional servers.∎

Reuven M. Lerner, a longtime Web/database developer and consultant, is a PhD candidate in learning sciences at Northwestern University, studying on-line learning communities. He recently returned (with his wife and three children) to their home in Modi'in, Israel, after four years in the Chicago area.

Cool as Ice!

No one will argue that there are different levels of cool. But nothing, and I mean nothing, says cool like a penguin. And some snow. And some ice. Oh, and the Antarctic. That's as cool as it gets.

MARCEL GAGNÉ

Mon Dieu, François! I realize it's a warm day outside, but it is positively freezing in here. Our guests will need coats, in August, no less. What are all these portable air conditioners doing here? Is that frost I see on the windows? François! I shudder—make that shiver—to think what this possibly could be about. Yes, this issue's theme is Cool Projects, but nowhere did it say frozen. And, when our editors said cool, I think they meant it in the sense of "really interesting and exciting". Never mind. Our guests will be here momentarily, and I don't think they are dressed for this. Quickly, run to Diane's *Manteaux de Cuir* across the street and beg her to provide us with some coats for tonight. *Vite!* Our guests are arriving as we speak.

Welcome, everyone, to a very chilly *Chez Marcel*. Please pardon the cold. My faithful waiter has once again taken a simple idea to its amusing, if somewhat outrageous, extreme. Nevertheless, he will return shortly with warm coats for all. In the meantime, please take your tables and make yourselves comfortable. Ah, François, you have returned with Diane. Thank you, Diane, for your help. While everyone slips into their coats, perhaps François can fetch the wine. There's a case of 2004 Bodegas Muga Reserva from Spain in the lower level of the cellar's east wing.

As François already has set the stage for us, we're going to explore some Linux coolness. The symbol of Linux coolness is, of course, the penguin. Tux, the Linux mascot (designed by Larry Ewing), is a penguin, and penguins show up pretty much everywhere you turn in the Linux world. In fact, you can't go near a Linux system, magazine, T-shirt, mouse pad, coffee mug or book, without running into some kind of penguin. That's okay for most people,

because, well, penguins are cute.

Ah, François, you have returned. Please, pour for our guests. Enjoy the wine, *mes amis*. This Spanish beauty is very rich, very complex, yet fresh tasting and well balanced. Hmm...make sure you fill my glass as well, François.

The first item on tonight's menu is Matthew Miller's *IceBreaker* (Figure 1). The premise is similar to events you see every day in the news. A bunch of penguins need to be captured and sent off to Finland. They are all on an iceberg in Antarctica, in an area where global warming hasn't yet started breaking the ice. Penguins, as it turns out, need to travel with ice or they just won't behave. The question becomes, "How much ice?"
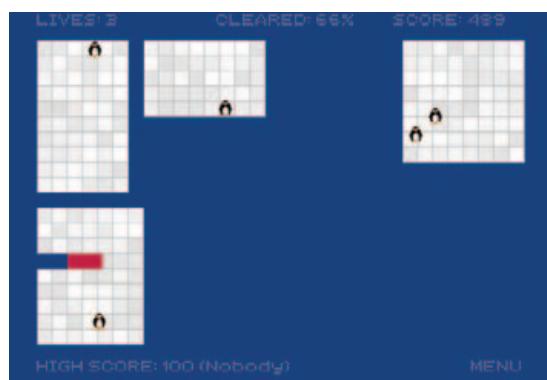
Figure 1. In *IceBreaker*, you must send your penguins packing to Finland with as little ice as possible.

As you can imagine, shipping penguins to Finland is expensive, so the order of the day is small ice chunks. When you left-click on the iceberg, a line is drawn across it, separating the two areas of ice. If a penguin hits the line as it is being drawn, your cut effectively is halted. A right-click changes the direction of the cut from horizontal to vertical (or vice versa). To clear an iceberg and move on, you need to clear at least 80% of the ice. Should you manage the job, another penguin is added and you get to start over on the next level.

The more penguins you add, the more complicated it becomes, as they bounce frantically across the ice field. On the off chance that you find this all too simple, there's a menu of options where you can change the difficulty level. Click MENU on the lower

Penguins and Linus Torvalds

Responsible for this whole penguin mania is Linus Torvalds, the Linux kernel's creator. When asked what he envisioned for a mascot, Linus replied, "You should be imagining a slightly overweight penguin, sitting down after having gorged itself, and having just burped. It's sitting there with a beatific smile—the world is a good place to be when you have just eaten a few gallons of raw fish and you can feel another burp coming."

Figure 2. If the game seems too easy, you can increase the difficulty.

Figure 3. *Snowball* is a combination jump-and-run and puzzle-solving game.

Figure 4. *Snowball*'s Options menu is the place to go for keyboard mapping.

right of the screen, and a pop-up menu appears (Figure 2). Not only can you change the difficulty here, but you also can turn sound effects on or off, check high scores and run the game in full-screen mode.

KBounce

A similar game called *KBounce* exists as part of the KDE games package, minus the cute penguins bouncing around.

Willi Kappler's *Snowball* (Figure 3) is a classic jump-and-run multiplatform game. It's also an interesting puzzler that requires a lot of thought before you can advance to the next level (of which there are 20). Your job is to find some way to help your penguin (Tux, in one of his many incarnations) release a trapped snowball and roll it into the exit. Along the way, you place (and remove) a limited number of ice blocks, collect gold coins and other treasure, all the while avoiding various dangers, including monsters. *Snowball* is written in Python, and it's available from **www.snowball.retrovertigo.de**.

On the right-hand side, a sidebar shows the number of available ice blocks as well as your current score and remaining lives. Using the Action key, you can either place or remove ice blocks. You use these blocks to climb to higher levels and to block the path of monsters. Find a way to release the snowball and guide it to the open doorway. The so-called Action key is, by default, the Enter key—something I found hard to manipulate when I was using cursor keys with the same hand. I switched the Action key to the spacebar instead via the Options menu (Figure 4).

Snowball hasn't been updated in a while, but it's still a lot of fun in its current form and sure to provide a few hours of frozen fun. It would be great if Willi

could be convinced to revisit his game or invite another developer to take over. If you want to try modifying *Snowball* on your own, there's an included level editor.

By now, you may have noticed that cold, snow, ice, penguins and Linux strangely seem to go together very nicely. Another great penguin-themed game, and one you must have a look at, is Ingo Ruhnke's *Pingus* (Figure 5). This is a game based on the classic *Lemmings* game (circa 1991), where you assist some friendly little creatures in escaping various dangers. *Pingus*, however, is much more than a clone. It has become a classic in its own right. Sit back, sip your wine, and relax while I tell you the *Pingus* story.

The Pingus, *mes amis*, have been living happily at the South Pole, presumably gorging themselves on fish. In time, their environment started to, er, go south, with the temperatures rising, the ice melting and the food supply getting tight. Rather than look for colder climes, as did other animals, the heroic Pingus decided to embark upon a quest to discover the source of this environmental havoc. You, as the

Figure 5. *Pingus* is the coolest *Lemmings* clone, ever!

Figure 6. The journey you face is difficult, and as such, you first must undergo rigorous training.

leader of the Pingus, will find yourself commanding hordes of Pingus who must be directed to safety or certain death. To do this, you instruct the Pingus to perform various tasks, which vary depending on the level you are currently playing. You need to think fast and act even more quickly to win each level.

If you feel up to the challenge of saving the Pingus, it's time to get your copy. The latest version of *Pingus* always is available from the *Pingus* Web site at **pingus.seul.org**. Packaged versions of *Pingus* are included with a number of different distributions, so check your distribution CDs or your usual software repositories. If you can't find it there, you always can pick up the latest package from the Web site.

When you launch *Pingus*, the main menu offers up four choices. One of these is labeled Story, and this is where you should begin (I will discuss the other choices shortly). Once you decide to embark on the journey, you will arrive at Mogorok Island, also known as Tutorial Island, to begin your training (Figure 6).

On subsequent starts, you will return to Tutorial Island, but you won't jump into the story immedi-

ately. The story of the Pingus (of which I've given you a short version) and why these poor penguins are on their perilous quest, is shown automatically only when you first play. If you want to reacquaint yourself with the tale, click the Show story check box at the top left.

At each level, there is an instruction screen providing some explanation of what you are facing and how you might deal with the challenges ahead (Figure 7). This can include digging through the ground or through walls, outfitting your penguins with backpacks to help them fly, turning some into blockers (so the others don't fall to their doom) and more. In some cases, you are forced to transform some of your penguins into bombers. Yes, that's exactly what it sounds like. Sometimes the only way to save the others is to sacrifice a few by making them blow up and, hopefully, blowing holes through whatever stands in the way of the others' safety.

Figure 7. With each new challenge, *Pingus* offers tips to help you on your quest.

The Tutorial Island I mentioned is a complete game in itself with nearly two-dozen levels. Once you leave Tutorial island, more Pingus action waits for you. Some of that action is easy to find, and some requires a little spelunking. From the intro screen, you can enter Tutorial Island by clicking the Story button. Click Levelsets, however, and a spooky Halloween adventure awaits. So far, neither of these options qualifies as hidden, and this is where the spelunking comes into play.

Each of these games, whether it is Tutorial Island or the Halloween adventure, is considered a level (or rather, a collection of levels). The path to the various levels is usually found in /usr/share/games/pingus/ data/levels (if you installed from source, the top-level directory won't be /usr, of course). Under that levels directory, you'll find the tutorial directory where the Tutorial Island levels are located. A couple additional

Figure 8. Hundreds of additional levels, including the Hellmouth, await you, if you dare to explore.

interesting directories are there, including one for your Halloween adventures.

Perhaps the most interesting ones are the wip (work in progress) directory and the playable directory. Do an `ls` in either of those directories, and you'll find a couple hundred other levels—not all of them playable, granted, but still fun to try. To run one of those levels, simply pass the full pathname to the pingus executable, like this:

```
pingus /usr/share/games/pingus/data/levels/
➥wip/hellmouth11-grumbel.pingus
```

Doing the above lets you play the game in the work in progress directory called hellmouth11 (Figure 8). Finally, should you feel so inclined, *Pingus* has a built-in level editor (that's the other menu option at the start), so you can create your own levels and contribute to the game's development.

Pingus is a great game, and it's great fun. The hidden levels offer a treasure trove of weird and wonderful side quests. Take my word for it, you need to check this one out. Besides, the future of the Pingus depends on you!

Ah, *mes amis*, I can see by the clock on the wall that we are nearly out of time, and there are still many penguins to save. Perhaps a few more minutes will bring us that much closer to getting our chilly friends to their goal. In the meantime, I'm sure we can convince

François to refill your glasses one more time while you huddle in your coats. I promise that the next time you arrive, I will make sure that the temperature approaches something more temperate. Raise your glasses, *mes amis*, and let us all drink to one another's health. *A votre santé! Bon appétit!*∎

Marcel Gagné is an award-winning writer living in Waterloo, Ontario. He is the author of the *Moving to Linux* series of books from Addison-Wesley. Marcel is also a pilot, a past Top-40 disc jockey, writes science fiction and fantasy, and folds a mean Origami T-Rex. He can be reached via e-mail at marcel@marcelgagne.com. You can discover lots of other things (including great Wine links) from his Web site at www.marcelgagne.com.

Resources

Penguin Logos of Every Kind: **www.linux.org/info/logos.html**

IceBreaker: **mattdm.org/icebreaker**

Pingus: **pingus.seul.org**

Snowball: **www.snowball.retrovertigo.de**

Marcel's Web Site: **www.marcelgagne.com**

The WFTL-LUG, Marcel's Online Linux User Group: **www.wftl-lug.org**

Movie Trivia and Fun with Random Numbers

DAVE TAYLOR

Use the shell to manipulate a list of movies from the Internet Movie Database (IMDb).

Last month, we had a lot of fun digging around within the Internet Movie Database, producing a set of scripts that together make it easy to generate a list of the top 250 movies on the site with release dates. The format of the output is:

```
All About Eve | 1950

Hotel Rwanda | 2004

Sin City | 2005

City Lights | 1931
```

This month, I take a look at how you can break those two fields up and randomly generate some likely release dates close to the actual date, then send it as a question on Twitter. For example, it might ask, "Hotel Rwanda was released in: 2000, 2001, 2004 or 2007?"

Splitting Up the Fields

Okay, this should be super easy for anyone reading this column. There are a bunch of ways to

> **The solution is a secret notational convention you can use in scripts when there's any sort of ambiguity like this—curly brackets.**

take a two-field data record and split it up, but my favorite tool for this sort of task is cut. So, we can do this:

```
moviename="$(echo $entry | cut -d\| -f1)"
releasedate="$(echo $entry | cut -d\| -f2)"
```

That was easy, right? Now, of course, if you want to be fancy about it, you'll want to strip any leading or trailing spaces too, which can be done with this sed command:

```
sed 's/^ //g;s/ $//g'
```

But, how do you get a random line out of a text file?

If you recall from previous columns, one of the secret features of the Bash shell's built-in mathematical capabilities—accessible with $(( )) notation—is the ability to get a random integer without any further fuss, like this:

```
echo $(( $RANDOM ))
```

Try it in your own command shell a few times, and you'll get a series of random integer values, like 29408 and 17501. To constrain it to the size of the file, we could do something fancy with `wc -l` to identify the number of lines in the actual data file, but because we already know we're grabbing 250 film titles from IMDb, it's easy just to use that value. Here's the first stab:

```
pickline="$(( $RANDOM % 250 )) "
```

It's not quite right though, because we'll get values 0–254. You can verify this by entering the command `echo $(( 5 % 5 ))`, for example. So, we need to shift things up one:

```
pickline="$(expr $(( $RANDOM % 250 )) + 1 )"
```

That produces a random number. To extract that value from a file of lines, there are a number of solutions, but I'll stick with sed. In that case, the solution for pulling out line 33, as an example, is:

```
sed -n 33p
```

If you change the value to a variable name, however, there's a problem:

```
sed -n $picklinep
```

You can't put a space between the variable name and the p, but if you don't, you have a bad variable name, because it's pickline, not picklinep. The solution is a secret notational convention you can use in scripts when there's any sort of ambiguity like this—curly brackets.

So, the line ends up as follows:

```
sed -n ${pickline}p
```

That does the trick, and in an application like this, sed is lightning fast too.

At this point, we have a data file of

```
    newvalue=$(expr $1 - $delta )
fi
```

This script can be tested easily by dropping it into a simple script, which I'll call random-years.sh. The result of applying this to the starting year 2000 is

Just like with the SAT and GMAT, it's important to avoid any possible patterns in answers.

interesting information, we can extract a random line from the file, and we can split the resultant data into the film title and release year. How about coming up with plausible alternative release years?

Calculating Random Years

My first inclination with generating random years was to add and subtract 1–3 years and then use those as the alternate values. If we were looking at, say, *Shaun of the Dead*, released in 2004, we might end up with 2001 and 2007 as the options. Match a film that's more recent though, such as 2007's *Grindhouse* (though why that's on the IMDb top 250 films list is beyond me), and we have a problem. Suggesting 2009 as a possible release date would be daft.

More important, it wouldn't take long for people to realize that it's the middle value that's always correct on the quiz—not good. Just like with the SAT and GMAT, it's important to avoid any possible patterns in answers.

As a result, we can try something a bit more complicated. Each possible year is the actual year of release plus or minus a random value of 1–5—close enough that it'll be challenging to remember the right year.

Here's the beginning of the script:

```
add="$(( $RANDOM % 2 ))"
delta="$(expr $(( $RANDOM % 5 )) + 1)"
```

Here, add will be 0 (false) or 1 (true) for later conditional testing, and delta is a value between one and five, just as we need. They can be applied as follows:

```
if [ $add -eq 1 ] ; then
  newvalue=$(expr $1 + $delta )
else
```

2002, 1998, 2005, 2001, 2003, 2004. Seems sufficiently random, yes?

Now, let's consider some nuances. First, we need to ensure that it's never past the current year, which can be done by grabbing that value from the date command with a format string: `date +%Y` (learn more about the many, many format strings that the date command understands with `man strftime`).

Second, here's a more interesting thought. If the movie came out a long time ago, we should have a bigger delta than if it's a recent release. In other words, if the movie is *Casablanca*, it came out in 1942, 66 years ago. *Iron Man*, which is also on the top 250 list, came out in 2008, 0 years ago. For *Casablanca*, we could have possible values of 1938 and even 1951, and it'd be a good quiz question for anyone who isn't a complete film nut. But, that far of a spread for *Iron Man* makes no sense. No one's going to think it might have come out in 1999.

What I'm thinking about in this situation then is that the delta might be a percentage of the age of the movie, normalized so that we always have some sort of spread. Maybe 20%? That'd give us a delta of 13.2 for *Casablanca* and 0 for *Iron Man*. That could work.

Ah, but I've run out of space. Next month, we'll go back to the random adjacent year function to wrap it up, and then look at how to get these questions out on Twitter rather than just on the Linux command line. Until then, "here's lookin' at you, kid."∎

Dave Taylor is a 26-year veteran of UNIX, creator of The Elm Mail System, and most recently author of both the best-selling *Wicked Cool Shell Scripts* and *Teach Yourself Unix in 24 Hours*, among his 16 technical books. His main Web site is at www.intuitive.com, and he also offers up tech support at AskDaveTaylor.com. Follow him on Twitter if you'd like: twitter.com/DaveTaylor.

Wiimote Control

KYLE RANKIN

Why let your Wii have all the fun? Find out how to connect your Wiimote to your computer and use it as a mouse or an input device for any number of popular gaming emulators.

If you think about it, there are almost as many ways to interface with your computer as there are Debian-based distributions—and that's a lot. Besides the trusty keyboard and optical mouse, there are trackpoint mice, touchpads, touchscreens, twiddlers, joysticks, presentation remotes and even devices that measure your brain waves. Although I mostly stick with my tried-and-true keyboard and trackpoint mouse (fingers on home row, thank you), when I started hearing about all the interesting things people were doing with the Wiimote (the main controller from the Nintendo Wii), I knew I had to give it a try.

Now traditionally, connecting a brand-new device to a Linux machine was an investment in Internet research, kernel module hacking, prayer and obscure programming skills I haven't used since college. I figured the mere fact that this was a Bluetooth device meant I was going to have to spend some quality time with hcidump. To my surprise, all the hard work already had been done for me, and I could connect and use a Wiimote on my laptop with only a few basic steps.

Configure udev

First, your kernel needs the uinput module available and loaded. This module is available in modern kernels, and my Ubuntu Gutsy install already had it. If you want to be able to connect to the Wiimote as a

The great thing about wminput is that all these mappings are completely configurable.

regular user, however, you need to add a new udev rule to extend permissions to the uinput device. I created a file called /etc/udev/rules.d/95-uinput.rules that contained the following:

```
KERNEL=="uinput", GROUP="plugdev"
```

Then, I made sure my user was a member of the plugdev group. If your system doesn't have a plugdev group, you could choose or create another group to use for this device. Next, run `/etc/init.d/udev reload` to make sure your changes are seen. Finally, I ran `modprobe uinput` to make sure the module was loaded, and I also added uinput to /etc/modules to make sure it was loaded at boot.

Install wminput

The next step is to install the wminput software. For me, this was simple, as wminput is packaged for my distribution; otherwise, you can download the source from the official site (**www.cwiid.org**). Then, make sure the Bluetooth device in your computer is enabled. For my laptop, I had to flip a switch on the side, but if you have an external USB Bluetooth adapter, for instance, now is a good time to plug it in. Finally, run wminput in a console and follow the directions:

```
greenfly@minimus:~$ wminput
Put Wiimote in discoverable mode now (press 1+2)...
Ready.
```

When you press buttons 1 and 2 on your Wiimote, it goes into discoverable mode, and the blue LEDs along the bottom start blinking. Sometimes you might not start discoverable mode fast enough, or wminput won't detect it, but as long as the LEDs on the Wiimote are blinking, it is still in that mode. So if wminput times out, just run the program again.

If you continually can't connect, you probably should double-check that your Bluetooth device is working. To do this, press buttons 1 and 2 on the Wiimote and then use hcitool to scan for the Wiimote. A successful scan will look like the following:

```
greenfly@minimus:~$ hcitool scan
Scanning ...
        00:1B:7A:3E:8C:54     Nintendo RVL-CNT-01
```

After wminput connects, you also can look in /var/log/dmesg for confirmation that the Wiimote is connected:

```
[ 1226.247203] usb 3-2: new full speed USB device using
➥uhci_hcd and address 13
[ 1226.288768] usb 3-2: configuration #1 chosen
➥from 1 choice
[ 1227.922403] input: Nintendo Wiimote as
➥/devices/virtual/input/input21
```

Use the Wiimote as a Mouse

Once the Wiimote is connected, the default bindings use it as a mouse. The accelerometers in the Wiimote are used to move the mouse pointer, so if you point the

Wiimote down or up, the mouse will move down or up, respectively, and if you roll the Wiimote to the left or right, the mouse will move left or right, respectively. If you look at /etc/cwiid/wminput/buttons, you can see the default mappings:

```
Wiimote.A               = BTN_LEFT
Wiimote.B               = BTN_RIGHT
Wiimote.Up              = KEY_UP
Wiimote.Down    = KEY_DOWN
Wiimote.Left    = KEY_LEFT
Wiimote.Right   = KEY_RIGHT
Wiimote.Minus   = KEY_BACK
Wiimote.Plus    = KEY_FORWARD
Wiimote.Home    = KEY_HOME
Wiimote.1               = KEY_PROG1
Wiimote.2               = KEY_PROG2
...
```

By default, wminput reads the configuration listed in /etc/cwiid/wminput/default to get its mappings. In this file you will see:

```
#acc_ptr

include buttons

Plugin.acc.X    = REL_X
Plugin.acc.Y    = REL_Y
```

Essentially, this file includes the buttons file for keybindings, and it also enables the use of the accelerometers for X and Y movements. The great thing about wminput is that all these mappings are completely configurable. If you look in /etc/cwiid/wminput, you should see a number of other example mappings you can use as inspiration. You also can store custom mappings in your home directory under ~/.cwiid/wminput. The button mappings use standard names for keys and mouse buttons that can be found in /usr/include/linux/input.h, but most of the names are pretty straightforward.

Wiimotes for NES Emulation
One of the first things I wanted to do with my Wiimote after it was connected was to use it as a controller for my various game system emulators. But, before I go any further, if you do use a game system emulator like nestra, fceu, snes9x or MAME, be sure you have full rights to use any ROMs you might have. Make an appointment with your lawyer for details, but essentially, to play a commercial ROM, you should own the corresponding game.

With the legal disclaimers aside, the Wiimote makes a great wireless NES (Nintendo Entertainment System) controller. All the basic buttons are there, and all that's left to do is re-arrange the button

mappings to work with either nestra or fceu NES emulators. Both programs use slightly different mappings, so I created files called buttons-fceu and buttons-nestra and placed them in ~/.cwiid/wminput. First, buttons-nestra:

```
Wiimote.A               = KEY_0
Wiimote.B               = KEY_1
Wiimote.Up              = KEY_LEFT
Wiimote.Down    = KEY_RIGHT
Wiimote.Left    = KEY_DOWN
Wiimote.Right   = KEY_UP
Wiimote.Minus   = KEY_TAB
Wiimote.Plus    = KEY_ENTER
Wiimote.Home    = KEY_PAUSE
Wiimote.1               = KEY_Z
Wiimote.2               = KEY_SPACE
```

After I set the regular NES buttons, I had a few extra to bind, so I bound the A button to pause the emulator, the B button to set nestra to normal speed and the home button to reset the game.

The fceu emulator had completely different keybindings, so here is my buttons-fceu file:

```
Wiimote.A               = KEY_F7
Wiimote.B               = KEY_F5
Wiimote.Up              = KEY_A
Wiimote.Down    = KEY_S
Wiimote.Left    = KEY_Z
Wiimote.Right   = KEY_W
Wiimote.Minus   = KEY_TAB
Wiimote.Plus    = KEY_ENTER
Wiimote.Home    = KEY_F10
Wiimote.1               = KEY_KP2
Wiimote.2               = KEY_KP3
```

In addition to the standard buttons, I bound the B button to save a game state, A to restore and home to reset the game.

Now, to use either of these configuration files, all I need to do is tell wminput to load these files instead:

```
greenfly@minimus:~/$ wminput -c ~/.cwiid/wminput/buttons-nestra
Put Wiimote in discoverable mode now (press 1+2)...
Ready.
```

Because wminput sends regular keyboard events, I don't have to do anything special to nestra or fceu.

Wiimotes for SNES Emulation
The Wiimote worked great for NES games, but how about SNES (Super Nintendo) emulation? I actually purchased a few different SNES games for the Wii virtual console, and I also bought a Classic Controller so I would have all the standard SNES buttons. It turns out that wminput also can bind

keys to the Nunchuck and Classic Controller attachments, so all I had to do for it to work with snes9x was create a new configuration file that mapped all the keys. Here is my buttons-snes9x file:

```
Wiimote.A               = KEY_X
Wiimote.B               = KEY_S
Wiimote.Up              = KEY_LEFT
Wiimote.Down    = KEY_RIGHT
Wiimote.Left    = KEY_DOWN
Wiimote.Right   = KEY_UP
Wiimote.Minus   = KEY_TAB
Wiimote.Plus    = KEY_ENTER
Wiimote.Home    = KEY_ESC
Wiimote.1               = KEY_C
Wiimote.2               = KEY_D

Nunchuk.C               = BTN_LEFT
Nunchuk.Z               = BTN_RIGHT

Classic.Up              = KEY_UP
Classic.Down    = KEY_DOWN
Classic.Left    = KEY_LEFT
Classic.Right   = KEY_RIGHT
Classic.Minus   = KEY_SPACE
Classic.Plus    = KEY_ENTER
Classic.Home    = KEY_ESC
Classic.A               = KEY_D
Classic.B               = KEY_C
Classic.X               = KEY_S
Classic.Y               = KEY_X
#Classic.ZL             =
#Classic.ZR             =
Classic.L               = KEY_A
Classic.R               = KEY_Z
```

Even though I planned to use the Classic Controller, I tried to map as many of the regular Wiimote keys to buttons that made sense, so you could potentially play at least some games with the regular Wiimote as well. If you notice, I also left bindings for the special ZL and ZR keys blank, so you could bind them to extra keys.

Wiimote Control for MAME

One of the best game system emulators out there is MAME. MAME emulates classic arcade games, and there are many guides out there (including in *Linux Journal* itself) on how to use MAME to create your own arcade cabinet. Well, I haven't cleared away the time for that project yet, but I did want to use my Wiimote and Classic Controller attachment for MAME games. MAME has a large number of bindings (press Tab in MAME to see a list), so it was difficult to choose which to bind to the extra keys. Here is a sample buttons-xmame file I created:

```
Wiimote.A               = KEY_P
Wiimote.B               = KEY_5
Wiimote.Up              = KEY_LEFT
Wiimote.Down    = KEY_RIGHT
Wiimote.Left    = KEY_DOWN
Wiimote.Right   = KEY_UP
Wiimote.Minus   = KEY_2
Wiimote.Plus    = KEY_1
Wiimote.Home    = KEY_F3
Wiimote.1               = KEY_LEFTCTRL
Wiimote.2               = KEY_LEFTALT

Nunchuk.C               = BTN_LEFT
Nunchuk.Z               = BTN_RIGHT

Classic.Up              = KEY_UP
Classic.Down    = KEY_DOWN
Classic.Left    = KEY_LEFT
Classic.Right   = KEY_RIGHT
Classic.Minus   = KEY_2
Classic.Plus    = KEY_1
Classic.Home    = KEY_F3
Classic.A               = KEY_LEFTCTRL
Classic.B               = KEY_LEFTALT
Classic.X               = KEY_SPACE
Classic.Y               = KEY_LEFTSHIFT
Classic.ZL              = KEY_5
Classic.ZR              = KEY_P
Classic.L               = KEY_Z
Classic.R               = KEY_X
```
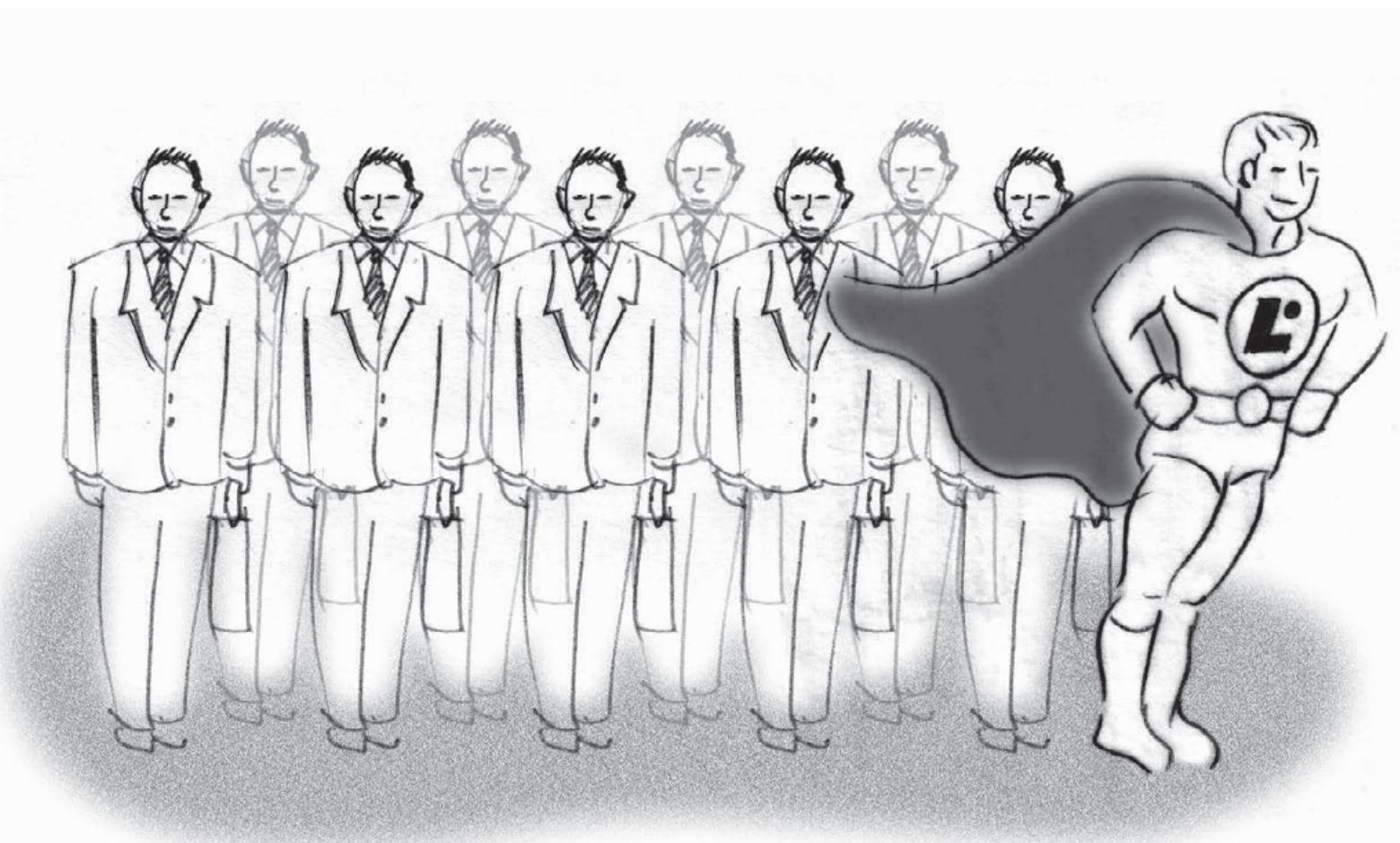
In addition to the standard bindings you might expect, the home key resets MAME; the plus key selects single player; minus selects two players; ZL on the Classic Controller and B on the Wiimote insert a coin; and ZR on the Classic Controller and A on the Wiimote pause. These are by no means perfect bindings, so I recommend you experiment with different keys that work better for you.

The possibilities with wminput go much further than what I've presented here. There also are configuration files that use the analog joystick inputs on the Classic Controller, the IR sensors on the Wiimote and the accelerometers on the Nunchuck. Wminput isn't just a handy way to play video games on your laptop or desktop. The fact that the connection to the computer is wireless makes the Wiimote a great gaming input for a MythTV client or other computer connected to your PC. As for me, I think I'll be spending a few more days trying to beat this impossible Super Mario Brothers hack that has been floating around the Internet.∎

Kyle Rankin is a Senior Systems Administrator in the San Francisco Bay Area and the author of a number of books, including *Knoppix Hacks* and *Ubuntu Hacks* for O'Reilly Media. He is currently the president of the North Bay Linux Users' Group.

WHY LPI CERTIFICATION?

RELEVANCE

- #1 Linux certification worldwide and growing
- Program framework created from industry needs and input
- Professional "Job Task Analysis"

CREDIBILITY

- Designed by professionals for professionals
- Internationalization through regional involvement
- Endorsed by global leaders in Open Source
- Recognized and accredited psychometric processes

VALUE

- A global standard in Linux professionalism
- Proven demonstration of knowledge and skills for customers and employers
- Provides benchmarks for HR recruitment and promotion
- Access to global network of professionals

Linux Professional Institute
LPIC-1

Linux Professional Institute
LPIC-2

Linux Professional Institute
LPIC-3

OSPRE

L® Linux Professional Institute

www.lpi.org

Linux Game Publishing and Ascaron Entertainment's *Sacred Gold*

Our team is truly tickled at how many high-end Linux-based games are now at our disposal. One of the latest is Ascaron Entertainment's *Sacred Gold*, which includes *Sacred* and its expansion, *Sacred: Underworld*. Linux Game Publishing is responsible for the Linux port. The companies plug *Sacred* as an action-filled role-playing game that "combines an exciting story line with great gameplay". In addition to porting a broad range of Linux-based games, Linux Game Publishing also supports open-source development by making available a number of libraries it has developed over time for its games.

www.linuxgamepublishing.com, eng.sacred-game.com

Jedox AG's Palo

Though you likely never will experience a GPL'd Microsoft Excel, you can use the open-source Palo 2.5 from Jedox to serve up Excel spreadsheets. Palo is a multi-user, high-performance data server application that allows workers enterprise-wide to access, change and collaborate on multiple spreadsheets in real time. Improvements in the new version 2.5 include a newly optimized MOLAP (Multidimensional OnLine Analytical Processing) engine, intelligent local data cache, faster multidimensional data processing, an enhanced multidimensional formula editor and advanced query capability. The workstation-resident data cache uses an "intelligent" technology to reduce calls to the central server. Palo is available in free, enterprise and government editions.
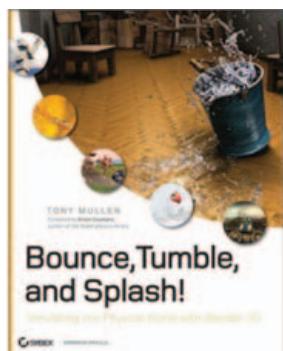
www.jedox.com

SugarCRM's Sugar Data Center Edition

Diversifying the open-source CRM space is SugarCRM with its new Sugar Data Center Edition. The new product line offers a complete set of systems management, provisioning and monitoring tools that enable service providers and large organizations to deploy and manage multiple instances—distinct versions of SugarCRM—from a centralized management console. In the absence of these capabilities, says SugarCRM, large enterprises are forced to eliminate multiple instances in a subdivision of their organization and make serious trade-off decisions regarding functionality and customizations/localizations for the sake of centralized and Web-services-based applications. With the new Sugar Data Center Edition, organizations "can get creative and deep with customizations and locations at no expense to one department or end user".

www.sugarcrm.com

Tony Mullen's *Bounce, Tumble, and Splash!* (Sybex)

To our squeals of delight, Sybex is tearing off its Clark Kent-like demeanor to present Tony Mullen's B*ounce, Tumble, and Splash! Simulating the Physical World with Blender 3D*. Blender is an immensely popular, multiplatform, open-source, 3-D content-creation suite. *Bounce, Tumble, and Splash!*, says Sybex, is the only title to offer "step-by-step instructions on Blender's more complex features while showcasing the unique objects and characters that can be created in Blender". Topics include soft bodies and cloth, the Blender particle system, static particles and hair, fluids, bullet physics, the Blender Game Engine and plant simulation. The book's tone is "friendly but professional" and focuses on full-color examples with clear, in-depth explanations of how each step was taken and why each choice was made.

www.sybex.com

Syuzi Pakhchyan's *Fashioning Technology* (O'Reilly)
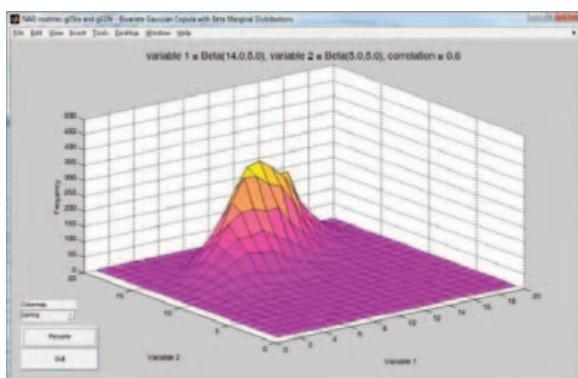
Geeks, start your...sewing machines! Such is the wish of Syuzi Pakhchyan, author of the new O'Reilly book *Fashioning Technology* that explores the integration of traditional sewing and assembly techniques with electronics and other new materials. The book is a guide to inventing creative clothing, housewares and toys that are fun, interactive, quirky and useful. Author Pakhchyan—an artist, roboticist and teacher—explains how to use smart materials such as thermo- and photochromatic inks that change color by touch or sunlight, magnetic and conductive paints, polymorph plastic, fiber optics and more. Each project, says O'Reilly, encourages readers to personalize and customize their own designs, materials and craft skills.

www.oreilly.com

ParAccel's Scalable Analytic Appliance

The job of ParAccel's new Scalable Analytic Appliance is to provide manageability for large- and medium-size enterprises struggling with the challenge of analyzing operational data in near real time or executing complex queries on multi-terabyte data warehouses. The new enterprise-class appliance is based on ParAccel's columnar, compressed, massively parallel relational database engine, combined with a managed storage infrastructure and industry-standard servers. The appliance utilizes a blended and dynamically balanced scan approach to take maximum advantage of both server- and SAN-based storage. It also leverages a new SAN-based approach for high availability and integrates tightly into managed storage control systems to manage backups, disaster recovery mechanisms, reporting and monitoring. A pilot program for the product is currently underway.

www.paraccel.com

Numerical Algorithms Group NAG Toolbox for MATLAB

If you use the MATLAB environment, you now can extend it heftily using Numerical Algorithms Group's NAG Toolbox. The Toolbox gives users access to more than 1,300 additional math and statistical algorithms for MATLAB. This additional mathematical and statistical functionality previously was unavailable, or it was accessible to MATLAB users only by purchasing multiple toolboxes. The company claims that "the NAG Library is used by many of the world's most prominent ISVs, scientists and academies, among others, because of its reputation for quality, flexibility and robustness". The NAG Toolbox is available for both 32- and 64-bit Linux and Windows and is compatible with MATLAB versions 2007a, 2007b and 2008a.

www.nag.com

Canonical's Ubuntu Netbook Remix

At the time of this writing, details remain sketchy, but by the time you read this, Canonical will have officially announced Ubuntu Netbook Remix, an ultraportable version of its popular Linux distribution. In interviews with the *Guardian* newspaper, Ubuntu founder and patron Mark Shuttleworth, revealed close collaboration with Intel, which produces chips for this sector. Shuttleworth sees Netbook Remix as one way that Linux will become more prevalent, as people access their files and information from a wider variety of devices connected to the Internet.

www.ubuntu.com

Fresh from the Labs
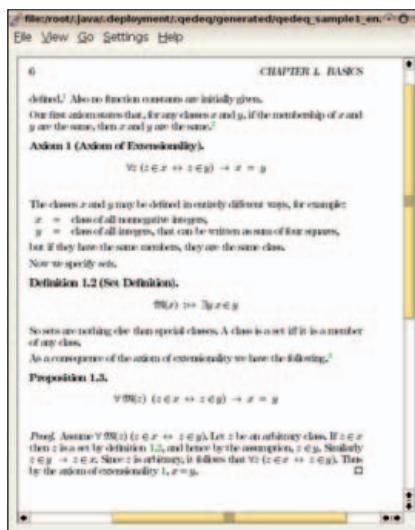
Hilbert II
(www.qedeq.org)

Here's one for the mind-bending cate-gory. In this age of shared information and decentralization comes another cool addition to the realm of shared consciousness. Hilbert II attempts to resurrect and build on the ideals of a near-dead project, QED (check the link at the end for a copy of QED's manifesto). Hilbert II's goals are:

> ...decentralised access to verified and readable mathematical knowledge. As its name already suggests, this project is in the tradition of Hilbert's pro-gram....Hilbert II wants to become a free, world-wide mathematical knowledge base that contains mathematical the-orems and proofs in a formal correct form. All belonging documents are published under the GNU Free Documentation License. We aim to adapt the common mathematical argu-mentation to a formal syntax. That means, whenever in mathe-matics a certain kind of argu-mentation is often used, we will look forward to integrate it into the formal language of Hilbert II. This formal language is called the QEDEQ format.

Hilbert II provides a program suite that enables a mathemati-cian to put theorems and proofs into that knowledge base. These proofs are automatically verified by a proof-checker. Also, texts in "common mathematical lan-guage" can be integrated. The mathematical axioms, definitions and propositions are combined to so-called QEDEQ modules. Such a module could be seen as a mathematical textbook that includes formal correct proofs. Because this system is not cen-trally administrated and refer-ences to any location in the Internet are possible, a world-wide mathematical knowledge

base could be built. Any proof of a theorem in this "mathemat-ical web" could be drilled down to the very elementary rules and axioms. Think of an incredible number of mathematical text-books with hyperlinks, and each of its proofs could be verified by Hilbert II. For each theorem, the dependency of other theorems, definitions and axioms could be easily derived.

The Complex World of Mathematical Collaboration through Hilbert II

Installation First things first, Hilbert II is a Java-based program. We generally try to avoid Java-based projects because this is *Linux Journal*, not "Platform-Neutral Java Webstart Journal", but the cooler projects are definitely worth examining, and besides, it does have a Linux-specific version. For the lazy, there is a webstart version that can be launched from your browser (see the link at the end of this section), which requires you to have the Java browser plugins installed. For the Linux version, the precondition for a working proto-type is a Java Runtime Environment, at least version 1.4. Hilbert II uses LaTeX for a lot of its functions, and there are some potential bugs that frequent users may run into, so check the Web site for further information on possible LaTeX requirements.

Head to the Download section of the Web site and download the Linux tarball. Extract it to your directory of choice, and open a terminal in the new qedeq directory.

At the console, enter:

```
$ ./qedeq_se.sh
```

Or, if that doesn't work, enter:

```
$ sh qedeq_se.sh
```

Usage Hilbert II works around XML files, and you'll need some of these XML files to get started. If you choose File→Load from web, a default file is provided from which you can begin to experiment. If you look in the main window, there's a tab called QEDEQ. Clicking on any entry on the left displays its nuts and bolts in this tab. Under Tools→LaTeX to QEDEQ, you can start playing around with your own formulae, and under Check→Check Mathematical Logic, you can make sure that your syntax and so on check out. To export your work for the world to see, going to Transform→Create LaTeX output creates a new LaTeX .tex file in the generated folder under qedeq.

But from here, you're on your own, because I haven't got a clue what I'm talking about in this world of advanced mathematics and formulae (and I've probably said something wildly inaccu-rate in the process of writing this sec-tion). However, I'm keen to see the results of this academic collaboration, where ideally knowledge should keep advancing and continue to be built upon, and I hope to see more of these mind-bending shared-consciousness projects.

- Webstart Version: **www.qedeq.org/ 0_03_10/webstart/qedeq.jnlp**

- QED Manifesto: **ftp.mcs.anl.gov/ pub/qed/manifesto**

vitetris
(victornils.net/tetris)

Ever been stuck working on a text-only mailing server and wished you had

some sort of decent gaming distraction? Well, you have a lot of options, such as adventure text games and *moon-buggy*, but my favorite discovery is *vitetris*, a *Tetris* clone with full color and many options. According to the *vitetris* Web site:

> *vitetris* is a terminal-based *Tetris* clone by Victor Nilsson. Gameplay is much like the early *Tetris* games by Nintendo. Features include:

- Configurable keys

- Highscore table

- Two-player mode with garbage

- Network play

- Joystick (gamepad) support on Linux

It has been tested on Linux, Cygwin, NetBSD and a few other UNIX-like systems. Library dependencies are minimal (only libc is required), and many features can be disabled at compile time.

Installation For those who prefer

binaries, included at the Web site are links to RPM packages and some tarballs built with gcc 3.4.6 for i486 Linux on Slackware 11.0. However, *vitetris* has very few dependencies, and 99% of you should be able to compile it from the source tarball (saving you from some of the inevitable binary incompatibility). Indeed, this is the easiest and most trouble-free compilation I've encountered in a long time, so I recommend compiling it.

Grab the latest tarball from the project's Web site, extract the contents, and open a terminal in the new folder. Once inside the vitetris directory, enter the commands:

```
$ configure
$ make
```

and, as root or sudo:

```
# make install
```

Once compiled, typing `tetris` at the command line loads the game.

Usage Once inside the game, you'll see a heap of cool options. For instance, you can change the height of the level you're in, enable rotation in both clockwise and counter-clockwise directions, and switch between game modes. These two game modes enable or disable attacking the other player with

completed lines and adding them to the bottom of their stack (game mode A is for attacking enabled, and B is for disabled). To start a game by yourself, choose 1 Player Game, and choose your difficulty level and game height to begin. On your keyboard, the left and right arrows move each piece left and right; the up arrow rotates the piece on screen; the down arrow makes a "soft drop"; and the spacebar makes a "hard drop", straight to the bottom of the screen.

If you want to change the keys or switch between rotation methods and so on, you can to that from the Options menu. If you want to play a two-player game, you also have to define Player 2's keys here. If you're having any problems displaying *vitetris* in your console and want to change the game's colors, or even switch to a monochrome mode, those options are available in the Options menu as well.
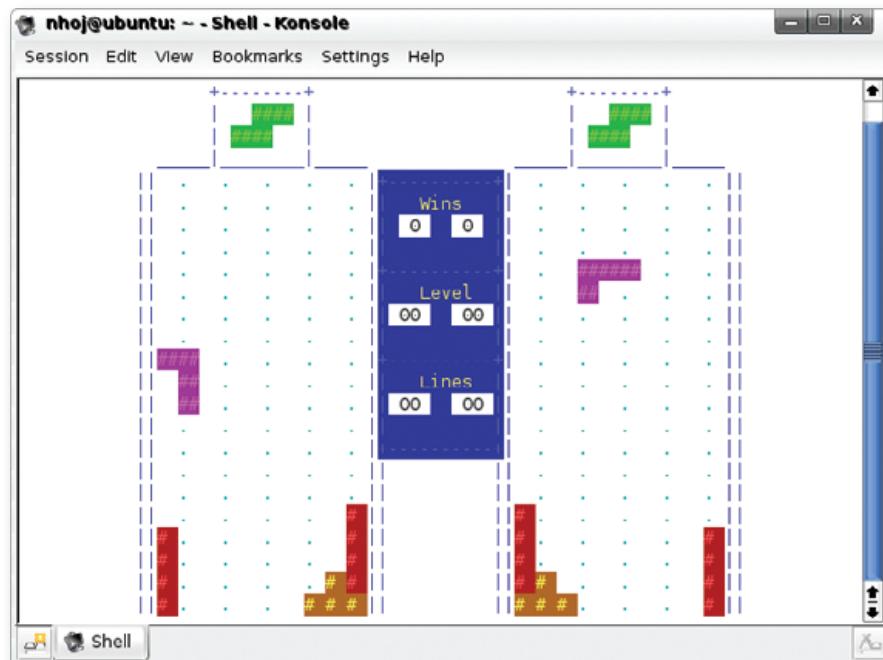
Ultimately, *vitetris* is a great *Tetris* clone by itself, but coupled with the fact that it runs on the command line without graphics, *vitetris* is a great addition to any system and will be a nice distraction the next time the X Window System won't start!

Tetuhi

(halo.gen.nz/tetuhi/code.html)

This was the craziest project I came across this month! Tetuhi is basically a program that takes an image and generates a game around it, but its appeal doesn't end there. Aside from making landscapes from parts of the image, Tetuhi also creates characters from other parts of the image, as well as other objects, such as food, ammo, friends and enemies, which all wriggle and move about as the engine morphs sections of the original image. On top of all that, it also has a dynamic and adaptive rule set with changing game modes—meaning each game and image may be truly random and different from the last.

Installation Tetuhi is definitely something that is still in development, so the usual `configure && make && make install` won't do you much good here. In terms of requirements, you need up-to-date versions of Python, GCC, Pygame, the Python Imaging Library, PyYAML and the Gnu Scientific

vitetris provides two-player *Tetris* fun without graphics.

I realise this crayon drawing itself doesn't show off Tetuhi's capabilities, but imagine that the hills and trees are pulsating in front of you and you are driving the sun....No, I'm not on mushrooms!

Library. Once you've installed those, head to the Tetuhi Web site and grab either the latest tarball or the latest code from the GIT repository.

Once you have either of those, extract it (if you have the tarball), and look at the directories c, img-c and perceptron. Open a terminal, and enter each one of these directories and run the commands:

```
$ make
```

and, as root or sudo:

```
# make python-install
```

This should work in all three directories without errors. If not, make sure you have all of the previously mentioned libraries installed and up to date.

Usage Now that the compiling is out of the way, head back to the main tetuhi directory, and enter the command:

```
$ ./tetuhi nameofimagehere.jpg
```

If everything has compiled properly, an image with some crazy instructions

should appear on screen, walking you through the first steps of the game. The best types of images to use are those with simplicity, such as stark backgrounds with bold elements at the forefront. Included on the Tetuhi Web page is a link to a tarball containing sample images for testing. My favorite is "hills-cars.jpg", whose line of land, trees and a car pulses and gyrates, while you control a wiggling sun— making for the trippiest game experience I've had in some time. Once you've enjoyed the first few plays, you may want to make a symlink to a pathed directory so that you don't have to keep entering Tetuhi's source directory.

Although the games themselves are rather simplistic (and lame in most cases), it's the implications of the image manipulation that are of real interest here. I can see parts of the code foundation making it into much larger-scale projects in the gaming and multimedia area in the future. Tetuhi's creator, Douglas Bagnall, is making particular efforts so that Tetuhi can be included on the One Laptop Per Child XO laptop, so it'll be interesting to see what kind of games and drawings children around the

Projects at a Glance

joyevmouse
(welz.org.za/projects/joyevmouse)

joyevmouse is a joystick-to-mouse mapper that converts joystick events to mouse events. Of course, this means that lazy people like myself who watch endless episodes of anime and *Top Gear* won't have to get off the couch. Conveniently, joyevmouse also runs entirely in user space. It does not run as a kernel driver nor does it need a patch. Extra documentation and users are lacking at this point, so check it out and see if it suits your needs.

The Stump Window Manager
(Stumpwm, **www.nongnu.org/ stumpwm**)

Stumpwm is a keyboard-driven, minimalist X11 window manager written in Common Lisp. Despite its visually minimalist approach (there are no window decorations, icons or even buttons), Stumpwm is designed to be fully customizable and very powerful. And, judging by its main feature, I'd say it is so, because Stumpwm is designed to be hackable while the actual program is running. The ultimate control freak will love this, and any Lisp fans also should to take a gander.

world will come up with to play in connection with Tetuhi's game rules.

Check out some of Douglas' other crazy projects at **halo.gen.nz**.∎

John Knight is a 23-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.

Brewing something fresh, innovative or mind-bending? Send e-mail to knight.john.a@gmail.com.

Hot and Bothered at Starbucks

Reviewing the Cradlepoint PHS300 DAN SAWYER

Cruising for hotspots on a Linux Laptop can be a royal pain. It's not that we don't have good Wi-Fi support—we do—it's more that a lot of places offer free Wi-Fi with strings and dongles. My favorite coffee shop for working in, for example, offers free Wi-Fi to customers, and they control access by means of PCMCIA cards with the router set to allow only those MAC addresses. Leaving aside the fact that I have no way to install their Windows-only drivers, my laptop sports only an ExpressCard slot, so I'm pretty much screwed no matter what.

Of course, if I could find a way to get my paws on a device that gives me Wi-Fi wherever I go, this wouldn't be a problem. Imagine writing a shipwreck story on the beach where it takes place while still having access to all the glorious resources of the Internet to help make sure you have the details of tall ship rigging, local wildlife life cycles and edible plants at your fingertips. Or, if you're not a half-mad fiction writer, you still could use such a device to blog about a movie you're watching from the back row of the theater or about a protest from a park bench nearby and get the drop on other bloggers who will have to wait in line for a table at Starbucks.

I recently discovered, much to my delight, that such a device does exist. The Cradlepoint PHS300—PHS standing for Personal HotSpot—is a compact little router that, once turned on, establishes a solid wireless cloud suitable for use by anyone with a properly equipped laptop or other Wi-Fi-enabled device (**www.cradlepoint.com/phs300/ phs300.php**).

Inside the Box

Technically speaking, the PHS is a wireless router/firewall designed to work with 3G phones and EVDO devices. You plug said device in to the PHS's USB port and turn them both on, and (after a bit of tinkering) you have a wireless access point to the Internet.

Opening up the box, you'll find only the router itself, a small pamphlet, a battery and a power adapter. The package doesn't contain the extras that usually come

with USB devices or computer parts. There isn't, for example, a driver disc or a manual, nor is there a USB cable for connecting the PHS to your 3G phone. In both cases, you're on your own.

Don't worry though; the small pamphlet actually contains all the information you're going to need. It's not terribly well organized (for example, you don't find out what the default router password is until several steps after you're told you need it), but it gives you the leg up you're looking for.

The router itself is small and light—not much bigger or heavier than a

double-thick checkbook. It has three indicator lights: one tracks battery status, one lights up when a Wi-Fi cloud is established, and the final one indicates connectivity with the phone and/or EVDO modem when plugged in to the single USB port.

Setting It Up

From there, the rest of the setup falls like a string of dominoes. Once the unit is powered up, use Wi-Fi Radar or GNOME Network to grab an IP address and log in to 192.168.0.1 to configure the router. Configuration is quite self-explanatory—about the only difference between this and setting up a normal SOHO router is the screen for configuring login information for your ISP, should it be necessary. All the current encryption standards, from WEP through WPA2, are supported.

What They Did Right

The PHS300 is advertised as a universally compatible, secure, simple solution for emergency response, vacation broadband and mobile business. Both the box and the promotional materials give the impression that it's a product that "just works".

I'm pleased to say that it performs as advertised. It's not easy to imagine what they could have done better. The PHS300 is battery-powered, with about a two-hour battery life, and it recharges either over USB or via the power adapter. Because it operates like any other router appliance, it's not just useful for connecting to the Internet on the go. It also works well for setting up a proper network between your laptop and your colleague's, a little feature I've found useful recently while out on a film shoot. It handily supports full 802.11g speeds behind the Net gateway, and it has easy-to-administer traffic management to keep your cellular bandwidth usage well within the limits of your service plan.

I have only three gripes with this little marvel box, and two of them are pretty minor. The lack of an included USB cable is irritating—mostly because including such things is de rigueur in the current

marketplace. The other minor quibble has to do with the battery light—namely, there isn't one. In fact, there's no way to know how much battery life you have left until the power light flashes red, which loosely translates to "this router will commit suicide in two minutes unless you plug it in to something from which it can draw power". Still, in the grand

segment for diagnostic techs on a wired network. Alas, despite its otherwise brilliant potential as a WAP, the lack of one port pooches the deal, which is particularly disappointing since its little brother, the CTR350, has one. Paying more for less isn't exactly my idea of a good time; however, the PHS300 makes up for it with the firmware's bandwidth

(at this point) it's one of only two battery-powered travel routers on the market (the other being Cradlepoint's CTR350). If you have a use for one, it's worth picking up and supporting a company that's advertising is Linux compatibility—and living up to it!∎

Dan Sawyer is the founder of ArtisticWhispers Productions (www.artisticwhispers.com), a small audio/video studio in the San Francisco Bay Area. He has been an enthusiastic advocate for free and open-source software since the late 1990s, when he founded the Blenderwars filmmaking community (www.blenderwars.com). He currently is the host of "The Polyschizmatic Reprobates Hour", a cultural commentary podcast, and "Sculpting God", a science-fiction anthology podcast. Author contact information is available at www.jdsawyer.net.

The PHS300 is advertised as a universally compatible, secure, simple solution for emergency response, vacation broadband and mobile business.

scheme of things, it's a minor problem.

The major irritant is an oversight that keeps the PHS300 from completely knocking my socks off. The thing doesn't have an Ethernet port, and not having one limits its utility when in the presence of a wired uplink. It also makes it useless as an independent Net

management and load-balancing abilities, which maximize the speed you get over your 3G device. Used as a 3G router, its transfer speeds outperform both the CTR350 and using a 3G phone or modem directly from your laptop.

Despite this, it's an excellent little appliance, quite reasonably priced, and

Cellular broadband terms of service vary widely from carrier to carrier and plan to plan. Using the PHS may be against your carrier's terms of service—check your service contract to make sure you're in compliance.

The Neuros OSD Connects Your TV to the Internet

Play digital video, view photos and listen to audio from memory cards or hard disks, or browse YouTube and record TV shows in MP4 with this small Linux-based box. Oh, and you can hack it too—it's open source. MARCO FIORETTI

The Neuros OSD is a very small and energy-efficient box that can play digital video, photos and music from several sources, including the Internet, on any TV or home theater system. It also can do the opposite—convert analog video in real time from RCA or S-Video inputs to MP4 format, save it on memory cards, external USB drives or, thanks to its Ethernet port, remote computers. For a Linux/free-software fan, the OSD also is interesting because it runs customizable, Linux-based firmware (OSD stands for Open-Source Device).

Figure 1. Neuros OSD Box

Main Features

One of the main reasons for buying an OSD is backup and consolidation of video archives. It's one small box, much cheaper than a normal computer, and it's all you need to migrate tens or hundreds of VCR tapes or DVDs onto one hard drive—if you accept the unavoidable degradation that comes from recording from an analog output. The user interface also has specific settings to optimize recording from a PlayStation. Additionally, a timed recording function also makes the OSD into a bare-bones PVR.

Going the opposite direction, the OSD can play anything it finds on memory cards, USB drives or remote devices on its RCA or S-Video ports. Particularly interesting is the presence of a YouTube browser. Besides MP4, the list of supported formats available on the Neuros Web site includes

Figure 2. YouTube in the Living Room, without a Computer

several variants of .avi, .asf, .mov and others. The firmware upgrade procedure described later in this article can add even more features to the OSD.

What You Get with the OSD

When I opened the box, I found several accessories: remote control with batteries, two RCA cables, multivoltage power supply, serial cable and an infrared blaster for controlling your TV, cable box or satellite receiver through the OSD. I got the non-US set, which also includes two RCA-to-SCART adapters. As far as I can tell, that and the plug on the power supply are the only differences between the European and US kits.

The remote comes with very detailed instructions for controlling most TV sets. There also is a Learning Mode with which it can learn the main functions of your TV or VCR remote by directly "listening" while you use it.

Finally, the OSD has a plastic stand that holds it in a verti-cal position, which I didn't find particularly robust or useful. All the plug-and-forget cables are on one side: power, RCA in and out, S-Video input, IR blaster, serial and Ethernet inter-faces. The "user" ports—two for memory cards and one for USB—are on the opposite side. With this layout, the OSD is more stable, which makes it easier to fit on the shelves of ordinary home theatre furniture, as it is flat on the bottom with the user ports facing the room.

I have tested the Neuros OSD on a standard analog PAL TV with a 16:9 32" screen, a generic 1GB USB MP3 player

Figure 3. OSD Accessories

Figure 4. Cables, Adapters and Other Accessories Included in the European Kit

and a 128MB SD memory card from Dikom. For network-related tests, I connected it to a port of a D-Link 604T ADSL modem/router.

Firmware Upgrade

Because the first thing you see when you open the box is a big, red sheet of paper saying, "Please upgrade firmware immediately", that's what I did. The procedure is simple, requiring just a bit of attention. Depending on how old the firmware loaded in your own OSD is with respect to the latest upgrade, some steps I cover here may be different, and some upgrading methods may not apply.
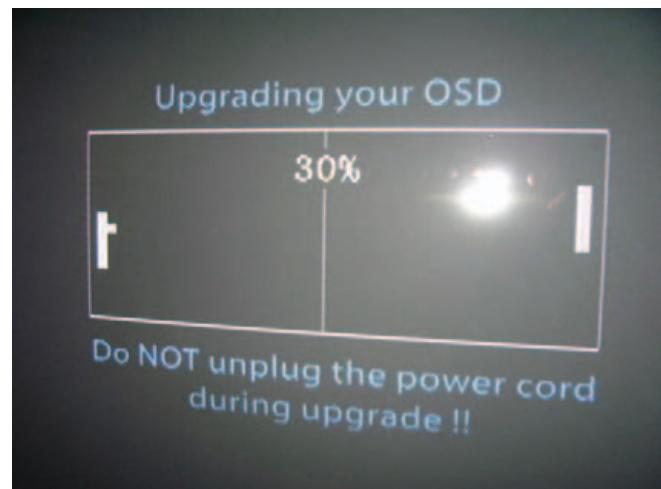
Figure 5. Waiting While the Firmware Upgrades, 1980s Style

First, hook up the OSD to your TV, and check which firmware version it currently is running by going to the Settings→Properties menu of the on-screen user interface. In my case, the version was 3.31-1.24. According to the Neuros Web site, this version wasn't new enough to upgrade directly from the Internet, so I had to download the latest one manually. In my case, this was an 11.7MB file called osd-3.33-1.75-02.849.upk.

Next, I copied that .upk file to a USB key, plugged it in to the OSD, selected the package from the file browser and ran the "Upgrade firmware" option. I chose a USB key because it was handy on my desk, even though the Web site warned that my current firmware may not be able to upgrade from such a device. Sure enough, when I tried it, the upgrade failed in less than one minute, with a "Sorry, package error" message.

The OSD, however, safely rebooted, so I got the memory card, plugged it in, copied the .upk file from the key to the card with the OSD file manager and re-issued the upgrade command. Everything went fine, and in about ten minutes I had the firmware that, among other things, can upgrade directly from the Internet or schedule automatic upgrades of stable or test versions at whatever frequency I choose.

Setting Up and Using the OSD

After upgrading, you'll see a "Thank You" screen that invites you to set up the OSD through a few interactive screens. The first one is for LAN configuration. Just as during a standard Linux installation, both static and DHCP configurations are possible. I tried them both without any problems. After configuring network and Wi-Fi, which you can skip altogether, you can configure the IR blaster.

Once everything was up and running, I finally started using the OSD. I've played MP3 files, recorded and played TV shows and YouTube clips and browsed digital pictures. After testing, I can say that the OSD works as advertised. Some parts of the user interface could be more efficient, but all in all, it is simple to use. A French language pack is already available, and Italian, Spanish, German, Dutch and Portuguese should follow soon.

The menu system is similar to standard living-room DVD players, with a sliding bar on the bottom that tells you how much free space there is on the internal memory or the external devices (memory cards or USB drives) you are using. You also can customize the graphic theme and screensaver.

There are three video recording modes. The first is called Quickstart, defined in the OSD manual as "take a leap of faith and simply press Record". The second is Standard mode, where you can change parameters, and finally, there's Advanced mode, which provides more flexibility and control but requires a bit more competence for proper use. Image and audio quality of TV recordings were almost indistinguishable from the originals, even with the default settings.

It's one small box, much cheaper than a normal computer, and it's all you need to migrate tens or hundreds of VCR tapes or DVDs onto one hard drive—if you accept the unavoidable degradation that comes from recording from an analog output.

The remote has standard, VCR-like keys to control video playback. Video recorded with the OSD takes about one hour per gigabyte at the highest quality. The maximum duration of a recording depends on the maximum file size supported by the host filesystem.

The YouTube browser is simple but effective. All the essential functions are grouped in Videos, Search, Favorites and Settings submenus. The Videos menu has buttons for listing all new clips or just the most-viewed ones for the time period you choose (day, week, month or year). Once you find the video you want, the OSD plays it full-screen. YouTube quality on standard TVs isn't great, but that's not the OSD's fault, and there are no glitches during playback if your Internet connection is fast enough.

The OSD photo viewer has full-screen, thumbnail and slideshow modes. In slideshow mode, you can configure the duration of each slide. The audio player has a playlist-creation functionality.

In my tests, all types of files (video, audio and pictures) were played with the same quality, without degradation or other problems, no matter whether they were on a memory card, USB drive or computer on the local network.

Hey, the Command Line!
The OSD is a nice and small box with serial port, Ethernet port, Linux inside and very little power consumption.

Besides a graphical interface built on the Qtopia toolkit, it has a LUA interpreter, a Telnet server and BusyBox. If this doesn't make a hacker want to mess with the OSD, nothing will. As a matter of fact, there already is a community customizing and extending the OSD in various ways or using it as a mini-server. To get inside the OSD, simply type `telnet` and the IP address, then log in as root with the default password, pablod. This drops you into a standard shell, within the limits of BusyBox.

In order to browse my PC partition from the TV, using the OSD remote, I simply typed:

```
mkdir /media/polaris
mount -t nfs 192.168.1.2:/mydata/osd_test /media/polaris
```

Note that, besides Telnet, you also can open an OSD console on your TV from the Advanced applications menu. The on-screen keyboard is much slower to use, but all the keys are there.
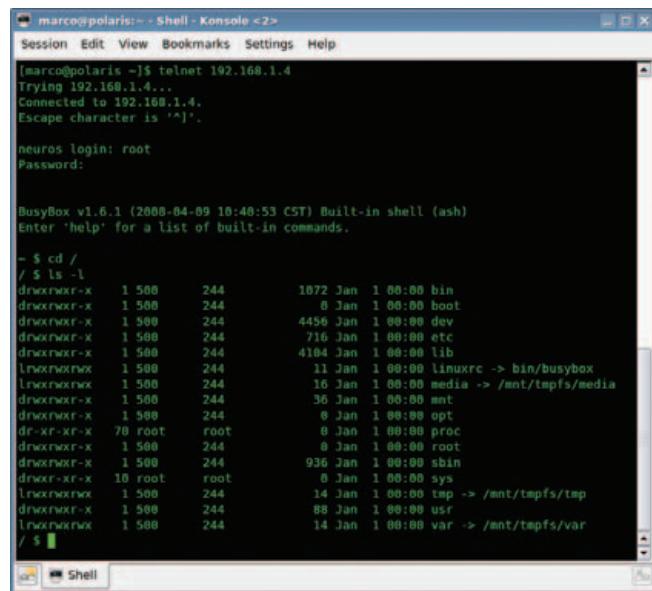
Figure 6. Looking into the OSD via Telnet

Advanced Features and Hacking
The Advanced applications menu also lists an MP4 Video editor (beta). When I tried it on an MP4 file on my computer, it wouldn't even open the NFS-mounted directory, which, as I already mentioned, was reachable without problems by the OSD file browser and picture viewer. Neuros confirmed to me that this application is still just an experiment, usable only on small clips stored in the cards or USB drives.

The list of features coming soon, some of which are Google Summer of Code projects, is really interesting. Besides Samba, Web and FTP servers, the latest announcements mention streaming via Fuse, a Last.fm client and an Ogg Theora codec. Currently, the software that actually plays movies and music and shows the menus, called osdmain, is not designed to communicate with other programs. However, work already is ongoing to overcome
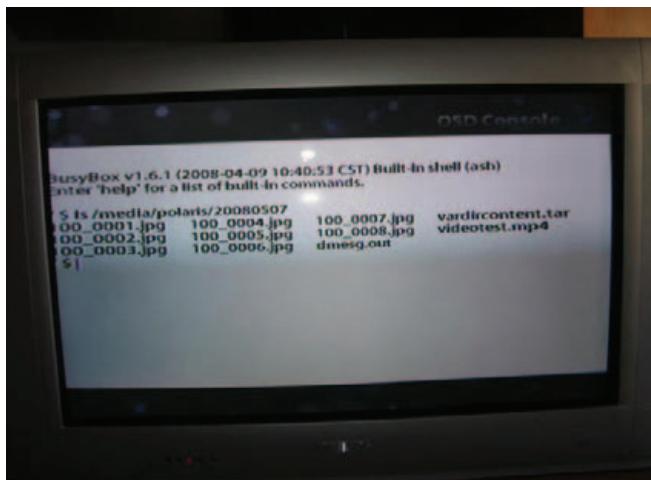
Figure 7. Browsing NFS Partitions from the TV Screen

this limitation and interact with the OSD over LAN. For more information, check out the Neuros Developer Web site (see Resources).

OSD in the Family and on the Road

Besides the main scenarios listed in the Neuros ads, I plan mostly to use the OSD in three other ways that are more interesting to me. First, the OSD makes it possible for kids to play YouTube clips, photos from their digital cameras or their MP3 playlists in the living room, without messing with dad's computer. Second, as the OSD is so small and light, I see it as a traveler's friend. Take it with you on vacations to view your digital photos right away on any motel TV or back them up to a USB drive, without carrying along a more expensive, fragile and bulkier laptop. Portability and the small size also mean I finally will be able to "steal" hours of VHS family movies whenever I visit relatives who often don't even own a computer. Saying, "Hi, Auntie, may I plug this tiny box in to your VCR and leave it there while we have dinner?" takes more time than actually doing it.

Missing Pieces and Problems

The USB interface supports only the 1.1 version of the standard. Neuros itself warns that recording to USB could cause frame drops due to speed bottlenecks. Adding a memory card adapter (which Neuros sells separately) to the kit would have made it more versatile. Also, the list of supported formats isn't 100% reliable. The firmware I tested, for example, can't handle the .mov videos generated by my Kodak camera. The audio played fine, but all I saw was a black screen.

The first thing I thought when I read the OSD datasheet was that the absence of digital inputs makes it impossible to copy DVDs or DV tapes without degradation. Neuros answered that the OSD is meant to offer flexibility and compatibility with the most common TV sets, at an affordable cost and with the simplest possible interface. They explained (and I agreed with them) that, in this context, adding digital output is not really necessary, especially because it wouldn't sensibly increase the final display quality. Digital input, instead, would have increased the cost enough to make the OSD really hard to sell.

Conclusion

All in all, I only had one real problem with the OSD, which I saved for last because it is (potentially) quite serious and also because it may well be solved by the time you read this.

As I mentioned before, the single functions work fine. The user interface, however, froze badly enough, in certain cases, to make the OSD unusable without doing a power cycle. To be more specific, this happened regularly when I had the Ethernet cable, the USB key and memory card all plugged in at the same time. The memory card alone also slowed the device, so part of the problem may be physical or formatting problems with the card itself. Even with other configurations, however, I noticed a recurring pattern. Heavy-load tasks, like playing or encoding video or audio, would go on without problem for hours, but using the remote too quickly or for more than a few minutes could slow down the OSD to a halt, especially when a storage device was plugged in.

By the looks of it, this is almost surely a bug in the particular firmware version that I tested, so don't judge the OSD by this problem, and check the Neuros Web site for updates. The Neuros OSD remains a handy and versatile device, although it's not exactly cheap. Taken one by one, all the features work well, and the device can be hacked and extended in many ways, so it could be a useful addition to your digital living room.∎

Marco Fioretti is a freelance writer and digital rights activist, author of the "Family Guide to Digital Freedom" (digifreedom.net) and member of several groups working on promoting wider adoption of Free as in Freedom formats and software.

At the time of this writing in May 2008, the OSD sells for $179 US at the Neuros on-line store. Outside the US, it will be available in some department stores in the UK and France (starting in June), with broader distribution in other countries starting later in the summer of 2008.

Resources

Supported Video Formats: **www.neurostechnology.com/neuros-osd-playback-settings**

User Guide: **wiki.neurostechnology.com/index.php/OSD_Guide**

Neuros Open-Source Page: **open.neurostechnology.com**

Developer IRC Channel: **open.neurostechnology.com/irc**

Developer Wiki: **wiki.neurostechnology.com**

Neuros On-line Store: **store.neurostechnology.com**

The BUG:
a Linux-Based
Hardware Mashup

It runs Linux, has a GPS, camera, motion detector and color touchscreen—and it's completely hackable!

Mike Diehl

Tinker Toys, Lincoln Logs, Erector Sets, Legos—I think I had just about every building toy there was when I was a child. Now that I'm all grown up, I still like to play with toys, and I still like to build things and connect them together. Only now, my toys are much more sophisticated, and some of them are even practical. I think much of the attraction that software development has for me is that I get to use my creativity to build applications that didn't previously exist, using a few software building blocks. I think most programmers and Linux users can relate to feeling this attraction.

However, many of the really neat things I would like to do in software aren't typically supported by hardware. For example, my home's thermostat doesn't talk to my groupware to see when I'll be home and want the house heated or cooled. My digital camera doesn't talk to my GPS to embed location information into the pictures I take, and I'm not able to add labels to my pictures with my PDA. To be able to build functionality like this, we need hardware that is open enough so we can hack on it and powerful enough that we can do nontrivial things with it. Finally, we need hardware that has a variety of functions built in to it. I have such a device; it's called a BUG from Bug Labs, and it's got to be the neatest thing I've seen in some time.
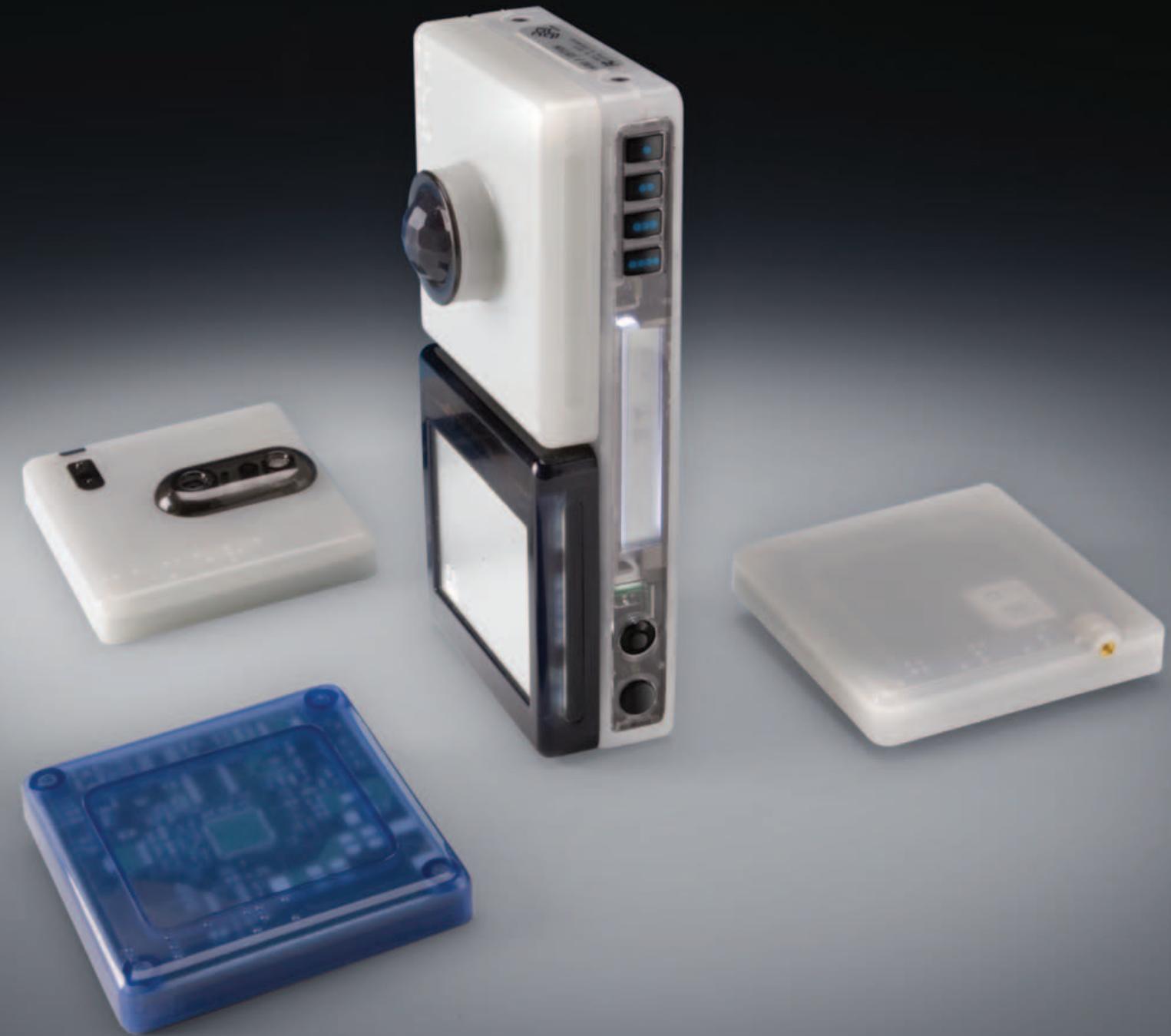
The BUG is an embedded Linux machine that accepts up to four external modules that provide various functionality. For example, the BUG I received had a color touchscreen module, a GPS module, a 2-mega-pixel digital camera module and an accelerometer with motion sensor. All of these modules plug in to the base unit. Once plugged in to the base unit, the modules expose their functionality as a kernel device and via a Java API. The idea is that you write a program that combines these functions into useful, or simply fun, applications. Peter Semmelhack, the CEO at Bug Labs, described it to me as a hardware mashup.

When I received my review unit, I opened the package in FedEx's parking lot and couldn't believe what I saw. The base unit is only 5-inches wide, 2.5-inches deep, and less than half an inch thick. There are two module ports on top of the unit and two ports on the bottom of the unit. With all four modules installed, the whole unit fits in the palm of your hand and is about the size of a large digital camera. The Web site indicates that the camera module can output still frames or MPEG video at ten frames per second. The unit comes with an LCD status display, four software-definable buttons, two menu buttons, a USBtoGo port and a piezo speaker—all of this and a tripod mount!

The unit also comes with a 512MB MMCmicro memory card installed. I found out the hard way that this is where the BUG stores its root filesystem. I decided to see what was on it, so I put the memory card in my PDA, which of course reformatted the card and squashed my BUG pretty handily. Fortunately, I was able to download a new image from the Bug Labs Web site, and I was up and running again in minutes. The root image consumes only about 30MB of the available 512MB, so there should be plenty of space for user programs, pictures and data.

Conspicuously absent from the unit is any type of labeling. None of the buttons are labeled, nor are any of the modules, although the modules do sport some Braille marking. This doesn't make the unit difficult to use, but it does make for a clean presentation. It also opens up the possibility for chassis modification, which reinforces the idea that the BUG puts the user in control.

I've shown my review unit to my wife and every other nerd I know, and the response has been the same each time. At first they don't know what it is. After I explain what it is and what it can do, they simply can't believe it. Typically, they leave saying, "that is just too cool!" And it is.

As I mentioned earlier, the BUG exposes all of its functionality via a Java API. I have to confess that I'm not a big fan of Java, but I understand that Java is a language many people already know and almost anyone can learn. Java also is an open standard, and Bug Labs, thankfully, is all about open standards, as I discuss later in this article.

Once connected and configured, the BUG integrates seamlessly with the Eclipse IDE. After following a few simple instructions, I was able to get Eclipse to recognize my BUG and all of the installed modules. Eclipse then presented me with a programming and hardware integration environment that even I could work with, and I'm not a Java programmer. There are lots of free source code examples available from the Bug Labs support site. I was able to download and install a calculator application, as well as a digital camera application within minutes. The example code is well written, and the API seems to be intuitive. The Bug Labs Web site has a lot of documentation for the API. I've never had a compelling reason to become proficient in Java—until now. A nerd like me could have a lot of fun with this device.

Bug Labs even provides a virtual BUG environment available from within Eclipse. The virtual environment allows you to plug modules in to a simulated BUG and run Java-based applications directly on the virtual device. The virtual BUG behaves almost exactly like a real BUG. Obviously, the GPS module, for example, provides bogus data, but it's still usable for software development and testing. You don't even have to own a BUG in order to develop software for it.

As I'm not a Java programmer and I don't use Eclipse, I was very interested in other ways to interact with the BUG. Getting it

connected to my network wasn't hard at all. The device's base unit doesn't have Ethernet or Wi-Fi capability; it connects to the network via USB. This meant that I had to upgrade the kernel on my workstation to enable USB networking, which presents itself as usb0 and acts just like any other network device. Note that, like most USB devices, the usb0 device won't be available until the BUG is connected and has finished booting up.

Once the BUG is booted, it runs the TWM window manager. Configuring my workstation to communicate with it was trivial, though the documentation on Bug Labs Web site made it a bit more complicated than necessary. The Web site indicated that you needed to install ifplugd, which I think is a neat program, but it's not needed in this case. All you have to do is configure the usb0 device with the right IP address and netmask. What I did was:

```
ifconfig usb0 10.10.10.1 netmask 255.255.255.0
```

The BUG has 10.10.10.10 as its IP address and expects to find its default gateway at 10.10.10.1. My workstation had to be configured to forward network traffic:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

After that, I was able to ssh into the device:

```
ssh root@10.10.10.10
```

Use root as the default password and change it to something more secure. You also should configure /etc/resolv.conf on the BUG so that DNS works properly.

Once you've logged in, you will be presented with a BusyBox prompt. You should feel free to take a look around. Much of what you see will be familiar to you. The fun begins when you start interacting with the application manager:

```
telnet localhost 8090
```

Typing help gives a list of commands that you can send to the application manager. To spare you the suspense, I'll tell you that you can use the install command to download an application from the Internet. For example:

```
install http://www.buglabs.net/application/download/43
```

This installs the BasicCalculator application, which is available from the Bug Labs Web site. By using the bundles command, you can determine which ID has been assigned to this application. In my case, the application was given the ID of 30. Then, you start the application using:

```
start 30
```

Shortly after issuing this command at the prompt, you will see a four-function calculator on the touchscreen—assuming you have the touchscreen module installed. I've found that using my fingers to interact with the touchscreen isn't extremely accurate. Once I dug up a stylus from one of my PDAs, I was able to use the BUG touchscreen with little or no effort.

Several applications are available from the Bug Labs Web

site. They tend to be well written and serve as good example programs from which to learn. There is sample code available on the Bug Labs Web site to exercise each of the available modules as well as the Java Abstract Window Toolkit (AWT) that comes with the BUG.

Bug Labs has a remarkable outlook when it comes to the openness of its products. To borrow a term from the CEO, Bug Labs embraces "Radical Openness". This policy is reflected in the use of Linux as the core of its system, Java as the main development language and complete documentation for the system and programming API. But, it goes beyond that. Bug Labs even has documented the pinout of the connector that its modules plug in to. I was told that if someone wanted to start producing third-party modules for the BUG, Bug Labs would support that effort. This policy, as well as the flexibility and sophistication of the device, makes the BUG a hacker's dream come true.

Every product has its drawbacks though, and the BUG is no different. The fact that none of the buttons and connectors are labeled makes the device less than intuitive. I actually had to look at the documentation that came with it. With such a small form factor, this device is just begging to be used in mobile applications. So although the USB networking gets the job done, the BUG really needs Ethernet or Wi-Fi capability. I'm told there will be an Ethernet module available soon.

Removing the various modules from the base unit is sometimes a bit unnerving. The latching mechanism holds the modules in place quite securely, and it's often difficult to un-install them. At first, it isn't even obvious how to go about it. Having removed and replaced the modules several times now, I've gotten used to the fact that I have to press harder than expected and that the unit's chassis is more sturdy than it looks. That said, I still haven't worked up the courage to exchange modules while the unit is running, although I'm told that they're hot-swappable.

The Bug Labs Web site is testing some of the modules that they plan to release in the second quarter of 2008. The QWERTY keyboard will be a welcome addition. Though the BUG has a built-in speaker, it's of rather poor quality, so the speaker module with I/O jacks will be nice. Neither of these promised modules seem to be in the same league as the modules already available. It's pretty hard to compete with a GPS module with an external antenna connection or a motion sensor module, but they're trying. I received an e-mail from my contact at Bug Labs indicating that they have about 80 new modules on their R&D list. Some of the modules on their list include a TV tuner, Servo interface, game controller, bar-code scanner, 3G modem and a Geiger counter!

My contact at Bug Labs went on to describe a module that they are working on that will open up the BUG to a whole new world of customization. They're about to release a module that exposes all of the BUG's hardware signaling and presents it in a manner much like a breadboard or breakout box. With such a module, it seems like it would be fairly easy to interface the BUG with a PIC microcontroller, or an external relay bank, or a Roomba—but I digress.

Remembering my earlier lamentations about the inadequacies of my existing electronic gadgets, what can we really do with a BUG? I have a few suggestions that are completely plausible and that I hope pique your interest in developing programs for the BUG.

Parents of teenagers might be interested in using a BUG to track their kids' driving habits. With a built-in GPS, an accelerometer and almost 512MB of memory, it wouldn't be difficult to track where kids go, and how fast they went. Such a device could be mounted in the trunk and would have the added benefit that if the kid decided to remove the unit and stash it at the library, where he told his parents he would be, the device could sense that it was being moved, using the motion detector module, and start filming the event—busted!

But, that's a bit too much Big Brother for me. I could see giving a BUG to a group of Boy Scouts at a camp-out. The device could be preprogrammed with GPS coordinates for various targets. The boys would be told to use the GPS to locate the targets and take a picture of them. The BUG could verify that the boys had reached the correct locations and store annotated pictures of each target. The accelerometer module could be used to measure the boys' minimum, average and maximum speed as they hiked up mountain passes and into valleys. This could evolve into a timed race between different groups.

I could go on, but I think you get the idea. Neither of these devices are on the market right now. Sure, you could use a GPS and a digital camera and get most of the same functionality described above, but part of the appeal of the BUG is that all of these features are combined in one unit and under user-programmable software control. With the appropriate modules installed, you can program the BUG to do anything you want it to do. Then, by installing different modules and running a different application, the same unit can provide an entirely different function—one of the few times when you are truly bound only by your imagination.

I don't have enough space to explore the BUG fully. I've spent hours looking at the Bug Labs Web site. I haven't written about the embedded Web server and associated Web services API. I've not written about the underlying Linux system. I've not written about the details of the SDK that are freely available from the Web site. I've not written about how the system hosts a service-oriented Java runtime component called OSGi that simplifies software development. For such a small device, there is a surprisingly steep learning curve. What originally attracted me to Linux was the fact that I could learn to do simple tasks quickly with Linux, but that I also could study Linux for years without ever running out of things to learn. I think the BUG is going to provide a very similar experience.∎

Mike Diehl works for Orion International at Sandia National Laboratories in Albuquerque, New Mexico, as a Linux server manager. Mike lives with his wife and three small boys, including a newborn, and can be reached via e-mail at mdiehl@diehlnet.com.

Resources

Bug Labs Web Site: **www.buglabs.net**

Download Site for BUG SDK: **buglabs.net/sdk**

BUG Wiki: **bugcommunity.com/wiki**

BUG Documentation: **bugcommunity.com/forums**

Getting Started: **buglabs.net/products**

Billix

A SYSADMIN'S SWISS ARMY KNIFE

Turn that spare USB stick into a sysadmin's dream with Billix.

BILL CHILDERS

Does anyone remember Linuxcare? Founded in 1998, Linuxcare was a company that provided support services for Linux users in corporate environments. I remember seeing Linuxcare at the first ever LinuxWorld conference in San Jose, and the thing I took away from that LinuxWorld was the Linuxcare Bootable Business Card (BBC). The BBC was a 50MB cut-down Linux distribution that fit on a business-card-size compact disc. I used that distribution to recover and repair quite a few machines, until the advent of Knoppix. I always loved the portability of that little CD though, and I missed it greatly until I stumbled across Damn Small Linux (DSL) one day.

After reading through the DSL Web site, I discovered that it was possible to run DSL off of a bootable USB key, and that old love for the Bootable Business Card was rekindled in a new way. It wasn't until I had a conversation with fellow sysadmin Kyle Rankin about the PXE boot environment he'd implemented, that I realized it might be possible to set up a USB key to do more than merely boot a recovery environment. Before long, I had added the CentOS and Ubuntu netinstalls to my little USB key. Not long after that, I was mentioning this in my favorite IRC channel, and one of the fellows in there suggested I put the code on SourceForge and call it Billix. I'd had a couple beers by then and thought it sounded like a great idea. In that instant, Billix was born.

Billix is an aggregation of many different tools that can be useful to system administrators, all compressed down to fit within a 256MB bootable USB thumbdrive. The 256MB size is not an arbitrary number; rather, it was chosen because

USB thumbdrives are very inexpensive at that size (many companies now give them away as advertising gimmicks). This allows me to have many Billix keys lying around, just waiting to be used. Because the keys are cheap or free, I don't feel bad about leaving one in a server for a day or two. If your USB drive is larger than 256MB, you still can use it for its designed purpose—storing files. Billix doesn't hamper normal use of the USB drive in any way. There also is an ISO distribution of Billix if you want to burn a CD of it, but I feel it's not nearly as convenient as having it on a USB key.

The current Billix distribution (0.21 at the time of this writing) includes the following tools:

- Damn Small Linux 4.2.5

- Ubuntu 8.04 LTS netinstall

- Ubuntu 7.10 netinstall

- Ubuntu 6.06 LTS netinstall

- Fedora 8 netinstall

- CentOS 5.1 netinstall

- CentOS 4.6 netinstall

- Debian Etch netinstall

- Debian Sarge netinstall

- Memtest86 memory-checking utility

- Ntpwd Windows password changing utility

- DBAN disk wiper utility

So, with one USB key, a system administrator can recover or repair a machine, install one of eight different Linux distributions, test the memory in a system, get into a Windows machine with a lost password or wipe the disks of a machine before repurpose or disposal. In order to install any of the netinstall-based Linux distributions, a
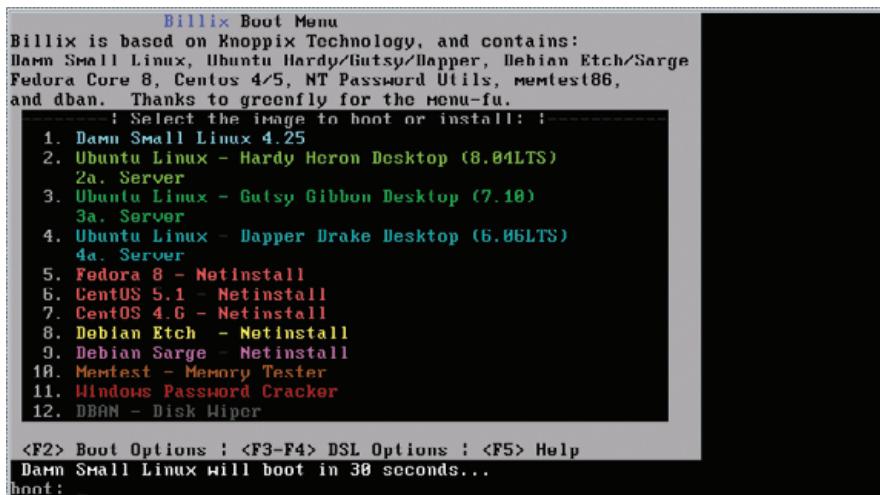
Figure 1. The Billix Boot Menu

working Internet connection with DHCP is required, as the netinstall downloads the installation bits for each distribution on the fly from Internet-based mirrors.

Hopefully, you're excited to check out Billix. You simply can download the ISO version and burn it to a CD to get started, but the full utility of Billix really shines when you install it on a USB disk. Before you install it on a USB disk, you need to meet the following prerequisites:

- 256MB or greater USB drive with FAT- or FAT32-based filesystem.

- Internet connection with DHCP (for netinstalls only, not required for DSL, Windows password removal or disk wiping with DBAN).

- install-mbr (part of the mbr package on Ubuntu or Debian, needed for some USB drives).

- syslinux (from the syslinux package on Ubuntu or Debian, required to create the bootsector on the USB drive).

- Your system must be capable of booting from USB devices (most have this ability if they're made after 2005).

To install the USB-based version of Billix, first check your drive. If that drive has the U3 Windows software on it, you may want to remove it to unlock all of the drive's capacity (see the Resources section for U3 removal utilities, which are typically Windows-based). Next, if your USB drive has data on it, *back up the*

data. I cannot stress this enough. You will be making adjustments to the partition table of the USB drive, so backing up any data that already is on the key is *critical*. Download the latest version of Billix from the Sourceforge.net project page to your computer. Once the download is complete, untar the contents of the tarball to the root directory of your USB drive.

Now that the contents of the tarball are on your USB drive, you need to install a Master Boot Record (MBR) on the drive and set a bootsector on the drive. The Master Boot Record needs to be set up on the USB drive first. Issue a `install-mbr -p1 <device>` (where `<device>` is your USB drive, such as /dev/sdb). *Warning:* make sure that you get the device of the USB drive correct,

or you run the risk of messing up the MBR on your system's boot device. The -p1 option tells install-mbr to set the first partition as active (that's the one that will contain the bootsector).

Next, the bootsector needs to be installed within the first partition. Run `syslinux -s <device/partition>` (where `<device/partition>` is the device and partition of the USB drive, such as /dev/sdb1). *Warning:* much like installing the MBR, installing the boot-sector can be a dangerous operation if you run it on the wrong device, so take care and double-check your command line before pressing the Enter key.

At this point, your USB drive can be unmounted safely, and you can test it out by booting from the USB drive. Once your system successfully boots from the USB drive, you should see a menu similar to the one shown in Figure 1. Simply choose the number for what you want to boot, run or install, and that distribution will spring into action. If you don't select a number, Damn Small Linux will boot automatically after 30 seconds.

Damn Small Linux is a miniature version of Knoppix (it actually has much of the automatic hardware-detection routines of Knoppix in it). As such, it makes an excellent rescue environment, or it can be used as a quick "trusted desktop" in the event you need to "borrow" a friend's computer to do something. I have used DSL in the past to commandeer a system temporarily at a cybercafé, so I could log in to work and fix a sick server. I've even used DSL to boot and

Troubleshooting Billix

A few things can go wrong when converting a USB key to run Billix (or any USB-based distribution). The most common issue is for the USB drive to fail to boot the system. This can be due to several things. Older systems often split USB disk support into USB-Floppy emulation and USB-HDD emulation. For Billix to work on these systems, USB-HDD needs to be enabled. If your drive came with the U3 Windows-based software vault, this typically needs to be disabled or removed prior to installing Billix.

If you're seeing "MBR123" or something similar in the upper-left corner, but the system is hanging, you have a misconfigured MBR. Try `install-mbr` again, and make sure to use the -p1 switch. You will need to run syslinux again after running install-mbr. If all else fails, you probably need to wipe the USB drive and begin again. Back up the data on the USB drive, then use fdisk to build a new partition table (make sure to set it as FAT or FAT32). Use mkfs.vfat (with the -F 32 switch if it's a FAT32 filesystem) to build a new blank filesystem, untar the tarball again, and run install-mbr and syslinux on the newly defined filesystem.

mount a corrupted Windows filesystem, and I was able to save some of the data. DSL is fairly full-featured for its size, and it comes with two window managers (JWM or Fluxbox). It can be configured to save its data back to the USB disk in a persistent fashion, so you always can be sure you have your critical files with you and that it's easily accessible.

All the Linux distribution installations have one thing in common: they are all network-based installs. Although this is a good thing for Billix, as they take up very little space (around 10MB for each distro), it can be a bad thing during installation as the installation time will vary with the speed of your Internet connection. There is one other upside to a network-based installation. In many cases, there is no need to update the newly installed operating system after installation, because the OS bits that are downloaded are typically up to date. Note that when using the Red Hat-based installers (CentOS 4.6, CentOS 5.1 and Fedora 8), the system may appear to hang during the download of a file called minstg2.img. The system probably isn't hanging; it's just downloading that file, which is fairly large (around 40MB), so it can take a while depending on the speed of the mirror and the speed of your connection. Take care not to specify the USB disk accidentally at the install target for the distribution you are attempting to install.

The memtest86 utility has been around for quite a few years, yet it's a key tool for a sysadmin when faced with a flaky computer. It does only one thing, but it does it very well: it tests the RAM of a system very thoroughly. Simply boot off the USB drive, select memtest from the menu, and press Enter, and memtest86 will load and begin testing the RAM of the system immediately. At this point, you can remove the USB drive from the computer. It's no longer needed as memtest86 is very small and loads completely into memory on startup.

The ntpwd Windows password "cracking" tool can be a controversial tool, but it is included in the Billix distribution because as a system administrator, I've been asked countless times to get into Windows systems (or accounts on Windows systems) where the password has been lost or forgotten. The ntpwd utility can be a bit daunting, as the UI is text-based and nearly nonexistent, but it does a good job of mount-

ing FAT32- or NTFS-based partitions, editing the SAM account database and saving those changes. Be sure to read all the messages that ntpwd displays, and take care to select the proper disk partition to edit. Also, take the program's advice and nullify a password rather than trying to change it from within the interface—zeroing the password works much more reliably.

DBAN (otherwise known as Darik's Boot and Nuke) is a very good "nuke it from orbit" hard disk wiper. It provides various levels of wipe, from a basic "overwrite the disk with zeros" to a full DoD-certified, multipass wipe. Like memtest86, DBAN is small and loads completely into memory, so you can boot the utility, remove the USB drive, start a wipe and move on to another system. I've used to this to wipe clean disks on systems before handing them over to a recycler or before selling a system.

In closing, Billix may not make you coffee in the morning or eradicate Windows from the face of the earth, but having a USB key in your pocket that offers you the functionality to do all of those tasks quickly and easily can make the lifeof a system administrator (or any Linux-oriented person) much easier.∎

Expanding Billix

It's relatively easy to expand Billix to support other Linux distributions, such as Knoppix or the Ubuntu live CDs. Copy the contents of the Billix USB tarball to a directory on your hard disk, and download the distro you want. Copy the necessary kernel and initrd to the directory where you put the contents of the USB tarball, taking care to rename any files if there are files in that directory with the same name. Copy any compressed filesystems that your new distro may use to the USB drive (for example, Knoppix has the KNOPPIX directory, and Puppy Linux uses PUP_XXX.SFS). Then, look at the boot configuration for that distro (it should be in isolinux.cfg). Take the necessary lines out of that file, and put them in the Billix syslinux.cfg file, changing filenames as necessary. Optionally, you can add a menu item to the boot.msg file. Finally, run `syslinux -s <device>`, and reboot your system to test out your newly expanded Billix.

I have a 2GB USB drive that has a "Super-Billix" installation that includes Knoppix and Ubuntu 8.04. An added bonus of having the entire Ubuntu live CD in your pocket is that, thanks to the speed of USB 2.0, you can install Ubuntu in less than ten minutes, which would be really useful at an installfest. There is good information on creating Ubuntu-bootable USB drives available at the Pendrive Linux Web site.

Alternatively, a really neat thing to do (but way beyond the scope of this article) is to convert Billix into a network-boot (via Pre-Execution Environment, or PXE) environment. I've actually got a VMware virtual machine running Billix as a PXE boot server.

Bill Childers is an IT Manager in Silicon Valley, where he lives with his wife and two children. He enjoys Linux far too much, and probably should get more sun from time to time. In his spare time, he does work with the Gilroy Garlic Festival, but he does not smell like garlic.

Resources

Billix Project Page:
sourceforge.net/projects/billix

Damn Small Linux:
www.damnsmalllinux.org

DBAN Project Page:
dban.sourceforge.net

Knoppix: **www.knoppix.net**

Pendrive Linux:
www.pendrivelinux.com

Syslinux:
syslinux.zytor.com/index.php

Pxelinux:
syslinux.zytor.com/pxe.php

U3 Removal Software:
www.u3.com/uninstall

Fun with E-Ink, X and Gumstix

How to get X running on a Gumstix embedded device with an E-Ink display and run all your favorite X11 applications.

Jaya Kumar

I'm excited by E-paper and the promise it holds. You've probably already heard about E-Ink's E-Paper Display (EPD) and seen it in recent E-book reader products. The E-Ink display media needs no power to hold an image, and it reflects light just like real paper. I've even seen recent products that make use of the physical flexibility of the E-Ink film in order to create "rollable" displays. But yes, this technology still has a few constraints. Current EPD displays are grayscale only and have measurable display latencies. However, I'm convinced that these limitations will disappear over time, as has happened with many other disruptive technologies.
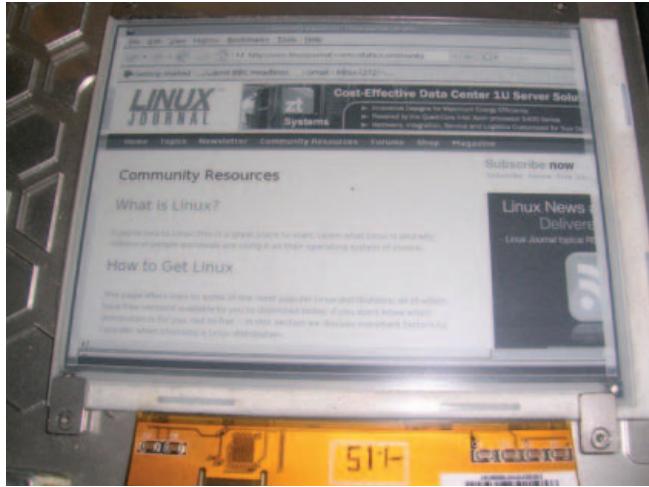
Figure 1. Firefox Displaying *Linux Journal* on an E-Ink Display

Figure 2. Running Dillo, xeyes, XClock and XLogo on an E-Ink Display

E-paper devices have been on the market since around 2006 or so. The good news is that Linux has become the de facto standard operating system for almost all of these devices. The two major products, Amazon Kindle and Sony PRS series, both utilize embedded Linux to achieve their functionality. Those products are great, but it also is fun to build your own device and use your own applications. That is where the Gumstix embedded device comes into play. E-Ink has a kit called the AM200 E-Paper Development Kit that provides all the hardware accessories you need to build your own E-paper device. Best of all, the kit is designed to work with Linux and is quite hack-friendly.

The AM200 kit serves to provide proof of concept for E-Ink. The software it provides helps you run a user-space application that lets you decode portable pixel map (PPM) images to the display. This is just to demonstrate the basic capabilities of the kit. It does not let you run normal X11 applications, such as xterm or xeyes. That's where this article comes in—it should help you add a set of building blocks that expand the system's functionality. These building blocks will enable the system to support a standard X server. Once you've gotten an X server on the system, you pretty much can run anything that's available from the realm of the penguin and the wildebeest, including your favorite Web browsers and PDF readers.

First, let's do a quick review of the hardware infrastructure we're using to better understand the software we need to add. The display shown in Figure 3 is an E-Ink Vizplex display that has a Thin-Film Transistor (TFT) back end. This is connected through the fine pitch (FPC) ribbon cable to the adapter board. This adapter board is there to distribute the correct signals to the display connectors and also to provide some expansion buttons that can be accessed using General-Purpose Input/Output (GPIO) from the Gumstix. This adapter board sources its signals from the Metronome controller board, which is the display interface. It carries the Metronome controller, which has an associated Linux framebuffer driver called metronomefb to which X will be talking. The Metronome controller board is connected via another FPC ribbon cable to the Gumstix board. The Gumstix board has the XScale PXA255 CPU and all the associated interfaces, such as Bluetooth, SD card, USB and others. The Lyre mainboard is the carrier for the Gumstix board. This Lyre mainboard has the power block, battery stuff,

Figure 3. Picture and Block Diagram of the Hardware: ❶ E-Ink Vizplex Display, ❷ Li Ion Battery, ❸ Display Adapter Board, ❹ Metronome Display Controller Board, ❺ Gumstix Board and ❻ Lyre Mainboard

status LEDs, USB network interface and USB serial interface that allows us to control the Gumstix from a standard PC.

Now, let's dig in to the software side of things. The Gumstix board has relatively strong Linux support. That said, the development environment for it is not yet perfect. But, these things are continuing to see rapid change, so be prepared to roll up your sleeves and hack a bit.

When your Gumstix board arrives, it could have one of two possible firmware configurations. The first is firmware built using a set of tools called buildroot. Buildroot is a collection of makefiles and scripts to help you set up a cross-compile environment for embedded systems. Buildroot will help you pull down everything you need to build a root filesystem image for your Gumstix board automatically. It was the de facto standard for Gumstix until about a year ago. The second possibility is a board with firmware built using the OpenEmbedded (OE) framework. OpenEmbedded is a new software framework for building embedded distributions. It is quite a bit more advanced than buildroot in terms of capabilities. It also is the currently preferred build environment for Gumstix. For the purpose of this project, either one should be good enough, and both are fairly easy to use. I cover the steps for both of them in this article.

The new software building block that we'll add to this system is something called deferred IO. Deferred IO is a recently added hack in the Linux kernel that allows nonmemory-mappable devices to pretend to be memory-mappable. It also allows us to hide the latency associated with the E-Ink display. This hack is what makes it possible to run X with various E-Ink controllers on Linux. We'll also add the Metronome and AM200 drivers that provide the Linux framebuffer interface for the Metronome controller together with the Gumstix board.

Okay, now we can do some real work. The first thing to do is establish a working environment on the Gumstix. Power on your Gumstix board and connect the two USB ports to your host machine. This lets you set up the USB-serial console and also creates a USB-net connection with which to transfer files.

The first is buildroot. I used the following incantation to set up a buildroot environment:

```
# mkdir gumstix_build
# cd gumstix_build
# svn co -r1441 http://svn.gumstix.com/gumstix-buildroot/
    ➥trunk gumstix-buildroot
# make defconfig
# make
```

This pulls down revision 1441 of the buildroot build. Select the appropriate option for your Gumstix board. Mine was a basix, so I selected the BR2_TARGET_GUMSTIX_BASIXCONNEX and GUMSTIX_400MHZ entries. This build revision worked fine

Once you've gotten an X server on the system, you pretty much can run anything that's available from the realm of the penguin and the wildebeest, including your favorite Web browsers and PDF readers.

You can use minicom or cu to use the Gumstix serial console. If your board is in working condition, you should see this when you connect:

```
Connected.

Welcome to the Gumstix Linux Distribution!

gumstix login:
```

The user name is root, and the password is gumstix. Log in to check whether your SD card works. If all is well, you should see the following:

```
gumstix login: root
Password:
Welcome to Gumstix!
# mount
/dev/root on / type jffs2 (rw)
proc on /proc type proc (rw)
/sys on /sys type sysfs (rw)
udev on /dev type ramfs (rw)
devpts on /dev/pts type devpts (rw)
/dev/mmcblk0p1 on /mnt/mmc type vfat
    ➥(rw,sync,fmask=0022,dmask=0022,codepage=cp437,iocharset=iso8859-1)
tmpfs on /tmp type tmpfs (rw)
```

The line with a mount entry called /mnt/mmc shows there is working SD card support.

Now that we've established a working environment, we can approach building the kernel. We want to use the 2.6.25 kernel, because it supports all the devices we are interested in and has the features we've talked about so far. To build this kernel, we need some things from our development environment—that is, the cross compiler that will allow us to compile binaries for the XScale CPU using a standard x86 host. Don't be frightened by this. It's not that hard. As mentioned before, there are two choices of tools to set up everything for you.

on my Ubuntu Edgy build machine, but you may need to experiment with the revision tag to find one that builds well on your chosen system. This build stage takes a while, as buildroot pulls down a wide set of sources and starts building everything from scratch. It will take at least several hours even on high-end machines. I typically leave this stage running overnight. Once it completes, key utilities will be added to your path. The ones that are necessary to our operation are:

```
./build_arm_nofpu/staging_dir/bin/arm-linux-gcc
./build_arm_nofpu/staging_dir/bin/mkimage
./build_arm_nofpu/staging_dir/bin/arm-linux-objcopy
```

That should settle things for buildroot users. If you are using OE, your incantation is slightly different:

```
# mkdir gumstix_build
# cd gumstix_build
# svn co https://gumstix.svn.sourceforge.net/svnroot/
    ➥gumstix/trunk gumstix-oe
# . ~/gumstix/gumstix-oe/extras/profile
# # edit your build config to select the
# # right machine (basix, connex, or verdex)
# bitbake gumstix-basic-image
```

Bitbake manages the dependencies and figures out what to build and how to build the cross-compilation environment. Suffice it to say that this stage is not the speediest to complete, so plan on taking an extended break before your tools are ready. On my build machine, the OE build takes about two days to complete. Please refer to the Gumstix OE Build Details link (see Resources) in order to set up your build config for your Gumstix board.

Now that you have a working environment, you need to pull down the mainline kernel tree that has the driver for the E-Ink Metronome controller and deferred IO support. I typically just pull down Linus Torvalds' latest tree. Once that is done, select and build using the right config file for your kit. On mine, the steps to do this are:

```
# git pull git://git.kernel.org/pub/scm/linux/
    ➥kernel/git/torvalds/linux-2.6.git
# cd linux-2.6
# cp arch/arm/configs/am200epdkit_defconfig .config
# make CROSS_COMPILE=arm-linux- ARCH=arm oldconfig
# make CROSS_COMPILE=arm-linux- ARCH=arm menuconfig
```

Then, select Device Drivers→Graphics support→Support for framebuffer devices, and make sure to turn on the module option for AM200 E-Ink EPD devkit support. Here's the next step, which builds some binaries:

```
# make CROSS_COMPILE=arm-linux- ARCH=arm
# arm-linux-objcopy -O binary -R .note -R .comment
   ➥-S arch/arm/boot/compressed/vmlinux linux.bin
# mkimage -A arm -O linux -T kernel -C none -a 0xa0008000
   ➥-e 0xa0008000 -n "uImage" -d linux.bin arch/arm/boot/uImage
```

This uImage file is what we will feed to the bootloader for our kernel. We also need to copy the modules that are used by the framebuffer drivers that we need to support X. We need to transfer these files to the SD card. The simplest way is to use an SD card reader and transfer it using your normal desktop mechanism. For example:

```
# cp arch/arm/boot/uImage /media/sd/
# cp drivers/video/metronomefb.ko drivers/video/am200epd.ko
   ➥drivers/video/sys*.ko drivers/video/fb_sys_fops.ko /media/sd/
```

Once that's done, you're ready to boot your new kernel on the Gumstix. To do this, you need to interrupt the normal boot process. On the Gumstix serial console, type reboot, and press any key to interrupt the normal boot process.

After that, you can use the following sequence of u-boot commands to let u-boot read the SD card, then retrieve and load the kernel into memory and finally boot from this memory:

```
# reboot
The system is going down NOW !!
Sending SIGTERM to all processes.
Please stand by while rebooting the system.
Restarting system.

U-Boot 1.1.4 (Nov  6 2006 - 11:20:03) - 400 MHz - 1161

*** Welcome to Gumstix ***

U-Boot code: A3F00000 -> A3F25DE4  BSS: -> A3F5AF00
RAM Configuration:
Bank #0: a0000000 64 MB
Flash: 16 MB
SMC91C1111-0
Can't overwrite "serial#"
Net:   SMC91C1111-0
Hit any key to stop autoboot:   0
GUM>
GUM> mmcinit
MMC found. Card description is:
Manufacturer ID = 464450
HW/FW Revision = c c
```

```
Product Name = Name 3070FDP?
Serial Number = 7a6a16
Month = 3
Year = 2007

GUM> fatload mmc 1 a2000000 uimage.bin
reading uimage.bin

1094024 bytes read

GUM> bootm
## Booting image at a2000000 ...
   Image Name:   uImage
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    1093960 Bytes =  1 MB
   Load Address: a0008000
   Entry Point:  a0008000
OK

Starting kernel ...

Uncompressing Linux.............
.......done, booting the kernel.

Welcome to the Gumstix Linux Distribution!

gumstix login: root
Password:
Welcome to Gumstix!
# uname -a
Linux gumstix 2.6.25gum-00000-ga052754 #16 PREEMPT
   ➥Wed Apr 30 22:54:35 EDT 2008 armv5tel unknown
```

Success! We now have booted a kernel with everything we need. The next step is adding Xfbdev. With OpenEmbedded, this step is fairly straightforward; simply execute the following command:

```
# bitbake xserver-kdrive
```

This generates an appropriate package (.ipk) file in the gumstix_build/tmp/deploy/ipk/armv5t/ directory. It should look something like xserver-kdrive-fbdev_1.4-r1_armv5te.ipk. Copy this file to the SD card, and use the following command to install it on your Gumstix:

```
# ipkg install /mnt/mmc/xserver-kdrive-fbdev_1.4-r1_armv5te.ipk
```

If you have buildroot, the process is somewhat similar, but you need to do some in-build munging. First, do the following:

```
# cd gumstix-buildroot
# make menuconfig
# # Go to Package Selection for Target-> then enable
# # the checkbox for the item labeled TinyX.
# # This is the other name for Xfbdev.
# make
```

This generates the Xfbdev and associated applications, such as xterm, XLogo, xeyes and others. You can copy these to

Figure 4. Xfbdev Running on Gumstix E-Ink

Figure 5. Ubuntu Desktop VNC-Viewed on a Gumstix E-Ink

your Gumstix's root filesystem. At this point, Xfbdev finally is on the system. We're almost at the finish line. On your Gumstix console, load the drivers we copied earlier:

```
# cd /mnt/mmc
# insmod drivers/video/syscopyarea.ko &&
   ➥insmod drivers/video/sysfillrect.ko &&
   ➥insmod drivers/video/sysimgblt.ko &&
   ➥insmod drivers/video/fb_sys_fops.ko
# insmod drivers/video/metronomefb.ko
# insmod drivers/video/am200epd.ko
fb1: Metronome frame buffer device, using 505K of video memory
```

The fb1 kernel message indicates that we now have a real framebuffer device to play with. Now, we need to point Xfbdev at the right device, so enter this:

```
# # make fb0 actually become fb1
# rm /dev/fb0
# mknod /dev/fb0 c 29 1
# /usr/X11R6/bin/Xfbdev -ac &
```

At this point, you should see the familiar crosshatch pattern of X. Yes, we finally did it! We now have an X11-enabled E-Ink display.

What to do next? Well, the Gumstix device has USB-net support, which means you now have a remote X11 display that you can connect to from any other machine on the network. You can point any X11 application at it—for example:

```
# # 10.0.0.2 is the E-Ink display
# # Slideshow
# for i in `ls *.jpg` ; do echo $i ;
   ➥ xloadimage -display 10.0.0.2:0 -global -rotate 90 $i ;
   ➥ done
# # PDF slideshow
# (( i = 1 )) ; while (( $i < 21 )) ; do echo $i ;
   ➥xpdf -remote eink /tmp/mybook.pdf $i -display 10.0.0.2:0 ;
   ➥sleep 3 ; (( i = $i + 1 )) ; done
# # Remote display an Ubuntu desktop on your E-Ink display
# DISPLAY=10.0.0.2 vncviewer ubuntu_box:1
```

The Future

A lot of work has been going on to enrich the Gumstix toolset and provide better integration with the AM200 kit. Projects like OpenEmbedded are simplifying embedded Linux work. I see a bright future for these displays (no pun intended). I don't think it will be much longer before we have displays that we can roll up just like paper and stuff in our back pockets. The costs will come down, and we'll be able to scatter them on a desk just like regular paper and treat them as extensions of our normal desktop display. Linux will continue to be at the forefront of this with its unique capabilities.

Acknowledgements

Jaya Kumar has been enjoying Linux since 1995 and is the author and maintainer of deferred IO, the fbdev drivers for E-Ink controllers. He is on a constant lookout for chocolate kulfi as well as cool new technologies to hack on and welcomes any and all feedback at jayakumar.lkml@gmail.com.

Resources

Gumstix OE Build Details: **www.gumstix.net/Software/ view/Getting-started/Setting-up-a-build-environment/ 111.html**

Gumstix Buildroot Setup: **docwiki.gumstix.org/Buildroot**

Using ipkg with Gumstix Feeds: **www.gumstix.net/Software/view/Getting-started/ Updating-and-adding-packages-via-ipkg/111.html**

E-Ink AM200 Prototype Kit: **www.e-ink.com/kits/ am200_index.html**

One Box.
SIXTEEN TRILLION
BYTES.

**Build your own
16-Terabyte file server
with hardware RAID.**

Eric Pearce

I recently had the need for a lot of disk space, and I decided to build a 16TB server on my own from off-the-shelf parts. This turned out to be a rewarding project, as it involved many interesting topics, including hardware RAID, XFS, SATA and system management issues involved with large filesystems.

Project Goals

I wanted to consolidate several Linux file servers that I use for disk-to-disk backups. These were all in the 3–4TB range and were constantly running out of space, requiring me either to adjust which systems were being backed up to which server or to reduce the number of previous backups that I could keep on hand. My overall goal for this project was to create a system with a large amount of cheap, fast and reliable disk space. This system would be the destination for a number of daily disk-to-disk backups from a mix of Solaris, Linux and Windows servers. I am familiar with Linux's software RAID and LVM2 features, but I specifically wanted hardware RAID, so the OS would be "unaware" of the RAID controller. These features certainly cost more than a software-based RAID system, and this article is not about creating the cheapest possible solution for a given amount of disk space.

The hardware RAID controller would make it as simple as possible for a non-Linux administrator to replace a failed disk. The RAID controller would send an e-mail message warning about a disk failure, and the administrator typically would respond by identifying the location of the failed disk and replacing it, all with no downtime and no Linux administration skills required. The entire disk replacement experience would be limited to the Web interface of the RAID controller card.

In reality, a hot spare disk would replace any failed disk automatically, but use of the RAID Web interface still would be required to designate any newly inserted disk as the replacement hot spare. For my company, I had specific concerns about the availability of Linux administration skills that justified the expense of hardware RAID.

Hardware Choices

For me, the above requirements meant using hot-swappable 1TB SATA drives with a fast RAID controller in a system with a decent CPU, adequate memory and redundant power supplies. The chassis had to be rack-mountable and easy to service. Noise was not a factor, as this system would be in a dedicated machine room with more than one hundred other servers.

I decided to build the system around the 3ware 9560 16-port RAID controller, which requires a motherboard that has a PCI Express slot with enough "lanes" (eight in this instance). Other than this, I did not care too much about the CPU choice or integrated motherboard features (other than Gigabit Ethernet). As I had decided on 16 disks, this choice pretty much dictated a 3U or larger chassis for front-mounted hot-swap disks. This also meant there was plenty of room for a full-height PCI card in the chassis.

I have built the vast majority of my rackmount servers (more than a hundred) using Supermicro hardware, so I am quite comfortable with its product line. In the past, I have always used Supermicro's "bare-bones" units, which had the motherboard, power supply, fans and chassis already integrated.

For this project, I could not find a prebuilt bare-bones model with the exact feature set I required. I was looking for a system that had lots of cheap disk capacity, but did not require lots of CPU power and memory capacity—most high-end configurations seemed to assume quad-core CPUs, lots of memory and SAS disks. The Supermicro SC836TQ-R800B chassis looked like a good fit to me, as it contained 16 SATA drives in a 3U

enclosure and had redundant power supplies (the B suffix indicates a black-colored front panel).

Next, I selected the X7DBE motherboard. This model would allow me to use a relatively inexpensive dual-core Xeon CPU and have eight slots available for memory. I could put in 8GB of RAM using cheap 1GB modules. I chose to use a single 1.6GHz Intel dual-core Xeon for the processor, as I didn't think I could justify the cost of multiple CPUs or top-of-the-line quad-core models for the file server role.

I double-checked the description of the Supermicro chassis to make sure that the CPU heat sink is included with the chassis. For the SC836TQ-R800B, the heat sink had to be ordered separately.

Figure 1. Front View of the Server Chassis

RAID Card Battery

I wanted the best possible RAID performance, which means using the "write-back" setting in the RAID controller, as opposed to "write-through". The advantage of write-back cache is that it should improve write performance by writing to RAM first and then to disk later, but the disadvantage is that data could be lost if the system crashes before the data was actually written to disk.

The battery backup unit (BBU) option for the 3ware 9560 RAID controllers protects this cached data from being lost by preserving it across reboots.

Ordering Process

I had no problems finding all the hardware using the various price-comparison Web sites, although I was unable to find a single vendor that had every component I needed in stock. Beware that the in-stock indications on those price-comparison Web sites are unreliable. I followed up with a phone call for the big-ticket items to make sure they actually were in stock before ordering on-line. Table 1 shows the details.

As you can see from Table 1, the hardware RAID components are about $1,000 of the total system cost.

Hardware Assembly

The chassis is pretty much pre-assembled. I had to insert some additional motherboard stand-offs and put on the rackmounting rails. I also snapped off some of the material on the plastic

Table 1. Parts List

QUANTITY	DESCRIPTION	SOURCE	PRICE PER UNIT	TOTAL PRICE
1	Intel Xeon 5110 Woodcrest 1.6GHz	Newegg	$211	$211
1	Supermicro MB X7DBE-O	Newegg	$426	$426
8	ATP AP28K72S8BHE6S 1GB RAM modules	ATP	$65	$520
1	Supermicro Chassis SC836TQ-R800B	Super Warehouse	$923	$923
1	3ware 9650SE-16ML	Newegg	$919	$919
1	3ware BBU-Module-04	The Nerds	$109	$109
1	Supermicro Heat Sink SNK-P0018	Wired Zone	$30	$30
16	Seagate ST31000340AS	Newegg	$274	$4,384
			Grand total:	$7,311

Figure 2. Inside View of the Server Chassis

cooling shroud to fit around the motherboard power cables.

The process of assembling the motherboard, CPU, heat sink, disks and memory was conventional, so I don't cover it here.

RAID Card Installation

Most of the 3ware 9650 controllers use "multi-lane" SATA cables with a single connector on the controller fanning out into four individual SATA cables. As this is a 16-port controller, four of the multi-lane cables connect to the SATA backplane. I made the process of connecting the SATA cables much easier by first removing the chassis cooling fans—they pop out quite easily. I also had to remove a couple of the disk backplane power connectors to access to the bottom-most SATA connector.

Be sure to connect the correct SATA cable to the correct SATA port, as a mistake here would be a disaster. You will need to determine the physical location of a disk with certainty
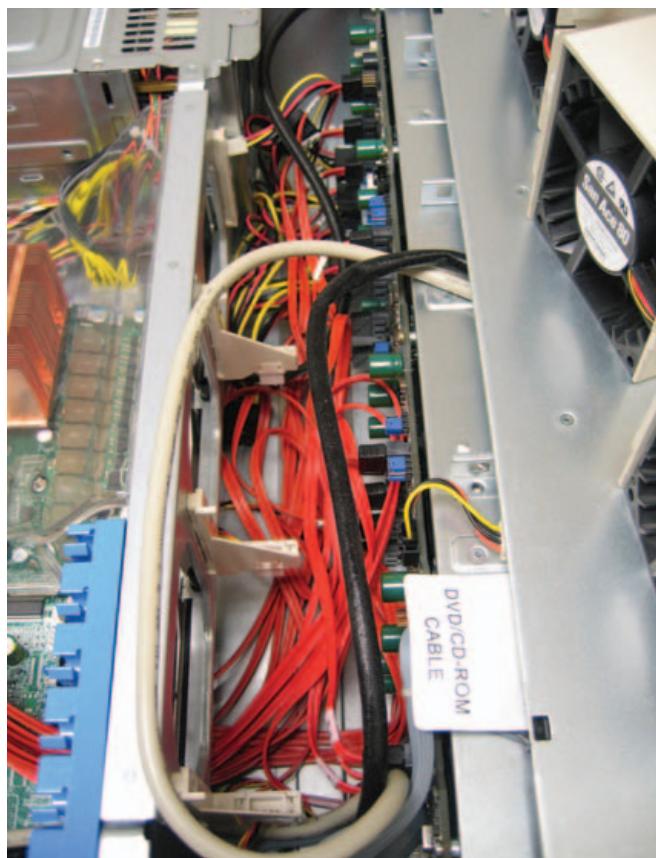
Figure 3. SATA Backplane with Cooling Fans Removed

when it comes time to replace one, or you will risk destroying the entire array. Familiarize yourself with the cable and disk numbering schemes before proceeding. For example, in the set of four multi-lane cables that came with my controller, one

cable was labeled with the first port at 0 (and ending at 3), and the other three had the first port at 1 (and ending at 4), while the backplane ports were numbered starting at 0 (and ending at 15), with the lowest numbered port at the bottom left (as viewed from the front). This seems to be a chassis-specific scheme, as other Supermicro chassis models number the ports from the top left down. SATA ports are numbered starting at 0 within the 3ware administrative interfaces. The 3ware administrative tools have a feature to "blink" a drive LED for locating a specific drive, but that is not supported in this particular Supermicro chassis.

The 3ware BBU typically is mounted on the controller card, but I have found that the controller starts complaining about battery temperature being too high unless there is generous airflow over the battery. I purchased the remote BBU option, which is a dummy PCI card that carries the battery and an extension cable that runs from the remote BBU to the main RAID controller card. I mounted the battery a couple PCI slots away from the RAID controller so it would be as cool as possible.
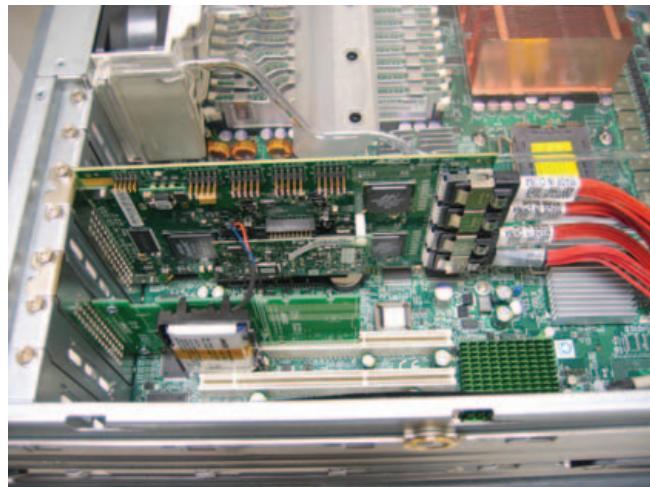
Figure 4. 3ware RAID Controller with Remote Battery Option

RAID Array Configuration

I planned to create a conventional set of partitions for the operating system and then a single giant partition for the storage of system backup files (named backup).

Some RAID vendors allow you to create an arbitrary number of virtual disks from a given physical array of disks. The 3ware interface allows only two virtual disks (or volumes) per physical array. When you are creating a physical array, you typically will end up with a single virtual disk using the entire capacity of the array (for example, creating a virtual disk from a RAID 1 array of two 1TB disks gives you a 1TB /dev/sda disk). If you want two virtual disks, specify a nonzero size for the boot partition. The first virtual disk will be created in the size you specify for the boot partition, and the second will be the physical array capacity minus the size of the boot partition (for example, using 1TB disks, specifying a 150GB boot partition yields a 150GB /dev/sda disk and an 850GB /dev/sdb disk).

You can perform the entire RAID configuration from the 3ware controller 3dm BIOS interface before the OS boots

(press Alt-3), or use the tw_cli command line or 3dm Web interface (or a combination of these) after the OS is running. For example, you could use the BIOS interface to set up just the minimal array you need for the OS installation and then use the 3dm Web interface to set up additional arrays and hot sparing after the OS is running.

For my 16-drive system, I decided to use 15 drives in a RAID 5 array. The remaining 16th drive is a hot spare. With this scheme, three disks would have to fail before any data was lost (the system could tolerate the loss of one array member disk and then the loss of the hot spare that would replace it). I used the 3ware BIOS interface to create a 100GB boot partition, which gave me a virtual sda disk and a virtual sdb disk of about 12.64TB.

I found the system to be very slow during the RAID array initialization. I did not record the time, but the initialization of the RAID 5 array seemed to take at least a day, maybe longer. I suggest starting the process and working on something else until it finishes, or you will be frustrated with the poor interactive performance.

OS Installation

I knew I had to use something other than ext3 for the giant data partition, and XFS looked like the best solution according the information I could find on the Web. Most of my Linux

experience involves Red Hat's Linux Enterprise distribution, but I had trouble finding information on adding XFS support. I specifically wanted to avoid anything difficult or complicated to reproduce. CentOS seemed like the best OS choice, as it leveraged my Red Hat experience and had a trivial process for adding XFS support.

For the project system, I installed the OS using Kickstart. I created a kickstart file that automatically created a 6GB /, 150MB /boot and a 64GB swap partition on the /dev/sda virtual disk using a conventional msdos disk label and ext3 filesystems. (I typically would allocate less swap than this, but I've found through experience that the xfs_check utility required something like 26GB of memory to function— anything less and it would die with "out of memory" errors). The Kickstart installation ignored the /dev/sdb disk for the time being. I could have automated all the disk partitioning and XFS configuration via Kickstart, but I specifically wanted to play around with the creation of the large partition manually. Once the Kickstart OS install was finished, I manually added XFS support with the following yum commands:

```
yum install kmod-xfs xfs-utils
```

At this time, I downloaded and installed the 3ware tw_cli command line and 3dm Web interface package from the 3ware Web site. I used the 3dm Web interface to create the hot spare.

Next, I used parted to create a gpt-labeled disk with a single XFS filesystem on the 14TB virtual disk /dev/sdb. Another argument for using something other than ext3 is filesystem creation time. For example, when I first experimented with a 3TB test partition under both ext3 and XFS, an mkfs took 3.5 hours under ext3 and less than 15 seconds for XFS. The XFS mkfs operation was extremely fast, even with the RAID array initialization in progress.

I used the following commands to set up the large partition named /backup for storing the disk-to-disk backups:

```
# parted /dev/sdb
(parted) mklabel gpt
(parted) mkpartfs primary xfs 0% 100%
(parted) print

Model: AMCC 9650SE-16M DISK (scsi)
Disk /dev/sdb: 13.9TB
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start    End     Size    File system  Name    Flags
1       17.4kB   13.9TB  13.9TB  xfs                  primary
(parted)   quit

# mkfs.xfs /dev/sdb1
# mount /dev/sdb1 /backup
```

Next, I made the mount permanent by adding it to /etc/fstab.

I now considered the system to be pretty much functional, and the rest of the configuration effort was specifically related to the system's role as a disk-to-disk backup server.

Performance

I know I could have used SAS drives with an SAS controller for better performance, but SAS disks are not yet available in the capacities offered by SATA, and they would have been much more expensive for less disk space.

For this project, I settled on a 16-drive system with a 16-port RAID controller. I did find a Supermicro 24-drive chassis (SC486) and a 3ware 24-port RAID controller (9650SE-24M8) that should work together. It would be interesting to see whether there is any performance down-side to such a large system, but this would be overkill for my needs at the moment.

There are still plenty of options and choices with the existing configuration that may yield better performance than the default settings. I did not pursue all of these, as I needed to get this particular machine into production quickly. I would be interested in exploring performance improvements in the future, especially if the system was going to be used interactively by humans (and not just for automated backups late at night).

Possible areas for performance tuning include the following:

1) RAID schemes: I could have used a different scheme for better performance, but I felt RAID 5 was sufficient for my needs. I think RAID 6 also would have worked, and I would have ended up with the same amount of disk space (assuming two parity drives and no hot spare), but my understanding is that it would be slower than RAID 5.

2) ext3/XFS filesystem creation and mount options: I had a hard time finding any authoritative or definitive information on how to make XFS as fast as possible for a given situation. In my case, this was a relatively small number of large (multi-gigabyte) files. The mount and mkfs options that I used came from examples I found on various discussion groups, but I did not try to verify their performance claims. For example, some articles said that the mount options of noatime, nodiratime and osyncisdsync would improve performance. 3ware has a whitepaper covering optimizing XFS and 2.6 kernels with an older RAID controller model, but I have not tried those suggestions on the controller I used.

3) Drive jumpers: one surprise (for me at least) was finding that the Seagate drives come from the factory with the 1.5Gbps rate-limit jumper installed. As far as I can tell, the drive documentation does not say that this is the factory default setting, only that it "can be used". Removing this jumper enables the drive to run at 3.0Gbps with controllers that support this speed (such as the 3ware 9560 used for this project). I was able to confirm the speed setting by using the 3ware 3dm Web interface (Information→Drive), but when I tried using tw_cli to view the same information, it did not display the speed currently in use:

```
# tw_cli /c0/p0 show lspeed
/c0/p0 Link Speed Supported = 1.5 Gbps and 3.0 Gbps
/c0/p0 Link Speed = unknown
```

The rate-limiting jumper is tiny and recessed into the back of the drive. I ended up either destroying or losing most of the jumpers in the process of prying them off the pins (before buying an extremely long and fine-tipped pair of needle-nose pliers for this task).

4) RAID card settings: Native Command Queuing (NCQ) is supposed to offer better performance by letting the drive electronics reorder commands for optimized disk access. I have found that NCQ is not always enabled by default on the 3ware controllers. It can be turned on manually using the queuing check box in the Controller Settings page of 3dm or via tw_cli:

```
# tw_cli /c0/u0 set qpolicy = on
```

The current setting can be verified on a per-drive basis via 3dm or by using tw_cli:

```
# tw_cli /c0/p5 show ncq
/c0/p5 NCQ Supported = Yes
/c0/p5 NCQ Enabled = Yes
```

5) Linux kernel settings: 3ware's knowledge base has articles that mention several kernel settings that are supposed to improve performance over the defaults, but I have not tried any of those myself.

6) Operational issues: despite all 16 disks being the same type and firmware version, some of them failed to display their model number properly in the various 3ware interfaces. For example, most of the disk model numbers are displayed correctly—for example, ST31000340AS. But, several show "ST3 INVALID PFM" in the model field. You can see this in the tw_cli interface. For instance, port 4 displays the model number properly, but port 5 does not:

```
# tw_cli /c0/p4 show model
/c0/p4 Model = ST31000340AS
```

```
# tw_cli /c0/p5 show model
/c0/p5 Model = ST3_INVALID_PFM
```

This situation would be intolerable in a system with a mix of drive types, as it would be difficult to determine which drive type was plugged in to which port. I was able to determine that the problem was with the drive firmware version and upgraded all the drives that exhibited this behavior.

As the system already was in active use before I determined that the firmware was the issue, I needed a way to upgrade each drive while keeping the system running. I could not simply upgrade the drives while they were part of an active disk array, as Seagate claims the upgrade could destroy data. I used the 3ware interface to remove the problem drive, which then forced the hot spare to replace it. The RAID controller automatically started to rebuild the RAID 5 array using the hot spare. I then physically removed the drive from the chassis and upgraded the drive firmware using another computer. After the upgrade, I re-inserted the drive and designated it as the new hot spare. The array rebuild operation took something like six hours to complete, and as I could remove and upgrade only one drive at a time, I was limited to one drive upgrade a day.

Conclusion

I have been using this system in production for several months and have consumed only a fraction of the available space:

```
# df -t xfs
Filesystem     1K-blocks       Used     Available  Use% Mounted on
/dev/sdb1      13566738304 2245371020 11321367284  17% /backup
```

I am quite happy with the result, as I have plenty of room to add more systems to the backup schedule, and I am confident I will not lose any backups due to hardware failures.■

Eric Pearce is the IT Lead for AmberPoint, Inc, an SOA governance software company based in Oakland, California. He has authored several books on UNIX and Windows system administration for O'Reilly & Associates.

TECH TIP Capture and Play Back Your Session

The script command is used to log an entire session. Type the command `script` at the command prompt, and script then copies everything you type and its response to the file typescript. Script starts a sub-shell; when you want to stop saving the session, end the sub-shell (normally with Ctrl-D or by typing `exit`).

A very useful feature of the script command is that it can output timing information to a separate file. The script and the timing information then can be used to replay the script.

The following example creates a script and timing data (timing data is always written to standard error):

```
$ script -t 2> timinginfo
Script started, file is typescript
$ ls
Desktop     test      scripts     redbooks
$ pwd
/home/jagadish
$ hostname
homepc
```

```
$ exit
exit
Script done, file is typescript
```

The entire terminal session then can be replayed later (with exact timing) using the `scriptreplay` command:

```
$ scriptreplay timinginfo
$ ls
Desktop     test      scripts     redbooks
$ pwd
/home/jagadish
$ hostname
homepc
$ exit
exit
```

Script is a useful tool for training and educational purposes.

—JAGADISH KAVUTURU

Linux for the Long Haul

Linux proves its worth more and more as you use it. MICHAEL SURRAN

Five years ago, I made one of the greatest life-changing decisions in my career—I switched my organization to the GNU/Linux operating system and supporting applications. It's not uncommon to read about businesses, schools and other organizations making this switch; however, what happens afterward? How do users adjust, and what about this total cost of ownership (TCO) we always hear of? Is Linux really ready for the desktop? Was it worth it to make the switch?

In 2002, Greater Houlton Christian Academy (GHCA) adopted Linux; you can read the details as to why and how in the February 2003 issue of *Linux Journal*. It's not an exaggeration when I describe this as a life-changing decision, not just for me, but for the school as well. I used to be a die-hard, Microsoft fanboy; now I use open-source software almost exclusively. Our school, which once had a mish-mash of dilapidated, old, donated computers that barely worked, now is recognized as being a leader in our region because of our computer technology—all of this from that fateful decision back in 2002.

Five years after the article was published, I find myself reflective, pondering where we've been and wondering what the future holds. Did I make the right decision? Would I do it again? There's much to consider in order to answer all these questions. Because that decision initially was based on financial need, let's first look at TCO.

Redmond Weighs In

Sometime after we adopted Linux, Redmond released a study claiming that the TCO for Linux actually was higher than for Microsoft Windows—even though Linux can be obtained for free. Microsoft has been pushing this idea ever since with its "Get the Facts" campaign. Had such a study existed in 2002, I might have wavered on making

Figure 1. Our third-grade students have no trouble using Linux as part of their lessons.

the switch. After all, price was the driving factor for us to use Linux in the first place. In some ways, that initial decision was a desperate decision. Since then, I've had time to consider TCO. So, was Redmond right?

The initial switch saved us money, because it allowed us to put what funding we had directly into hardware while avoiding the Microsoft tax (pre-installed Windows on computers). In fact, we could not have upgraded our computers if we had to purchase proprietary software as well. That's not to say there aren't some hidden costs in having the IT staff install software on bare-bones hardware, but for us, the savings far outweighed any extra labor costs. What is more important, however, is how using Linux and open-source applications continues to save us money today.

But, before discussing this continued savings, I need to stress that software evolves. Applications improve, bugs and

security holes are patched (hopefully), and new technologies emerge. With proprietary software, it can be years between major releases, and upgrading to that new release costs money. With open source, applications are improved all the time. After making the initial switch to Linux, one needs to consider how to keep up with the latest patches, upgrades and releases.

Being a tweaker who loves to squeeze every bit of efficiency from my computers, I was attracted to a distribution called Gentoo. Not only did it allow me to optimize Linux and thousands of applications for our computers, but also I found the package management system far superior to other distributions I had played with. It also forced me to learn the under-the-hood details about the Linux kernel, the GNU programs and many other OS management techniques that have helped me as a Linux administrator.

Speed Gives Life

Now, pay close attention, not only has Linux dramatically increased in usability and features during the last five years, but on the same hardware, it also has increased in speed. In other words, an upgrade really feels like an upgrade! In retrospect, try this with Windows. Our current base of computer hardware, which was modern in 2002, would not even run Vista, let alone run it faster than XP. However, our latest Linux upgrade is noticeably faster than the Linux we ran a few years back. In fact, our 2002 computers that average 256MB of RAM feel faster and more responsive than today's typical computers running Windows XP or Vista, and we have the latest in open-source software installed.

So, let's finish our TCO analysis. Not only did switching to Linux save us money in the initial switch, but also, every time I perform a system upgrade by typing `emerge -vauKD world` (it's that easy), we're saving money. We don't have to pay a company for every upgrade of every application for every seat. More important, I'm not forced to throw away good hardware and purchase new equipment in order to implement my software upgrade cycle. If we were running a "Microsoft shop", I'd have to retire almost every computer in our school and purchase all new equipment in order to upgrade to Vista. Now that's an expensive upgrade.

Although money is a big deal to a private school, there obviously is more to consider when switching an organization to a different operating system. A major consideration of mine was the "free as in freedom" roots of the Free Software movement. As the school's system administrator and the guy who has to make it all work, I have enjoyed this freedom during the past five years. I've taken advantage of being able to access and modify the source code. Many of my administrative duties have been simplified by customizing Linux for our school setting. Whether it is writing my own bootscripts or even creating my own software, I've been able to tailor our computer network in ways that I just could not easily or even legally do with proprietary software.

Figure 2. Even our boss, headmaster Mark Jago, uses Linux for his daily work.

Windows Genuine Disadvantage

There also is a freedom from worry. I don't need to concern myself with Windows Genuine Advantage, product activation and per-seat licensing. With Linux, you don't need to worry about how many processors your servers use or how many cores your next desktop computers will have. You don't need to consider special license restrictions for virtualization. You don't have to endure audits from the Business Software Alliance. As our band teacher loves to say, "No worries!"

Freedom extends outside the four walls of our school as well. For example, although OpenOffice.org can read and write Microsoft Word documents, the real advantage is that I can provide a copy of this software freely to any teacher or student, especially if that person can't afford to buy Microsoft Office. Anything we do in the classroom, students can do at home using their own copy of the free software we use. This gives us a tremendous advantage as an educational institution.

There's something else I consider when thinking about freedom—the freedom to access my data. I personally don't mind the existence of proprietary software in the world, but I strongly oppose proprietary standards and protocols that lock users from their own data. I want our documents, whether they be school records or a student's homework, to be accessible via an open and well-documented format. A recent experience in trying to access my own data stuck in a locked, proprietary format has made me appreciate all the more the true strength of open software and standards—freedom!

Five years is a long time to consider the wisdom of a decision. As the school's system administrator, I shoulder the burden of maintaining our computers, our network and our servers. What has it been like administering Linux since the switch? I'll be honest. There have been times when I've spent days trying to get something working right in Linux. However, I still use Windows enough to know that administering a Windows network isn't all cake and ice cream either. My experience with Linux is that once a setup is working, it stays working. Sometimes the initial setup takes longer, but once everything is configured right, it just works and works well. With distributions like Ubuntu, even that initial setup is becoming easier.
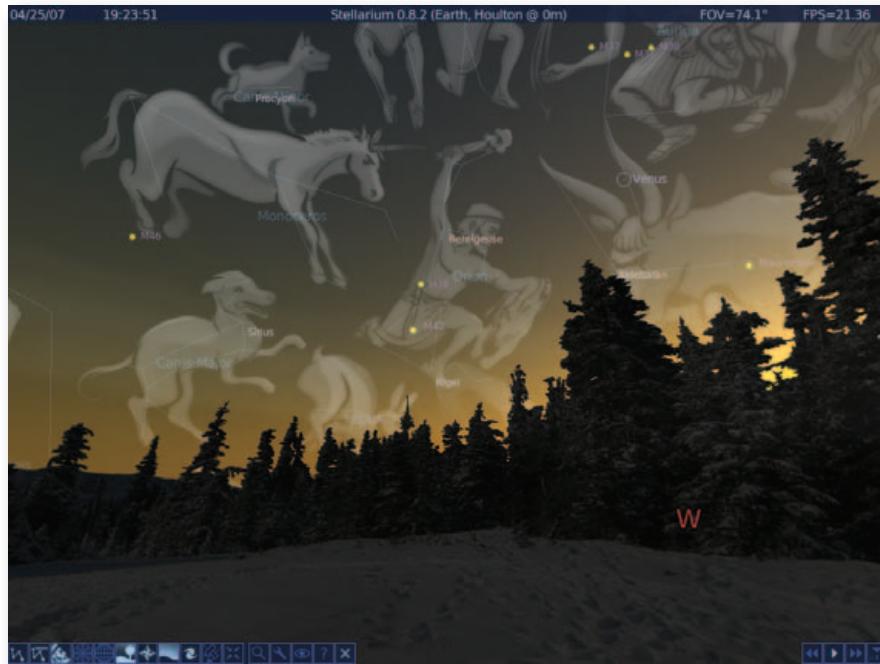
Figure 3. Stellarium is an example of the quality programs available in open source.

The Real Customers

Now, let's talk about the users of our Linux desktops. I'm a teacher as well, so I have to use Linux in the same way our teachers and students use it. That said, I'm a geek, and sometimes we geeks need to see the world through the eyes of a typical user. Personally, I love using Linux! I'm using it right now to type this article, and never do I think, "Oh, how I miss Microsoft Word." Never!

In fact, it's when I'm in a Windows environment that I find myself missing this feature or that feature. This is why the argument that says Linux is playing catchup with Windows is so flawed. Sure, Linux uses a mouse and icons and menus exactly like Windows does, but what else would we use? Is a hybrid car not innovative just because it uses a steering wheel like every other car? I say, "hogwash!" Many features found in open-source software are innovative, many of which only recently, if at all, have found their way into Windows. For example, I love my multiple desktops, and my productivity suffers without them. I love tabbed browsing and have used it for years. I love KDE, but even more important, I love how the desktop environment is not welded to the operating system. Users can chose KDE or GNOME or IceWM or have no GUI at all (great for servers and robots). I love, love, love the power of the Bourne-Again shell (Bash). I could spend the entire article sharing wonderful features that are unique to Linux. However, let's get back on track.

My experience has been that average adult computer users don't understand or even care about the power of multiple desktops, scriptable shells and so forth. For them, using a computer is a means to an end. They have a job to do, and the less the computer gets in the way, the better. The challenge comes when adults are faced with the unfamiliar. I stress adults here, because working with children and teenagers has been a totally different experience. Second-graders come into the lab and, with ease, use Linux to perform any task they would in Windows or Mac OS X. Teenagers line up and ask me to burn them Linux CDs for their home computers. However, most of you reading this probably deal with adults, and we adults are often old dogs.

They say that you can't teach an old dog new tricks. I don't agree with that, but sometimes old dogs do growl and fuss and even bite when forced to learn those new tricks. This can be especially true if the users aren't very computer-savvy to begin with. This means they are relying on icons, menus and options being at specific places and doing specific things. For this reason, many open-source programs try to replicate the feel of software with which the majority of adults are familiar. This is understandable, and it makes the transition easier than you might think. Although I had a few instances of resistance when we first switched to open-source software, most of the staff adapted quite well. Training is needed, but that mechanism already should be in place, regardless of what software an organization uses. Software and user interfaces change over time, and users find themselves adapting, regardless of whether the switch is to Linux or the latest version of Windows. Although adults often resist change, they *can* change. Actually, after a little time, they become comfortable with the change and may even be glad for the change. I know many average computer users who now sing the praises of OpenOffice.org Writer, for example.

It has probably become apparent that during these last five years, I've become an advocate for Linux and open-source software in general. However, it would be dishonest of me to sing praises only without revealing the pitfalls I've encountered over the years.

The Downside

As the system administrator, a real thorn in my side has been hardware compatibility. I've had little problem installing Linux on a variety of computers, but peripherals such as printers, scanners and Webcams can be a serious pain in the neck. Too many hours have been wasted trying to get unsupported hardware to work. However, the lesson here is to buy only from vendors who support Linux with drivers and/or detailed specifications. As more organizations adopt Linux, vendors either will have to support Linux or lose their business
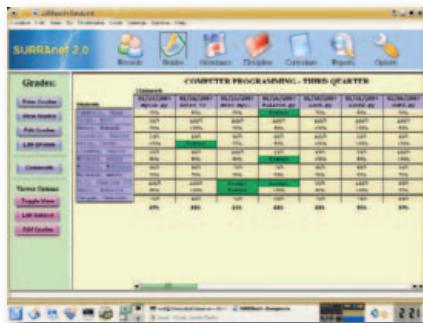
Figure 4. When I couldn't find an open-source program that met our needs, I wrote my own.

to those who do.

Something I find as irritating as the giant Maine mosquito is the use of proprietary protocols, standards and codecs that exclude Linux users from certain parts of the Internet. The Internet was built on open protocols, and it probably wouldn't exist in any meaningful way today if it had been locked up with proprietary standards owned by individual companies. Yet, there still are Web sites and services using closed protocols. It is highly frustrating when we cannot access on-line content because we don't have a proprietary plugin, such as ActiveX or Adobe Shockwave. For example, our school wants to use an on-line education tool to enhance our curriculum, but the company that offers this tool relies on Shockwave. So we are "locked out" because of this one missing piece.

Finally, a lack of key commercial software is a real issue. Some good people in the Free Software community don't want commercial software on Linux, but I have to be more pragmatic. When there is a fine open-source alternative to a key commercial product (such as with OpenOffice.org and Microsoft Office), I am happy to use it. Unfortunately, not all proprietary software has a good open-source equivalent. Until there is, the solution isn't eloquent. GHCA has a single Windows machine in our office for the sole purpose of running Intuit's QuickBooks. I suppose we could use Wine, but that brings its own headaches.

Despite these pitfalls, I have no regrets. Let's look at those big questions again. Is Linux ready for the desktop? Yes. Our teachers and students have been using Linux on the desktop successfully for the last five years. What about TCO? Every organization is unique, but Linux has saved us many thousands of dollars, and we're a small school! Have users adjusted? Absolutely. Was it worth the switch? There is no doubt in my mind. That's not to say there haven't been bumps in the road, but to quote Robert Frost, "I took the [road] less traveled by, and that has made all the difference." I look forward to where this road we call Linux will lead us in the future.■

Michael Surran is the head of GHCA's Computer Science department. He is responsible for building and maintaining GHCA's Linux network, and he teaches Computer Programming, Computer Technology, Research and Presentation, and the CS electives. Surran promotes open source in education both locally and regionally through newscasts and seminars.

Resources

Linux from Kindergarten to High School: **www.linuxjournal.com/article/6349**

Making the Switch to Open Source Software: **www.thejournal.com/articles/16448**

Harnessing the Power of Open Source Software: **video.google.com/videoplay?docid=7860580137648446279**

SchoolForge: **www.schoolforge.net**

GHCA's Computer Lab: **www.ghca.com/computers**

Zenoss and the Art of Network Monitoring

If a server goes down, do you want to hear it?   JERAMIAH BOWLING

If a tree falls in the woods and no one is there to hear it, does it make a sound? This the classic query designed to place your mind into the Zen-like state known as the silent mind. Whether or not you want to hear a tree fall, if you run a network, you probably want to hear a server when it goes down. Many organizations utilize the long-established Simple Network Management Protocol (SNMP) as a way to monitor their networks proactively and listen for things going down.

At a rudimentary level, SNMP requires only two items to work: a management server and a managed device (or devices). The management server pulls status and health information at regular intervals from the managed devices and stores the information in a table. Managed devices use local SNMP agents to notify the management server when defined behavior occurs (such as errors or "traps"), which are stored in the same table on the server. The result is an accurate, real-time reporting mechanism for outages. However, SNMP as a protocol does not stipulate how the data in these tables is to be presented and managed for the end user. That's where a promising new open-source network-monitoring software called Zenoss (pronounced Zeen-ohss) comes in.

Available for most Linux distributions, Zenoss builds on the basic operation of SNMP and uses a comprehensive interface to manage even the largest and most diverse environment. The Core version of Zenoss used in this article is freely avail-

Available for most Linux distributions, Zenoss builds on the basic operation of SNMP and uses a comprehensive interface to manage even the largest and most diverse environment.

able under the GPLv2. An Enterprise version also is available with additional features and support. In this article, we install Zenoss on a CentOS 5.1 system to observe its usefulness in a network-monitoring role. From there, we create a simulated multisystem server network using the following systems: a Fedora-based Postfix e-mail server, an Ubuntu server running Apache and a Windows server running File and Print services. To conserve space, only the CentOS installation is discussed in detail here. For the managed systems, only SNMP installation and configuration are covered.

Building the Zenoss Server

Begin by selecting your hardware. Zenoss lacks specific hardware

requirements, but it relies heavily MySQL, so you can use MySQL requirements as a rough guideline. I recommend using the fastest processor available, 1GB of memory, fast enough hard disks to provide acceptable MySQL performance and Gigabit Ethernet for the network. I ran several test configurations, and this configuration seemed adequate enough for a medium-size network (100+ nodes/devices). To keep configuration simple, all firewalls and SELinux instances were disabled in the test environment. If you use firewalls in your environment, open ports 161 (SNMP), 8080 (Zenoss Management Page) and 514 (if you integrate syslog with Zenoss).

Install CentOS 5.1 on the server using your own preferences. I used a bare install with no X Window System or desktop manager. Assign a static IP address and any other pertinent network information (DNS servers and so forth). After the OS install is complete, install the following packages using the yum command below:

```
yum install mysql mysql-server net-snmp net-snmp-utils gmp httpd
```

If the mysqld or the httpd service has not started after yum installs it, start it and set it to run for your configured runlevel. Next, download the latest Zenoss Core .rpm from Sourceforge.net (2.1.3 at the time of this writing), and install it using rpm from the command line. To start all the Zenoss-related dæmons after the .rpm has been installed, type the following at a command prompt:

```
service zenoss start
```

Launch a Web browser from any machine, and type the IP address of the Zenoss server using port 8080 (for example, http://192.168.142.6:8080). Log in to the site using the default account admin with a password of zenoss. This brings up the main dashboard. The dashboard is a compartmentalized view of the state of your managed devices. If you don't like the default display, you can arrange your dashboard any way you want using the various drop-down lists on the portlets (windows). I recommend setting the Production States portlet to display Production, so we can see our test systems after they are added.

Almost everything related to managed devices in Zenoss revolves around classes. With classes, you can create an infinite number of systems, processes or service classifications to monitor. To begin adding devices, we need to set our SNMP community strings at the top-level /Devices class. SNMP community strings are like passphrases used to authenticate traffic between devices. If one device wants to communicate with

another, they must have matching community names/strings. In many deployments, administrators use the default community name of public (and/or private), which creates a security risk. I recommend changing these strings and making them into a short phrase. You can add numbers and characters to make the community name more complex to guess/crack, but I find phrases easier to remember.

Click on the Devices link on the navigation menu on the left, so that /Devices is listed near the top of the page. Click on the zProperties tab and scroll down. Enter an SNMP community string in the zSNMPCommunitiy field. For our test environment, I used the string whatsourclearanceclarence. You can use different strings with different subclasses of systems or individual systems, but by setting it at the /Devices class, it will be used for any subclasses unless it is overridden. You also could list multiple strings in the zSNMPCommunities under the /Devices class, which allows you to define multiple strings for the discovery process discussed later. Make sure your community string (zSNMPCommunity) is in this list.

Installing Net-SNMP on Linux Clients

Now, let's set up our Linux systems so they can talk to the Zenoss server. After installing and configuring the operating systems on our other Linux servers, install the Net-SNMP package on each using the following command on the Ubuntu server:

```
sudo apt-get install snmpd
```

And, on the Fedora server use:

```
yum install net-snmp
```

Once the Net-SNMP packages are installed, edit out any other lines in the Access Control sections at the beginning of the /etc/snmp/snmpd.conf, and add the following lines:

```
##       sec.name  source        community
com2sec local      localhost     whatsourclearanceclarence
com2sec mynetwork 192.168.142.0/24    whatsourclearanceclarence

##  group.name   sec.model  sec.name
group MyROGroup  v1         local
group MyROGroup  v1         mynetwork
group MyROGroup  v2c        local
group MyROGroup  v2c        mynetwork

##          incl/excl subtree  mask
view all    included   .1         80

##               context sec.model sec.level prefix  read   write  notif
access MyROGroup ""        any       noauth   exact   all    none   none
```

Do not edit out any lines beneath the last Access Control sections. Please note that the above is only a mildly restrictive configuration. Consult the snmpd.conf file or the Net-SNMP documentation if you want to tighten access. On the Ubuntu

server, you also may have to change the following line in the /etc/snmp/default file to allow SNMP to bind to anything other than the local loopback address:

```
SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -I -smux -p /var/run/snmpd.pid'
```

Installing SNMP on Windows

On the Windows server, access the Add/Remove Programs utility from the Control Panel. Click on the Add/Remove Windows Components button on the left. Scroll down the list of Components, check off Management and Monitoring Tools, and click on the Details button. Check Simple Network Management Protocol in the list, and click OK to install. Close the Add/Remove window, and go into the Services console from Administrative Tools in the Control Panel. Find the SNMP service in the list, right-click on it, and click on Properties to bring up the service properties tabs. Click on the Traps tab, and type in the community name. In the list of Trap Destinations, add the IP address of the Zenoss server. Now, click on the Security tab, and check off the Send authentication trap box, enter the community name, and give it READ-ONLY rights. Click OK, and restart the service.

Return to the Zenoss management Web page. Click the Devices link to go into the subclass of /Devices/Servers/Windows, and on the zProperties tab, enter the name of a domain admin account and password in the zWinUser and zWinPassword fields. This account gives Zenoss access to the Windows Management Instrumentation (WMI) on your Windows systems. Make sure to click Save at the bottom of the page before navigating away.

Adding Devices into Zenoss

Now that our systems have SNMP, we can add them into Zenoss. Devices can be added individually or by scanning the network. Let's do both. To add our Ubuntu server into Zenoss, click on the Add Device link under the Management navigation section. Enter the IP address of the server and the community name. Under Device Class Path, set the selection to /Server/Linux. You could add a variety of other hardware, software and Zenoss information on this page before adding a system, but at a minimum, an IP address name and community name is required (Figure 1). Click the Add Device button, and the discovery process runs. When the results are displayed, click on the link to the new device to access it.

To scan the network for devices, click the Networks link under Browse By section of the navigation menu. If your network is not in the list, add it using CIDR notation. Once added, check the box next to your network and use the drop-down arrow to click on the Select Discover Devices option. You will see a similar results page as the one from before. When complete, click on the links at the bottom of the results page to access the new devices. Any device found will be placed in the /Discovered class. Because we should have discovered the Fedora server and the Windows server, they should be moved to the /Devices/Servers/Linux and /Devices/Servers/Windows classes, respectively. This can be done from each server's Status tab by using the main drop-down
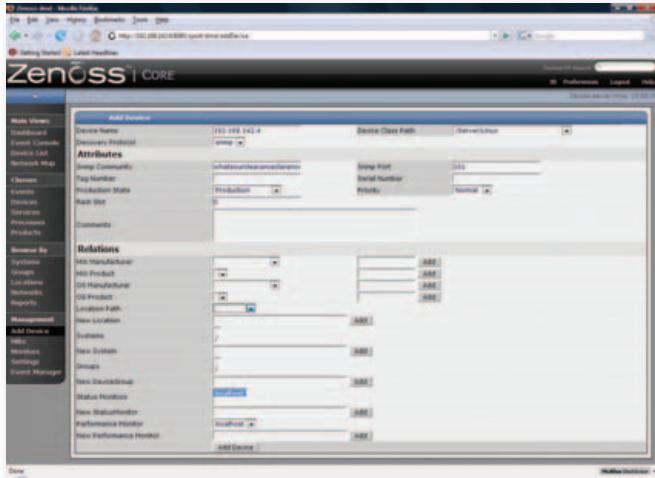
Figure 1. Adding a Device into Zenoss

Figure 3. Performance data is collected almost immediately after discovery.

Figure 2. The Zenoss Dashboard
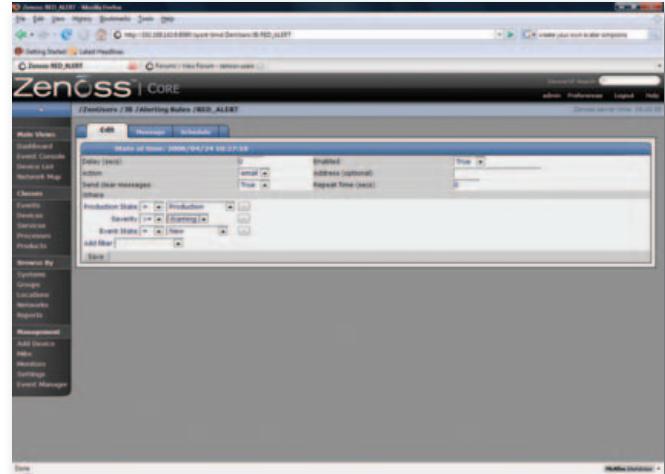
Figure 4. Creating an Alert Rule

list and selecting Manage→Change Class.

If all has gone well, so far we have a functional SNMP monitoring system that is able to monitor heartbeat/availability (Figure 2) and performance information (Figure 3) on our systems. You can customize other various Status and Performance Monitors to meet your needs, but here we will use the default localhost monitors.

Creating Users and Setting E-Mail Alerts

At this point, we can use the dashboard to monitor the managed devices, but we will be notified only if we visit the site. It would be much more helpful if we could receive alerts via e-mail. To set up e-mail alerting, we need to create a separate user account, as alerts do not work under the admin account. Click on the Setting link under the Management navigation section. Using the drop-down arrow on the menu, select Add User. Enter a user name and e-mail address when prompted. Click on the new user in the list to edit its properties. Enter a

password for the new account, and assign a role of Manager. Click Save at the bottom of the page. Log out of Zenoss, and log back in with the new account. Bring the settings page back up, and enter your SMTP server information. After setting up SMTP, we need to create an Alerting Rule for our new user. Click on the Users tab, and click on the account just created in the list. From the resulting page, click on the Edit tab and enter the e-mail address to which you want alerts sent. Now, go to the Alerting Rules tab and create a new rule using the drop-down arrow. On the edit tab of the new Alerting Rule, change the Action to email, Enabled to True, and change the Severity formula to >= Warning (Figure 4). Click Save.

The above rule sends alerts when any Production server experiences an event rated Warning or higher (Figure 5). Using a filter, you can create any number of rules and have them apply only to specific devices or groups of devices. If you want to limit your alerts by time to working hours, for example, use the Schedule tab on the Alerting Rule to define a window. If

2008

5th Annual

HIGH PERFORMANCE ON WALL STREET

September 22, Monday   Roosevelt Hotel, New York, NY

Madison Ave at East 45th St. next to Grand Central Station

High Performance Computing, Grid, Blade, Virtualization, Low Latency, Linux systems will all be there.

The 2008 High Performance on Wall Street will return to the Roosevelt Hotel, New York by popular demand. Attendees reported this is the best New York show to see HPCsystems that are changing the way Wall Street does business.

Plan to see HPC, Linux, Grid, Blade, Utility, Open Source solutions for IT management in the financial sector.

Wall Street IT chiefs are looking for reduced total-cost-of-ownership, reduced space, reduced heat, and increased energy efficiency. The big savings are in consolidations, time-saving deployment, and money-saving Grid applications.

Presentations for the conference are now being sought. For further information, for sponsorship and exhibit opportunities, contact

Russell Flagg
Flagg Management Inc
(212) 286 0333 or
flaggmgmt@msn.com

Show Hours:	8 - 4:00 pm
Conference Hours:	9 - 4:50 pm

2007 Sponsors

IBM   intel   CISCO   Microsoft

redhat   U1 Technologies   Novell

SUPERMICRO   Celoxica   AMD

GIGASPACES   Skyler   TERVELA

QLOGIC   Data Synapse   SD Times

Conference Producer: A-TEAMGROUP

Show Management:
Flagg Management Inc
353 Lexington Ave, NY10016
(212) 286 0333
flaggmgmt@msn.com

Conference Management:
Pete Harris
(718) 237 2796
pete@a-teamgroup.com

Register for the low-cost conference and free Show badge:

www.highperformanceonwallstreet.com

no schedule is specified (the default), the rule runs all the time. In our rule, only one user will be notified. You also can create groups of users from the Settings page, so that multiple people are alerted, or you could use a group e-mail address in your user properties.
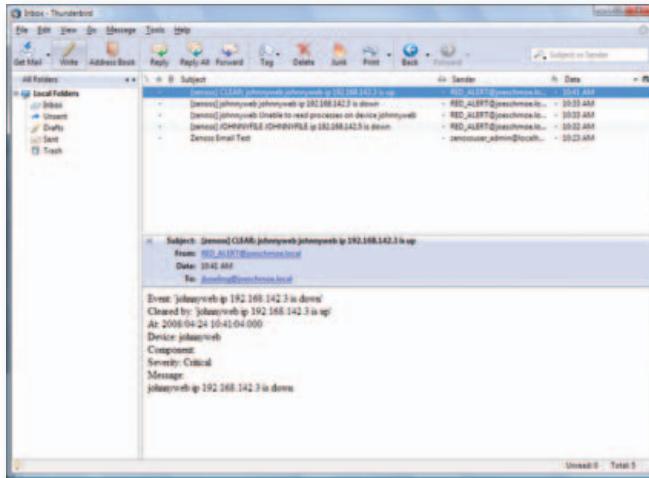
Figure 5. Zenoss alerts are sent fresh to your mailbox.

Services and Processes

We can expand our view of the test systems by adding a process and a service for Zenoss to monitor. When we refer to a process in Zenoss, we mean an active program, usually a dæmon, running on a managed device. Zenoss uses regular expressions to monitor processes.

To monitor Postfix on the mail server, first, let's define it as a process. Navigate to the Processes page under the Classes section of the navigation menu. Use the drop-down arrow next to OS Processes, and click Add Process. Enter Postfix as the process ID. When you return to the

Using a filter, you can create any number of rules and have them apply only to specific devices or groups of devices.

previous page, click on the link to the new process. On the edit tab of the process, enter `master` in the Regex field. Click Save before navigating away. Go to the zProperties tab of the process, and make sure the zMonitor field is set to True. Click Save again. Navigate back to the mail server from the dashboard, and on the OS tab, use the topmost menu's drop-down arrow to select Add→Add OSProcess. After the process has been added, we will be alerted if the Postfix process degrades or fails. While still on the OS tab of the server, place a check mark next to the new Postfix process, and from the OS Processes drop-down menu, select Lock OSProcess. On the next set of options, select

Lock from deletion. This protects the process from being overwritten if Zenoss remodels the server.

Services in Zenoss are defined by active network ports instead of running dæmons. There are a plethora of services built in to the software, and you can define your own if you want to. The built-in services are broken down into two categories: IPServices and WinServices. IPServices use any port from 1-65535 and include common network apps/protocols, such as SMTP (Port 25), DNS (53) and HTTP (80). WinServices are intended for specific use with Windows servers (Figure 6).

Adding a service is much simpler than adding a process, because there are so many predefined in Zenoss. To monitor the HTTP service on our Web server, navigate to the server from the dashboard. Use the main menu's drop-down arrow on the server's OS tab arrow, and select Add→Add IPService. Type `HTTP` in the Service Class Field. Notice that the field begins to prefill with matches as you

Figure 6. Zenoss comes with a plethora of predefined Windows services to monitor.
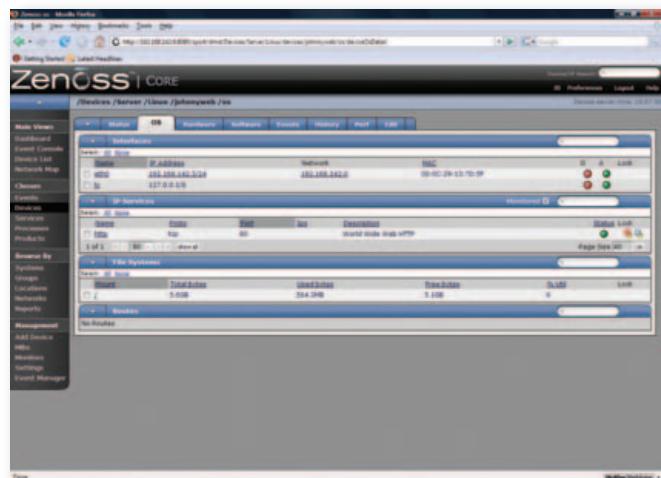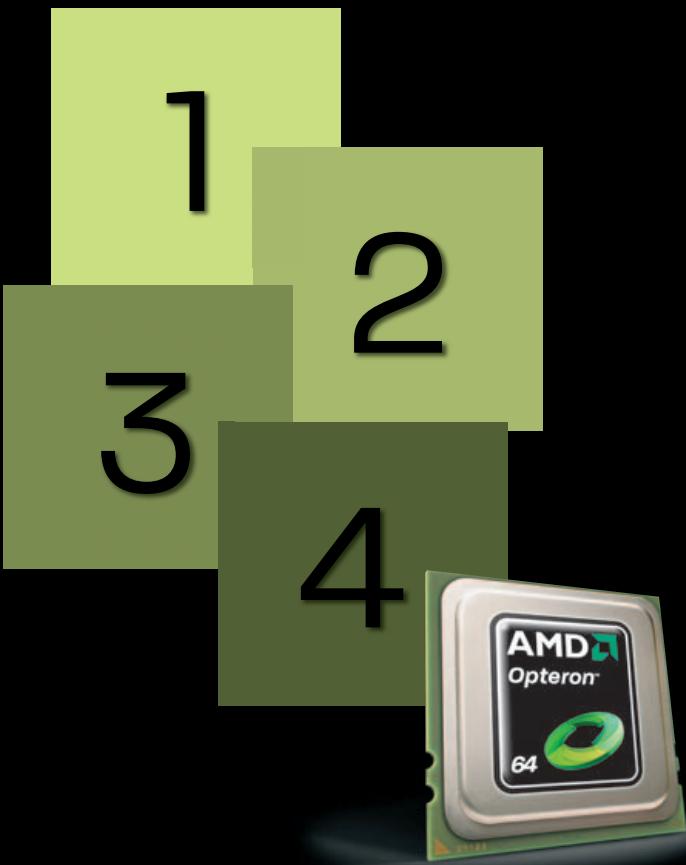
Figure 7. Monitoring HTTP as an IPService

Everything you expect from an AMD Opteron™ Processor – to the fourth power

The Quad-Core AMD Opteron™ Processor delivers much more than just four cores.

1 Industry-leading performance-per-watt. For industry-leading savings. **2** Consistent architecture, consistent power and thermal envelopes. For superior investment protection. **3** True quad-core with Direct Connect Architecture. For a whole new level of performance. **4** Optimal virtualization technology.

To see a wide array of Quad-Core AMD Opteron™ processor-based solutions visit www.amd.com/opteronnow

AMD
Smarter Choice

type the letters. Select TCP as the protocol, and click OK. Click Save on the resulting page. As with the OSProcess procedure, return to the OS tab of the server and lock the new IPService. Zenoss is now monitoring HTTP availability on the server (Figure 7).

Only the Beginning

There are a multitude of other features in Zenoss that space here prevents covering, including Network Maps (Figure 8), a Google Maps API for multilocation monitoring (Figure 9) and Zenpacks that provide additional monitoring and performance-capturing capabilities for common applications.

In the span of this article, we have deployed an enterprise-grade monitoring solution with relative ease. Although it's surprisingly easy to deploy, Zenoss also possesses a deep feature set. It easily rivals, if not surpasses, commercial competitors in the same product space. It is easy to manage, highly customizable and supported by a vibrant community.

Although you may not achieve the silent mind as long as you work with networks, with Zenoss, at least you will be able to sleep at night knowing you will hear things when they go down. Hopefully, they won't be trees.■
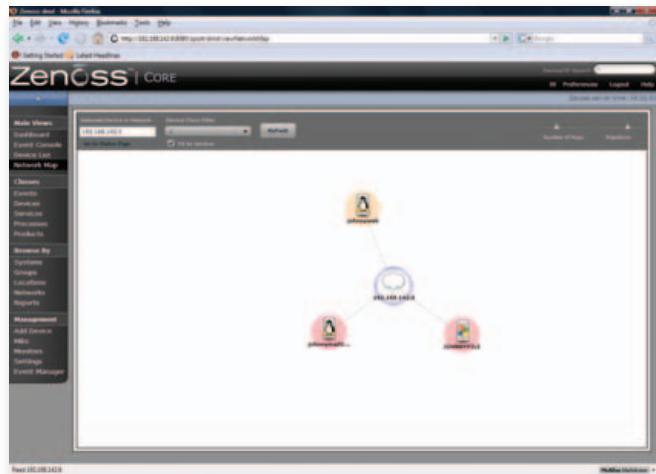
Figure 8. Zenoss automatically maps your network for you.
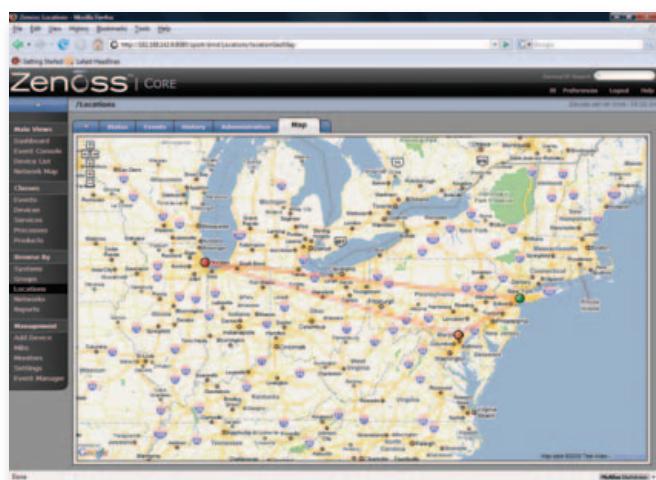
Figure 9. Multiple sites can be monitored geographically with the Google Maps API.

Jeramiah Bowling has been a systems administrator and network engineer for more than ten years. He works for a regional accounting and auditing firm in Hunt Valley, Maryland, and holds numerous industry certifications, including the CISSP. Your comments are welcome at jb50c@yahoo.com.

Resources

Zenoss: **www.zenoss.com**

Zenoss SourceForge Downloads Page: **sourceforge.net/ project/showfiles.php?group_id=163126**

NET-SNMP: **net-snmp.sourceforge.net**

CentOS: **www.centos.org**

CentOS 5 Mirrors: **isoredirect.centos.org/centos/5/isos/i386**

HUNDREDS OF LINUX SOLUTIONS ARE WAITING TO BE ADOPTED

WON'T YOU TAKE AN OPEN SOURCE SOLUTION HOME TODAY?

Today's open source solutions drive growth, productivity, and competitive advantage from the data center to the desktop and in mission-critical apps of all kinds. Are they doing everything they could for your organization?

There's only one event that brings you the high-level strategic guidance, unmatched product exhibition, and nuts-and-bolts training to put it all to work. Come to LinuxWorld and discover what Linux and open source can do for your most pressing business needs—and your entire enterprise.

REGISTER TODAY AT
www.linuxworldexpo.com

LINUXWORLD
CONFERENCE & EXPO ®

August 4–7, 2008

The Moscone Center
San Francisco, CA

AN IDG WORLD EXPO EVENT

LinuxWorld is open to business professionals only. No one under 18 years of age will be admitted.

How to Fake a UFO Landing

The magic of Voodoo.   DAN SAWYER

A flying saucer descends onto an open field and lands, kicking up dust all around it. If this happened in a remake of *The Day the Earth Stood Still* nobody would blink. But imagine that instead of a big beautiful image executed with the precision and care of a big-budget feature, what you're watching looks like it fell on the cutting-room floor of *The Blair Witch Project*. You're not seeing this in a comfortable screening room at your local cinema, where the picture is clean, sharp and bigger than life, but rather you're standing gathered around a booth at your local science-fiction convention. The guy playing the video isn't a producer. He isn't even an independent filmmaker. He's a guy who's genuinely convinced that this video can't be faked. If it were, he says, the seams would show, and whoever gave this to him really did record evidence of alien visitation that the government is covering up, and that by showing this publicly, he's taking a terrible risk. But, he feels he must expose the fraud that governments and aliens perpetrate on unsuspecting citizens!

This scenario may sound like it clawed its way out of the *X-Files'* wastebasket, but as VisualFX technology gets ever cheaper and more ubiquitous, faking a video like this

> **This scenario may sound like it clawed its way out of the *X-Files'* wastebasket, but as VisualFX technology gets ever cheaper and more ubiquitous, faking a video like this becomes no problem.**

becomes no problem. Of course, it takes a lot of expertise and dedication to get the colors, shadows and reflections to match convincingly. One would think that getting the movement to match as the camera person runs and zooms with a handheld shot would be the most difficult part of the equation. Once upon a time, this was true.

It used to be that the only way you could achieve the movement precision necessary to sell an effect like this was to put your camera on a motion control rig and have a computer record the movements in the field and then reproduce them exactly (though to a smaller scale) in the post house where the artificial elements (in our case, the flying saucer and the dust) were photographed. Aside from being very cumbersome and expensive, this approach sharply limited the kinds of shots that an FX artist could do to those that could be reproduced by an electric gimbal and a prime lens.

No longer. The late 1990s saw a great flowering of research and development into the area of computerized match moving—matching the movement of different visual elements so that they appeared to exist organically in the same scene. Putting the computer in the mix both at the match moving and at the compositing stage gives a lot more control and freedom than previously.

Why Use Match-Moving Software at All?

Human visual acuity isn't the best on the planet, but it is startlingly good. With a little practice, an ordinary fellow sitting in the audience for *The Matrix* can spot the grain mismatch in shots that were too hastily done. Our visual cortex does the differential calculus to tell us "this doesn't belong here". It follows that this same mental apparatus could be employed to create the trickery in the first place. With most VisualFX work, there are complicated tools, and then there is doing it by hand. Like most other fields of human endeavor, the better an artist is, the fewer training wheels he or she generally will rely on. So, why not do match moving by hand?

The short answer is that many artists do, under some circumstances. Other times, there is an interaction between the artist and the match-moving software, with the artist choosing points for the software to track, either because the tool doesn't detect the right points, latches on to points that aren't appropriate or doesn't do point detection at all.

However, the art of motion tracking is nontrivial. Although our visual cortexes are excellent at detecting error, they are somewhat less excellent at projecting perfection outward. We do not create grand, realistic paintings naturally—indeed, we have to be taught to see light, shadow, form and so on in a certain way in order ponder even attempting to work like a Bouguereau or a Leonardo. Similarly, our ability actually to distinguish motion that doesn't fit is quite keen, although our ability to create a perfect motion path is coarse by comparison—something we don't notice until after we play it back and see the drift creeping in even with the most careful hand-tracks.

Of course, a match-moving program won't always get a perfect track, but the interaction of a good artist with a good program delivers top-notch results.

Why Get a Voodoo Doll?

Aside from the fact that it's free, why use Voodoo for this project?

The truth is that Voodoo isn't going to solve every match-moving problem, even leaving out the ultra-delicate moves that the higher-end match movers handle better. The field of match moving is basically divided in two: 2-D motion tracking and 3-D camera tracking.

2-D motion tracking is the technology used in compositors to affix a new element to a specific point in the frame. A user generally will select one or two feature points, and the computer then will follow the points around the frame as the objects move within it. When the tracker slides off the selected point, the artist gently will correct it to keep the track from drifting. Two commonplace examples of this process can be seen in blurring suspects' faces on *Cops!* and in placing virtual advertisements on infield walls at baseball games. 2-D tracking tracks only the position of an object within the frame, which gives it a double-barreled Achilles' heel: parallax and perspective.

Parallax is the phenomenon whereby foreground objects seem to move faster than do background objects. As your point of view moves, the angle at which you perceive objects changes subtly, which is why you see a parallax driving down the road. With 2-D tracking, your track marks are pretty much all you get. This can be a problem if, for instance, you're moving over a greenscreen and the digital set is supposed to extend for quite a ways down in depth. As soon as you add depth to lateral movement, particularly when your track marks are close to the camera, you need to work in 3-D, or you have to fake parallax by hand—a dubious and difficult undertaking that easily shatters the illusion you're trying to create. A really good artist can pull it off, but it takes a lot of practice.

Perspective is the other wild card in the equation. Lenses do not see the world as it is. Instead, every lens distorts the world in certain mathematically predictable ways. This distortion is closely related to focal length and aperture, and measuring the distortion accurately is essential to tracking elements properly in the shot. This wild card gets even wilder with zoom (extending the lens to get a closer shot) and dolly-in (moving the camera toward a subject) movements, which involve constantly changing perspective in one fashion or another along the z axis, which is the axis that 2-D motion tracking can't cope with. Perspective changes also can be faked, but it's far more difficult than faking parallax and far more time consuming.

This is where 3-D camera tracking comes in. Instead of simply tracking the location of certain user-selected features to create a good 2-D track, the computer attempts to guess the position and motion of the camera based on the footage. Pitch, yaw, roll and lens length are all calculated based solely on the finished video (though any information you have and can input manually will make it work faster). The ability to reconstruct all these parameters accurately means that the problems of parallax and perspective are solved, even during dolly and zoom moves. Needless to say, this is a mathematically complex process designed to test the minds of even the most ardent effects artist who wasn't also a comp-sci or optics major at a university. Nonetheless, the algorithms for pulling this off are well known and included in most camera-tracking packages.

Although most 2-D motion trackers are built in to existing compositing systems (such as After Effects), 3-D camera trackers operate on a standalone basis and export their

data—camera settings and movement, as well as the "point cloud"—to various 3-D programs, and it is in the 3-D program where the magic happens. The 3-D program also gives an extra measure of control and refinement beyond what the tracker itself allows, as you can tweak the camera animation curves.

I said earlier that the 1990s saw a lot of funding into creating software like this. Well, as every tech-junkie knows, where thy research funding is, there thy grad students also will be. Thanks to a team of particularly dedicated grad students in Hannover, Germany, the technology to match camera movement in three dimensions is available to Linux and Windows users for free—a very good deal, considering that comparable commercial packages run upward of several thousand dollars a seat. For the savings, you do sacrifice some sophistication in the ability to fine-tune your shot, but for most applications, Voodoo does very well.

So, grab a copy of it, and let's get you ready for your appearance on the *Art Bell* show, peddling your newest Genuine UFO Video (tm)!

The Incantation

First, head over to **www.digilab.uni-hannover.de/ download.html**, and download a package appropriate to your system. Note that there are no source packages— Voodoo may be freeware, but it is not, and probably never will be, open source. So, grab the binary that is convenient for you. Note that there are no x86_64 binaries available. If you have a 64-bit system, just grab the x86 package—it doesn't depend on any 32-bit libs to work, and it won't choke on execution.

Pop open a command window, and use `tar -xvzf` to open the archive. Next, move into the resulting voodoo-versionnumber directory, then a further level deep into the /bin directory, and run `./voodoo`.

The Bloodletting

Anyone familiar with old Roger Corman movies will realize that bloodletting is an essential step in working good voodoo. In this case, it's your video that needs to be bashed into tiny bits. Voodoo will not chew through video, it works only on still image sequences.

A quick ffmpeg call will give you the image sequence you require:

```
ffmpeg -i videofilename.avi -f image2 %03d.png
```

Once done, run Voodoo.

The interface, at first glance, is simple—two pull-down menus and a flipbook player. That simplicity proves quite illusory as you begin to delve into it. Camera tracking is complex, and the toolbox here is extensive, but the nature of the task means that you can learn gradually, and very little work will get you a good initial track.

To start, go to the File menu and select load→image sequence, and load up the image sequence you just created. Be sure to set movement type and interlace settings,
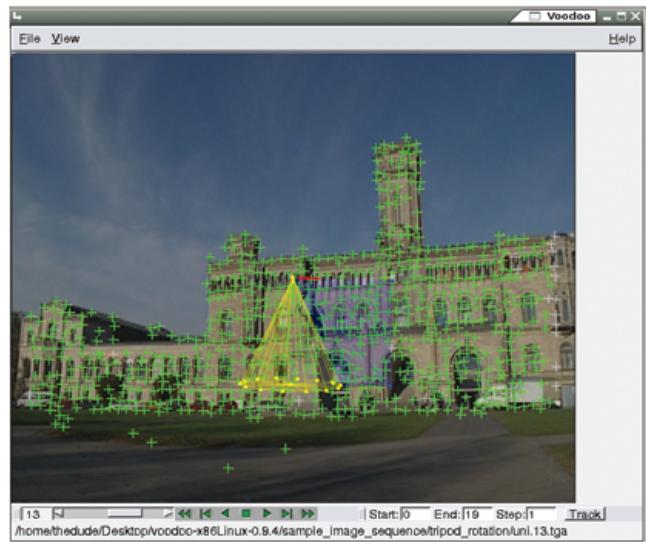

Figure 1. Voodoo Interface
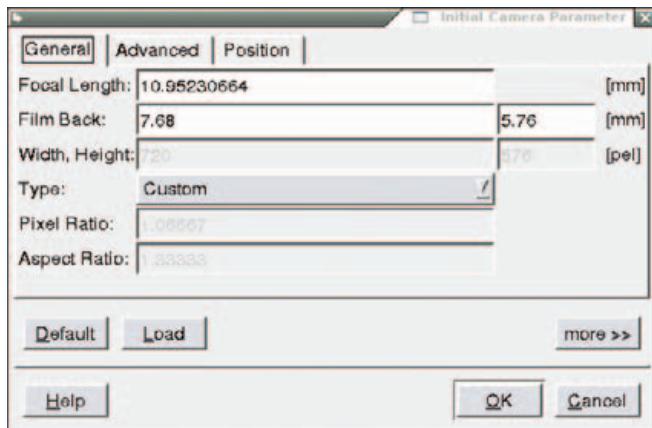
Figure 2. Camera Settings Window

or your track will not come out properly. Play the clip through once with the flipbook to make sure there aren't any obvious errors.

Now, you need to load camera settings (File→load→initial camera). This is vital if you want the track to work properly, but it's also very difficult to get right if you weren't keeping notes on the set for focal length, aspect ratio, film back and (less important) skew angles. If you didn't keep proper notes, enter your best guess and go from there, you always can tweak it later.

The work flow from here is pretty simple. Play through the clip to make sure the whole sequence loaded properly, then press track. The computer will select a few dozen track points and follow them through the duration of the clip. Depending on the complexity and length of the clip, this process can run anywhere from a few seconds to a few hours.

Once the track is done, play it through again, watching

particularly the motion path of the different points. If you can't see any drift, you're golden—you can skip ahead to the export step. If the track is lacking, there are a number of ways to tweak it. You can refine by adjusting the tracking algorithms in the View→Controls menu, and rerun the track, selecting refine instead of discard in the dialog that presents itself to augment the track you've already created. You can do much the same through adjusting the camera settings, although if you do this, you'll be better off running the track from scratch.

A number of other refinement tools are also available. You can pull up the modeling box (View→Modeling Tools) and use it to add track masks and 3-D primitives to help you spot drift, and it (along with the Fpoint track editor) lets you delete, change or add new track points manually, so you can direct the tracker to watch the right things and

For the savings, you do sacrifice some sophistication in the ability to fine-tune your shot, but for most applications, Voodoo does very well.

make it ignore the wrong ones, such as people or cars in the foreground. Once done, run the track again, again selecting refine rather than discard.

You can watch the reconstructed camera motion, and manipulate it to a certain extent, in the 3-D viewer window, available through the View menu.

When you have a track you find satisfactory, go to File→Save, and pick your export format. Be sure to export all the Fpoints—having them helps if you're going to need to do any complex interaction, as they will help you guide where you put alpha masks and such—like if you chose to do some of your masking in your 3-D program.

Sticking In the Pins
In Blender, importing your track data will give you something like what is shown in Figure 3.

The point cloud is a representation in 3-D space of the track points from Voodoo, and the camera has applied to it all the animation data (pitch, yaw, roll, position and lens length) to re-create the movement that the original camera engaged in. It is possible that upon import you will need to re-orient parts of your scene, but if you've done your job properly, all that needs to be done now is to finish your 3-D UFO (texturing, animation and so on), and create your dust cloud with your particle engine. Marrying these elements together with the tracked footage is a job for your compositor—Blender has a quite capable one built in, which I covered in depth in the October 2007 issue of *Linux Journal*.

With a bit of practice, you'll have your own fake UFO video suitable for posting on YouTube or fooling media
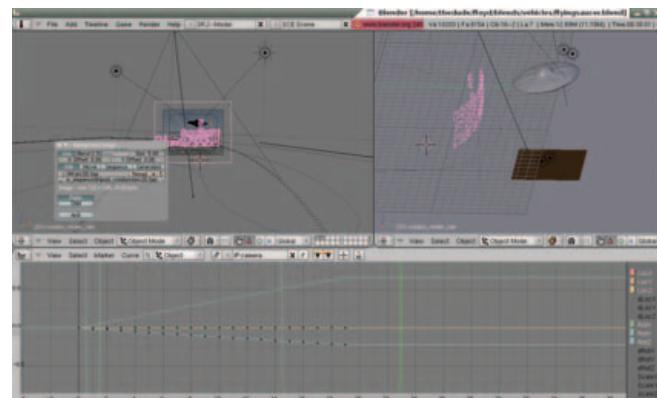
Figure 3. Blender File with Voodoo Track Imported and Point Cloud Showing
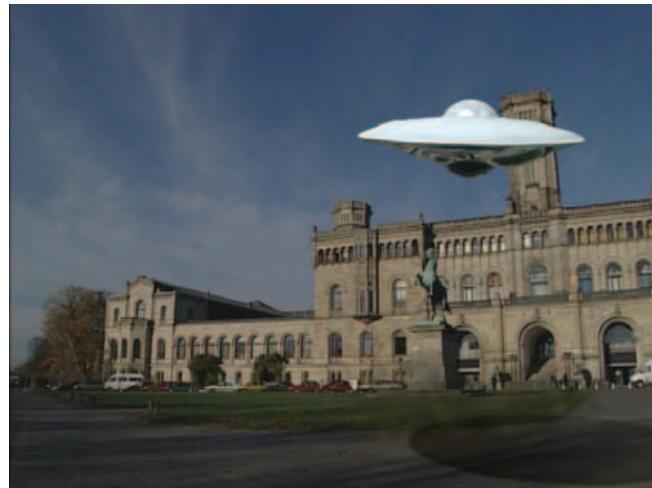
Figure 4. Finished Shot

pundits. Like anything, camera tracking takes practice to get right, but the toolset provided by Voodoo puts this technique well within the reach of any hobbyists willing to learn a bit about optics and spend some time training their eyes. Refer often to the on-line help—Voodoo is one of those rare freeware products with excellent documentation built right in.

Until an open-source camera tracker of equal sophistication presents itself, Voodoo likely will remain the only free camera tracker for Linux—at least in a price range that end users can afford. All hail the grad students and their advisors at the University of Hannover, Germany. Let's hope their excellent work remains free to use for the foreseeable future!■

Dan Sawyer is the founder of ArtisticWhispers Productions (www.artisticwhispers.com), a small audio/video studio in the San Francisco Bay Area. He has been an enthusiastic advocate for free and open-source software since the late 1990s, when he founded the Blenderwars filmmaking community (www.blenderwars.com). He currently is the host of "The Polyschizmatic Reprobates Hour", a cultural commentary podcast, and "Sculpting God", a science-fiction anthology podcast. Author contact information is available at www.jdsawyer.net.

Quantum GIS:
the Open-Source Geographic
Information System

Exploring Quantum GIS (QGIS) using an example of real-estate planning.   JAMES GRAY

If you've ever zoomed around the globe with Google Earth, you know how much fun it can be to work with geospatial data. When I need a diversion, I often fire up Google Earth and float above the skyscrapers of Manhattan or revisit former stomping grounds.

For a deeper level of control with geospatial information—where you're the chef who concocts the whole stew—dive into a geographic information system, or GIS. A GIS lets you control all the elements that go into the geophysical world you want to explore. Stripping GIS down to its essentials, you could call it computer-based mapmaking. However, because a GIS is powered by a database, the opportunities for advanced analysis are light-years beyond anything you could do with a paper-based map. A GIS not only will make you feel like you have the world in your hands—look out for that "I'm playing God" feeling—but you also probably will do something extremely useful with it for your work or private life.

This article introduces a sample project with Quantum GIS (QGIS), one of the most advanced and powerful open-source GIS packages for the desktop. Although QGIS has some excellent documentation, new users might find the terminology a bit stilted and missing some information. The authors of the documentation assume you already are familiar with GIS and that you're coming to QGIS from a proprietary alternative, such as the popular ArcGIS from ESRI. I, on the other hand, assume you've never used a GIS before.

A QGIS Test Project: Finding a Place to Build

To illustrate some basic functions of a desktop GIS, I use QGIS to make preparations for a fantasy of mine, which is to create an ecologically friendly real-estate development. In this exercise, I locate a parcel of agricultural land in Washtenaw County, Michigan, near Ann Arbor, where I can restore a former wetland and build a cluster of homes nearby. I chose Ann Arbor due to its proximity to drained wetlands in rural areas, as well as local demand for homes in areas with lots of wildlife.

To accomplish this task, I explore how to load QGIS on your system; find the geospatial data for the task; load that data into QGIS; and view, set up and analyze that data to do the job at hand. Along the way, I introduce key concepts and important terms.

Getting QGIS on Your System

QGIS has a useful, comprehensive Web site with plenty of resources to get you started. Beyond the free application download, you'll find a wiki, help forums and loads of documentation. QGIS has versions for Mac OS X, Windows and several variants for Linux users: source, Debian, Ubuntu Gutsy and OpenSUSE. Given that repositories are provided, installation should be easy and straightforward. All you need to do
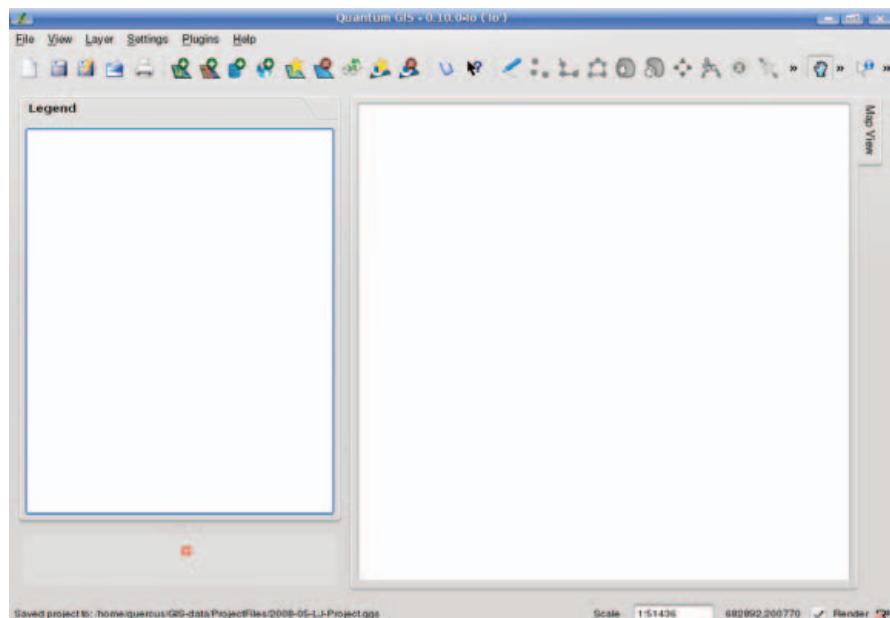

Figure 1. The application QGIS 0.10 offers a clean, intuitive user interface.

is add the requisite repository to your favorite package manager. If you must install from source, there are plenty of on-line guides explaining the process. See Figure 1 for a look at QGIS's GUI.

GIS is a complex application requiring knowledge about data formats, how a GIS functions and general cartography. Let's rip through a quick, need-to-know primer on GIS.

A GIS Needs Geospatial Data

As mentioned previously, using a GIS is essentially mapping on a computer. To do this mapping, you need to find data related to geography, typically called geospatial data. This geospatial data that we will introduce into QGIS consists of two elements, namely spatial features and attribute data. Examples of spatial features might include streets, rivers or land cover—any feature you might find on a map. Meanwhile, attribute data describes the characteristics of the spatial features and is stored in a database within the GIS. For example, most of those streets have names and lengths; the land-cover types have names and areas associated with them. In the case of land cover, a GIS might store attribute-related categories, such as high-density urban, low-density urban, cropland, forest and so on, which you then could query easily.

Peel Back the Layers

Your paper road map would think you were completely mad if you commanded it to "just show me the rivers and mountains, please" or "flip the county boundaries on and off". On the other hand, because a GIS portrays data in similar groupings of geographic elements, called layers, your computer will execute your command and not label you loopy. Some examples of layers are countries, cities, rivers and oceans. A GIS allows you to control which layers are displayed on your screen at any time.

Layers can consist of two types, namely features and surfaces. In our above list, the layers with countries, cities, rivers and specific buildings are feature-based; oceans are one single, continuous expanse and, thus, are a surface.

How a GIS Formats Data: Vector vs. Raster

The hefty challenge for a GIS is to portray our lovely yet complex world accurately yet rapidly—and without the need for a cluster! There are two tricks, or methods, a GIS uses to create a digital representation of Earth's features on your desktop.

The first method is using vector data (the type used later in this article). As complicated as the world can be, a GIS can represent any geographical object using three geometric elements—namely points, lines and polygons. Small stuff like community centers and traffic lights can be portrayed as points. Features such as rivers and pipelines are really just glorified lines, so they can be shown as such. Finally, nearly everything else, such as a state park, though it might be oddly shaped, is finite and contained in boundaries, making it a polygon at the end of the day. Broadly speaking, the vector format is analogous to traditional maps, where the world is abstracted with symbology, and precision is very important.

The second method is raster data. Raster data is used to portray Earth's characteristics that have no shape visually, including measurements like ocean depth, forest-cover type, elevation and annual rainfall. Some image types you will encounter include GeoTIFFs, Erdas Imagine Images, GRASS AIGs and USGS Digital Elevation Models. Some common examples of raster-based imagery are satellite images and aerial photos. In these two types of raster imagery, the value of each cell is a measurement of light that is reflected off the Earth's surface. Particular ranges of these values can signify specific land-cover or vegetation types.

Vector-Based Data Formats in GIS

As you splash around in the world of GIS, you also will encounter a plethora of vector-based spatial file formats. If you have ever used the application ArcGIS from ESRI, you probably are familiar with geodatabases and coverages, two of the most common spatial file formats in proprietary GIS. Of these two more-advanced spatial data formats, only coverages are usable in QGIS, but not geodatabases. In addition, in QGIS, we can utilize ESRI shapefiles, which are plentiful in online data repositories and a sort of standard, as they have been around a long time. In fact, shapefiles are the standard format for ESRI's ArcView, which is the company's previous generation of GIS applications. Essentially, a shapefile is a set of files with vector-based location and attribute data, which can be represented in a GIS application.

a map projection.

In a GIS, you need to consider the projection, because any map you view or create is essentially flat like a paper map. Thus, the same concept applies to both situations.

Just as important as the map projection is the coordinate system. A coordinate system is the Cartesian system of x and y axes that a GIS uses to define locations on a map. This is opposed to the latitude and longitude system that defines location on a sphere.

In larger projects, knowledge of projections and coordinate systems is very important, and if a mismatch exists among different parts of a project, life can get frustrating quickly. Fortunately, this project is simple enough to avoid much concern, as I am working at the county level and all my shapefiles come from the same data source. However, when working with larger areas and multiple data sources, it is important to be familiar

must acquire. Thus, for this project, we need layers that depict, respectively, land use, areas with potential for wetland restoration, roads and hydrography (rivers and lakes). In general, the most common format for each layer will be in the form of a shapefile, which QGIS can handle without a hitch.

So where can I obtain these shapefiles? Fortunately, a plethora of excellent repositories of free, downloadable geospatial data exist. An excellent example is the public Michigan Geographic Data Library (MGDL), which offers a vast collection of vector- and raster-based data at the watershed, county and state levels. Just some of the datasets available include those I am looking for, as well as aerial photos of the entire state, federal census information, geology, soil types, public land ownership and topography. In the MGDL, the default format for vector-based data is the shapefile.

From the MGDL, I can download the following datasets at the county extent:

- Michigan Geographic Framework Hydrography (lakes and rivers).

- 1992 National Land Cover Dataset.

- Michigan Geographic Framework Transportation (roads).

- Potential Wetland Restoration.

Loading Shapefiles into QGIS

Loading shapefiles into QGIS is done by clicking the toolbar icon labeled Add vector layer, which looks like a plus sign hovering over a map; it opens a standard open file dialog. By preselecting ESRI shapefile (suffix .shp) from Files of Type, I can be sure I'm opening the right file, which is useful, because a shapefile is actually a bundle of files. As I load each shapefile, it shows up under its original name on the left under the Legend window, which acts as a sort of table of contents.

After unpacking the datasets, I load these five shapefiles in this order: allroads_161v7b.shp (roads), hydro_161v7b.shp (rivers), hydropoly_161v7b.shp (lakes), Washtenaw_Potential_Restoration_Area.shp

Stripping GIS down to its essentials, you could call it computer-based mapmaking.

QGIS also supports some other file formats, such as MapInfo and PostGIS. PostGIS is especially interesting, as it is an open-source spatial database technology. PostGIS "spatially enables" the PostgreSQL server, allowing it to be used as a back-end spatial database for GIS and—for those who are familiar with GIS technologies—as such, is similar to ESRI's SDE or Oracle's Spatial extension.

Some Hard-Core Cartography: Projections and Coordinate Systems

Two other important concepts critical to any cartographic endeavor are map projections and coordinate systems.

Remember the big, flat world map you had in your fourth-grade classroom? The one with Greenland bigger than Africa? That map is an ideal illustration of what happens when you depict a round object such as the earth onto a flat map. Converting a 3-D globe onto a 2-D map is called

with these concepts and standardize your projection and coordinate system project-wide.

Enough Theory, Let's Get Some Data!

At this point, we have enough GIS theory to understand what we're doing and start the real-estate planning project. At this stage, I track down the requisite data.

This project involves finding a parcel of land in Washtenaw County, Michigan, where I can build a cluster of homes in a natural setting. I am looking for a suitable land parcel that was once a wetland but today is agricultural and suitable for conversion back to a wetland. The ideal site will be close to a river or lake, have good road access and be as close to the city of Ann Arbor as possible.

When you embark on a GIS-based project, it's wise to specify all of the elements you need, because in general, each will likely be one of the layers you

(the name says it all) and Washtenaw_nlcd_1992.shp (land use).

Making Things Look Right

Unfortunately, upon loading the shapefiles, the sum total map that's displayed on the right in the Map View window looks like a big rectangle covered with random black and green blobs and no lines. Where are the roads, lakes and rivers I loaded? One reason for the odd display and missing elements is that the layers I added first are buried under the county-wide land-use layer, which sits on top of everything else. I can begin to solve this problem by dragging the land-use layer down to the bottom of the Legend and tinkering with the other layers so they all are visible.

The other reason for the strange-looking map is that QGIS defaults to display one color for every characteristic in the shapefile. For the road layer, defaulting to one color is fine, because it is simply a collection of lines. However, layers with thousands of polygons are more complicated. All of the many land-use types default to the same color, thus creating no differentiation among them. I must give each land-use type its own unique color manually. To do so, I first right-click on the land-use entry in the legend and select Properties from the menu. On the Symbology tab, I change the drop-down menu next to Legend type from the default value of Single Symbol to Unique Value. Using the drop-down menu in Classification Field, I can select which field in the database to

Figure 2. The attribute table displays the data contained in a particular layer, for example, a shapefile.

classify. In my case, I classify a field called GRIDCODE, which contains the code that designates the land-use code for each polygon in the layer.

How do I know which database field I should classify, as well as the meaning of each classification? To find out, I sometimes need to leave the Layer Properties menu and examine the attribute table, a display of the database containing the attribute data for the layer. For example, I can examine the attribute table of the land-use layer by right-clicking on the title in the Legend (on the main GUI) and selecting the command Open attribute table. An example of an attribute table is shown in Figure 2. The land-use attribute layer contains a field ID to designate each polygon, as well as the field GRIDCODE to classify each one. Oftentimes the

attribute table also contains a field with the label for each classification. Although such a field is missing from the land-use attribute table, a separate file with classifications is found in a text file included in the downloaded dataset.

After consulting the attribute table and the file containing classifications, I am ready to continue with the classification of the field GRIDCODE back in the Layer Properties menu. Pressing the Classify button populates the window below with the unique classification codes found in the layer. I can label each classification as I wish using the "Label" field, and I can give each classification its own color with the Fill color option.

After finishing the classification, I also want to do some more house-keeping to make the Legend and

---

TECH TIP — Server Name in Title Bar

If you often have multiple putty, terminal, ssh or screen sessions connected to various remote servers, one good way to organize them is to have a small script that places the name of the remote server in the title bar:

```
#!/usr/bin/perl -w

$name = $ARGV[0];
unless ($name) { $name = `/bin/hostname` }
print "\033]0;$name\007";
```

Save this, and make it executable. If, for example, you save it as name, you simply can run name to place the name of the current server in the title bar of your current session.

If you want to label the session with something besides the hostname of the server, just specify the label on the command line:

```
# name "Mail Server"
```

—FRED RICHARDS
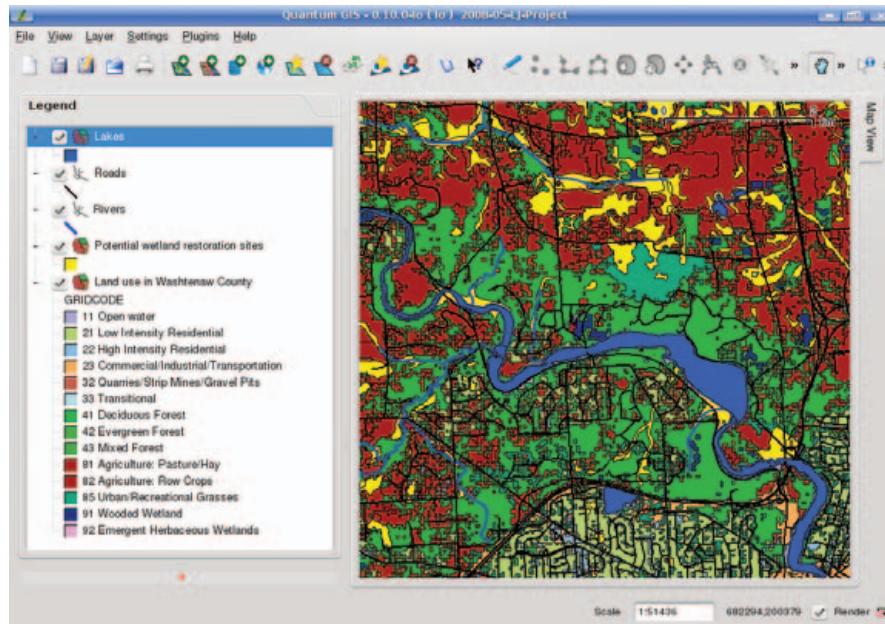
Figure 3. After modifying the properties of each layer and changing the layer names in the Legend, the Map View is readable and ready for analysis.

Map View more useful, such as making the colors of the other layers more intuitive (for example, blue lakes) and thickening the lines designating the roads and rivers. I can carry these out also with the Layer Properties dialog (right-click on layer name→Properties). A right-click on the layer name also gives me the option to change the layer name displayed in the Legend.

ones. For this application, I am interested to know only whether the land use is agricultural or forested, not the specific type of each. Fewer colors makes my map less busy.

Navigating a Map in QGIS
Although QGIS contains several essential tools, I briefly discuss only three here: the Pan, Zoom and Identify Features tools.

QGIS has versions for Mac OS X, Windows and several variants for Linux users: source, Debian, Ubuntu Gutsy and OpenSUSE.

Post-housekeeping, the Map View in QGIS finally takes shape. I finally can recognize features such as roads and rivers, and now that the land-use types are differentiated, I easily can tell which areas are urban, agricultural, forested and so on. Figure 3 shows the end result.

To simplify visual analysis on my map, I also applied the same color to similar land-use categories. For instance, I applied the same color to two different agricultural categories as well as three different forest-related

The most essential tool for navigating around a layer is the Pan tool, the toolbar icon in the shape of a hand. If I click on that tool, I quickly can drag my map around the Map View window.

However, if I want to change the level of detail in the Map View, I must switch to the Zoom tool. Although the Zoom tool is intuitive in function, beware, for it is disappointingly unintuitive in practice for three reasons. First, the Zoom tool resides in the View menu and is not available as a

toolbar option. Second, the Zoom In and Zoom Out functions work only using the wheel of a mouse. Because I work on a laptop, I had to acquire a USB mouse just to have zooming capabilities. Third, unlike with most GIS and graphics applications, QGIS does not simply allow one to draw a box around the desired zoom-to area.

Meanwhile, the Identify Features tool is more straightforward and less cumbersome. To activate the tool, I simply press the toolbar icon designated by a mouse arrow next to the letter i in a blue circle. Then, I can navigate to any feature in the Map View window and essentially call up that feature's characteristics—that is, its entry in the attribute table. In order to select the appropriate feature, however, I must select the correct layer in the Legend. For example, if I am searching for information about a lake, I can't be on the Roads layer—the Lakes layer must be selected. Figure 4 shows how I clicked on a large lake and learned its size, elevation and name, Ford Lake.

Finding and Saving Ideal Locations
Now that I've covered the basics of GIS, found the requisite shapefiles, loaded those files into QGIS and explored basic navigation, it's time to find and record locations for my housing project. To find ideal sites where I can restore a wetland on agricultural land close to Ann Arbor, I pan and zoom around my map and toggle layers on and off.

After searching for a time, I decide to save some sites for later reference. The best way to do this is to create my own layer (shapefile). To do this, I click on the New Vector Layer icon in the toolbar, and because all I need are specific locations, I opt for a point-based shapefile. At the same time, I must build an attribute table, which I do by clicking on the Add Attribute button. I need only one string-based field, which I label Locations.

Now that I have my own shapefile, as long as that layer is selected in the Legend, I can add my own points to it by selecting the Toggle Editing tool. Once the tool is selected, the button

Figure 4. The Identify Features tool gives you detailed information on a particular feature. Be sure you've chosen the right layer in the Legend.

right next door on the toolbar, the Capture Point tool, is activated, and I can create points anywhere I choose. I create a point for each potential building site I find and add a label to each, as prompted by QGIS. I press the Toggle Editing icon once again to leave edit mode and return to normal browse mode.

Thus far, QGIS has been useful in giving me a broad perspective on natural and man-made features, as well as land-use characteristics. This is much more than what nearly every paper map or Google Earth will give me. Still, QGIS can't do everything. Unfortunately, I probably can't acquire a shapefile with current land-ownership status.

Therefore, I must utilize other resources, such as the County Clerk, in order to discover who owns which parcels. Clearly my work has only just begun.

Closing Words in QGIS
The free and open-source QGIS turns out to be an appropriate tool for projects involving land use, such as my search for a site to restore a wetland and build an eco-friendly housing development. In this project, I was able to locate the geospatial data I needed from a free geospatial data repository, load it into QGIS, tailor the data to my liking and designate a plethora of potential building sites. Besides land-use projects, you also can delve into demo-

graphic data, satellite and aerial-photo imagery, other natural and man-made features and more. Although cramming on GIS concepts and conventions was required, working with QGIS and other GIS applications, although a bit challenging at first, is extremely useful, rewarding and fun.■

James Gray is *Linux Journal* Products Editor and a graduate student in environmental science and management at Michigan State University. A Linux enthusiast since the mid–1990s, he currently resides in Lansing, Michigan, with his wife and cats.

LJ pays $100 for tech tips we publish. Send your tip and contact information to techtips@linuxjournal.com.

TECH TIP   Create a Master index.html of /usr/share/doc

Most Linux distros come packed with documentation in the /usr/share/doc directory, but rarely is there an easy way to get an overview of what's there. The following script creates a master index of all the index.html files in /usr/share/doc and outputs it as index.html in the user's home directory:

```
#!/bin/bash

input_dir=/usr/share/doc
output_file=~/index.html

cat >$output_file <<EOF
<html>
<head>
```

```
<title>$input_dir</title>
</head>
<body>
EOF

find $input_dir -iname 'index.html' | \
    sed 's/[^ ]*/\<br\>\<a href="file:&"\>&\<\/a\>/' \
    >> $output_file

cat >>$output_file <<EOF
</body>
</html>
EOF
```

—BILL ZIMMERLY

Build a MythTV Box without Breaking the Bank

How to turn your old PC from a dust magnet into a state-of-the-art media center.

P. SURDAS MOHIT

At my wedding, I received an important piece of advice from a couple whose wedding we had attended a couple years earlier: get a second TV. The idea is that while she's watching *America's Next Top Model*, I can watch hockey or mud wrestling or something. However, a much better solution to that age-old problem (and many others) is to set up a DVR system using Isaac Richards' open-source MythTV software. That way, we can watch whatever shows we want at any time.

The Basics
This article shows how to build a MythTV box on a budget and how to avoid some common pitfalls. The following is the hardware you'll need:

■ A computer: the first step is to get your hands on an old computer. You already may have one gathering dust in the basement. I bought one for $70 on Craigslist. I was cautious and chose one with a 1.6GHz AMD Athlon processor. You should be able to get by on much less by minimizing the load on your processor. Price: $0–$70.

■ Tuner card: you need a tuner card to take the digital or analog television signal and turn it into something your computer understands. The best ones for use with Linux are the Hauppauge WinTV-PVR series; the PVR-150 is a single tuner with a built-in MPEG encoder, the PVR-350 is a single tuner with built-in MPEG encoder and decoder, and the PVR-500 has two tuners with an MPEG encoder only. I bought my first PVR-150 for $85 (in Canada), including a remote and IR blaster (I'll explain what that is later). Regular prices in the US range to as low as $60. I bought my second one on sale for $25. I recommend starting with one PVR-150, and then buying another later if you feel the need. Price: $25–$85.

■ Hard drive: your computer likely already has a hard drive, but it's probably not larger than 8GB, and you'll need a bigger one. The size depends on how much of a library you plan to build. I bought a 250GB hard drive at first, then picked up a 500GB external drive later on. Price: $60–$80 for 250GB; $90–$150 for 500GB.

■ Video card: the choice of video card is very important, particularly if it's an older model. You may need to buy a new video card if your existing one doesn't have a TV-out connection. NVIDIA has the best Linux support, and you can

Figure 1. My MythTV Setup

run into a lot of problems with an older ATI card, as they haven't released proprietary drivers for them. Price: $0–$60.

■ DVD drive: if you want to watch DVDs or burn recorded programs to DVD, you need a DVD-ROM or DVD-RW drive. I definitely recommend this, as they are not too expensive these days. Price: $35.

Installation
I like Ubuntu and use it on my other computers as well, so I decided to install Mythbuntu—a MythTV-centered distribution based on Ubuntu. Unfortunately, I had trouble installing both Mythbuntu and Ubuntu itself—probably because of my RAM limitations—so I installed Xubuntu (a lightweight Ubuntu running the Xfce desktop manager) instead. The install was quite easy; however, one unexpected (but easily fixed) problem emerged. I couldn't boot after successfully installing the operating system. The GRUB bootloader would spit out "Error 18". The problem turned out to be that on older computers, the BIOS can't handle partitions larger than 8GB. So, you have to partition the disk and create a boot partition (or root partition) that is smaller than 8GB.

A word on filesystems: I used the ext3 filesystem in the initial install, but used XFS (which is better at deleting large files) on an external drive that I bought later. I wouldn't recommend

doing this unless you're familiar with XFS. I've had some issues with it—for example, it tends to become unmounted pretty frequently, requiring me to remount it, which is pretty annoying when you're trying to watch TV. The ext3 filesystem works just fine, but you should enable slow delete in the back-end settings (under General).

The next issue is connecting your MythTV box to the TV. If you have a new TV (particularly an LCD TV), you may have a VGA port in the back. If so, great—simply connect it as you would a monitor. If not, you'll need to connect the TV-out port on your video card (or PVR-350 tuner card) to the TV using an S-cable. You also need to add an entry to the /etc/X11/xorg.conf file. You may become very familiar with this file, particularly if you get a new TV or video card. Following the install, I added the following Monitor entry:

```
Section "Monitor"

Identifier "Samsung"

DisplaySize 400 300

HorizSync 35 - 50

VertRefresh 60 - 60

Option "DPMS"

EndSection
```

Your entry will depend on the make of TV you have. When I got an LCD TV, I changed it to the following:

```
Section "Monitor"

Identifier "LG 32LC7D"

UseModes "Modes[0]"

DisplaySize 1360 768

HorizSync 31.0 - 60.0

VertRefresh 60.0

Option "DPMS"

EndSection
```

I also had to add a new section:

```
Section "Modes"

Identifier "Modes[0]"

ModeLine "1360x768" 85.5 1360 1424 1536 1792 768 771 777 795
  ➥+hsync +vsync
```

```
ModeLine "1216x684" 74.2 1216 1356 1396 1650 684 704 709 750
  ➥+hsync +vsync

EndSection
```

If you installed Mythbuntu, you should be ready to go at this point. If not, use the Synaptic package manager to install MythTV and any plugins you want. Configuring MythTV the first time can be a daunting task, due to the large number of available options. You gradually will become familiar with many of them, but I'll walk you through the basic initial setup next.

Setup

MythTV is made up of a front end and a back end. The back end does most of the work: it records programs, resolves conflicts and controls access to the database. It always is running in the background. If you have a Hauppauge WinTV-PVR tuner card, recording programs takes very little effort by the processor, just a small percent. The front end is the interface for the back end. You use it to schedule and watch programs, watch and burn DVDs, play music and so on.

Figure 2. The Watch Recordings Screen of the MythTV Front End

When you install MythTV, you automatically are sent to the back-end setup. You can access it at any time by running `mythtv-setup` from a terminal. The first screen of the General setup menu allows you to set IP addresses for the back end. I recommend starting with one combined back-end/front-end machine, so just leave these as is (both local and Master should be set to 127.0.0.1 or localhost). You can skip the next screen unless you have high-definition cable. You also can skip the remaining screens for now.

Next, you need to set up your capture card. If you have a Hauppauge card, simply select the card type that mentions PVR-x50, and it should recognize the card automatically. Under Video Sources, indicate from where it should download television listings. The main one for open-source uses is Schedules Direct. Visit the Web site to create an account and set up your channel lineup.

Next, under Input connections, associate each tuner card with a listings source. Under Storage Directories, you can

indicate where you want to store recordings. The latest version of MythTV allows you to specify multiple directories, which can be useful if you have multiple hard drives (and do not use a logical volume manager).

Figure 3. The Tuner Card Setup Screen

Now you should be ready to start watching and recording TV. There are two options for interfacing with your new MythTV box: using a keyboard and/or mouse (presumably Bluetooth) or using a remote control to navigate the MythTV interface and log in remotely with SSH to perform system administration tasks, such as installing and updating software. I recommend the latter. If you don't have an HDTV, you will find it difficult to read any text or menus on the screen. Setting up the remote is easy to do if you follow one of the many how-tos available on-line. In order to allow remote login to your system via SSH, you need to install the SSH server. In Ubuntu, this is the package called openssh-server. Start it with the following:

```
/usr/sbin/sshd
```

Before doing this, make sure you are using a secure password (especially for the mythtv user that is created automatically).

If you have digital cable, as I do, you'll run into some additional complications. In order to change the channel, you need to change the channel on the cable box. You can do this with an IR blaster, which should come with your Hauppauge remote. There is an excellent on-line guide on how to set this up (see Resources). Keep in mind that you need to use the lirc_pvr150 module and *not* the lirc_i2c module; the reverse is true if you have no blaster. You need to set up the blaster right in front of the IR receiver on the cable box; I just taped mine on. When I upgraded Ubuntu on my MythTV box, the remote and blaster support broke, so you may want to set aside some time for upgrades in case this happens to you too.

MythTV isn't only a DVR. Through the plugins, it has the capability to be an all-in-one entertainment system. The

Figure 4. My IR Blaster Setup
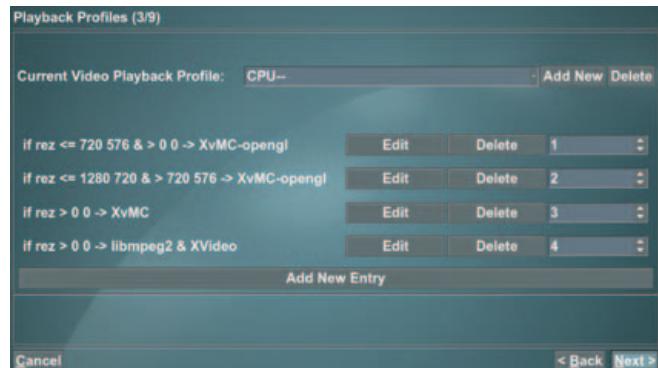
Figure 5. Enabling XvMC under TV Settings

plugins I use on a daily basis are MythMusic, MythVideo, MythArchive and MythWeb. The first two allow you to play music and videos in various formats (including DVD). MythArchive lets you burn recordings and videos to DVD, and MythWeb allows you to control your system remotely with an easy-to-use Web interface.

I dumped my music collection onto my MythTV box, and as I have a good set of speakers, I use it as my main stereo system. Watching and backing up DVDs are also a snap using MythVideo. If you have an HDTV, try downloading the HD video of the moon from the Japanese *Kaguya* spacecraft and watching it in your living room. MythArchive and MythWeb are helpful when you're not at home, because you can burn your favorite programs or movies and take them with you or download them remotely. MythWeb requires a little setup, but it's fairly easy. Make sure you enable authentication before opening it up to the outside world.

Performance

You may run into a few problems on your low-end system. The two main ones involve high CPU usage and insufficient RAM. You can operate MythTV with 256MB of RAM; however, I experienced frequent freeze-ups, so I upgraded to 512MB. You should keep an eye on how well it's performing and

consider upgrading if necessary. Another problem I experienced early on was that playing recordings or DVDs consumed such a large fraction of CPU time (70%–80%), that running other processes tended to cause the playback to become jerky. In particular, commercial flagging and database accesses at the beginning and end of recording a program produced annoying jerkiness. I resolved this problem entirely by replacing my ATI video card, which does not have a proprietary Linux driver, with an NVIDIA card. The skipping stopped almost completely when I replaced the video card, as did the seemingly unrelated problem of slow menu scrolling. CPU usage dropped to around 40%–50%.

Another benefit of NVIDIA's superior Linux support is that part of the MPEG decoding work can be delegated to the video card using XvMC (X-Video Motion Compensation), reducing the load on the CPU. To enable XvMC, go to Utilities/Setup→Setup→TV Settings→Playback. On the third screen, change the Playback Profile to CPU--. XvMC didn't kick in on mine until I deleted the top line of the profile (referring to ivtv). You can tell if it's operating because the on-screen display changes to grayscale. You also can tell because the CPU usage will go way down. The Xorg process dropped to less than 10% during playback; the sum of Xorg and mythfrontend is always less than 30%. As a result, additional processes (including creating and burning DVDs) no longer affected playback.

Satisfaction

For a pretty small sum—$85 if you get a tuner card on sale and already have a computer and up to around $500 for a multicard, multidrive system built from scratch—you can build a fully functional MythTV box. TV watching will never be the same.

Be warned: MythTV is an amazing piece of software, but it is free software that is constantly under development. Be prepared to get your hands dirty and tinker under the hood if something goes wrong or everything isn't working as you'd like. Have fun with it—test-drive different themes, tweak the settings and try the various plugins. After all, that's what Linux is all about.■

P. Surdas Mohit is a postdoctoral fellow at the Institute of Geophysics and Planetary Physics at Scripps Institute of Oceanography.

Resources

MythTV: **www.mythtv.org**

Hauppauge WinTV-PVR Tuner Cards:
www.hauppauge.com/site/products/prods_pvr.html

Mythbuntu: **www.mythbuntu.org**

How to Install an IR Blaster:
www.blushingpenguin.com/mark/blog/?p=24

Mixing Up a Generative Mobile Feast

Why chase the iPhone, when we can free the world with open mobile things?   DOC SEARLS

I've been writing, one way or another, for *Linux Journal* since 1996. Through that whole time, we've focused more aspirational attention on one receding goal than on any other: the desktop. Our progress has been asymptotic, on a curve that approaches but never arrives. And it won't, as long as we're chasing Apple and Microsoft, rather than blazing trails where both those companies can only follow.

One trail is the MID: the Mobile Internet Device. In last month's UpFront section, we reported that some of these seemed to be on track for public unveiling at the Summer Olympics in Beijing, which run from August 8–24. Because this is our August issue, it's possibly happening as you read this. LinuxWorld Expo is also happening this month, in San Francisco, California, from August 4–7.

But the more interesting trail, the one where we can score the biggest win of all, is in the mobile phone frontier. At the far edge of that is a point where the distinction between the MID and the phone verges on zero. Between here and there are many fun possibilities, and a few scary ones.

The fun trail follows a vector we can project by connecting dots drawn earlier this year. One was the purchase of Trolltech by Nokia. Another was news that Nokia was helping port Ubuntu Linux to the ARM architecture (used by the Nokia N810, among many other devices)—atop reports that the Hildon Input Method (HIM) already had migrated from Nokia's Maemo to Ubuntu Mobile. Another one—and perhaps the biggest—was news that a new broadband venture was planned by Sprint, Clearwire, Comcast, Google, Time Warner and Intel.

It's easy to project a lot of baggage onto that last one, and to get lost in a fog of vendor sports color commentary, but the Linux angle is relatively straightforward. Google's Android—a Linux-based software platform for open mobile devices (primarily mobile phones)—gets a shot at success via a coalition of partners that can produce or support many different kinds of devices and new wireless ways to connect them to the Net.

Another potential green field will be opened by 802.11y. Think of it as high-power Wi-Fi, governed by a "lite" licensing regime approved by the FCC about a year ago. With this regime, licensees pay a small fee for a non-exclusive nationwide license. They, then pay a smaller additional fee for every high-powered base station they deploy. Nobody on the receiving end requires a license. Nor do the operators. Licensing allows stations to be identifiable, which allows multiple operators politely to avoid interference. The specification covers signaling protocols both to discover and prevent interference. Proponents, including Peter Ecclesine of Cisco (from whom I learned what I just wrote), believe 802.11y holds much promise for a world where individuals and communities can connect just about any way they like, free from rule by giant carriers.

Throw OLPC Wi-Fi-based mesh networking in there too. It has lots of possibilities outside its original mission as well.

Nobody knows yet where any of this will go. But the contest, as usual, will be between freedom and control, open and closed, proprietary and public domain. The difference this time will be the playing field, which will be inside companies as well as out in the marketplace. On one side will be those that advocate locked-down appliances with a reduced subset of features that advantage only the company selling the devices and its partners in the equipment, network or content production and distribution businesses. On the other side will be those who understand that markets grow fastest on generative foundations—ones that support, with little or no bias, a maximum variety of uses.

The best example of a locked-down mobile appliance is the iPhone. Although it's beautiful, useful and ground-breaking, the iPhone also will remain closed to all but the applications Apple approves, many of those for Apple's own purposes. Apple will continue to pioneer here, but with a closed and controlling bias.

We have no equivalent best examples of generative mobile devices. Nokia's N series (770 800, 810...) is an early prototype, and there still is no widespread cellular (or cell-like) system that provides uncrippled Net connectivity alongside VoIP service.

But, generative mobile is the goal. To mix metaphors a bit, think of the means as a recipe. The ingredients are there. The table is set. What we need now are some chefs to mix the ingredients together and produce some meals that are appetizing to the marketplace.

As I explained in this column last month, the best examples of recent generative inventions are the white-box PC and the Internet. The PC started changing the world in the 1980s, and the Internet started doing the same in the 1990s. Our hope now is that the world will be changed in the 2000s by generative mobile devices that connect to an equally generative Internet.

The one sure thing is that Linux will be the base ingredient. The rest is up to the chefs.■

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

The choice is yours. Just be sure to choose the one you can trust to manage and support your IT infrastructure for you. ← And that would be Rackspace.

- 3-time Microsoft Hosting Partner of the Year – 2007, 2005, 2003
- Industry-leading Service Level Agreements and Guarantees
- SAS70 Type II Audit for Intensive Service Level Customers

experience
fanatical support

rackspace.com/linuxjournal
888-571-8976