AN SSC PUBLICATION

FREE NAVIGATION SYSTEM PLOTS YOUR FRIENDS TOO: **GPSDRIVE**

LINUX JOURNAL™

Since 1994: The Original Magazine of the Linux Community • APRIL 2005

Failover for your servers:
STEELEYE LIFEKEEPER

SHELL SCRIPTS IN SPACE

(Earth too!) Special "cookbook" solutions issue

SECURE WIRELESS LAN ON A BUDGET

Make your existing wi-fi hardware safe with open-source tools

VIRTUAL THEATER

on your Web browser

ZERO-CLICK PRESENTATIONS

Take your ideas from Web to screen automatically with scripting

USA $5.00 CAN $6.50
www.linuxjournal.com
04
0 71486 03102 4

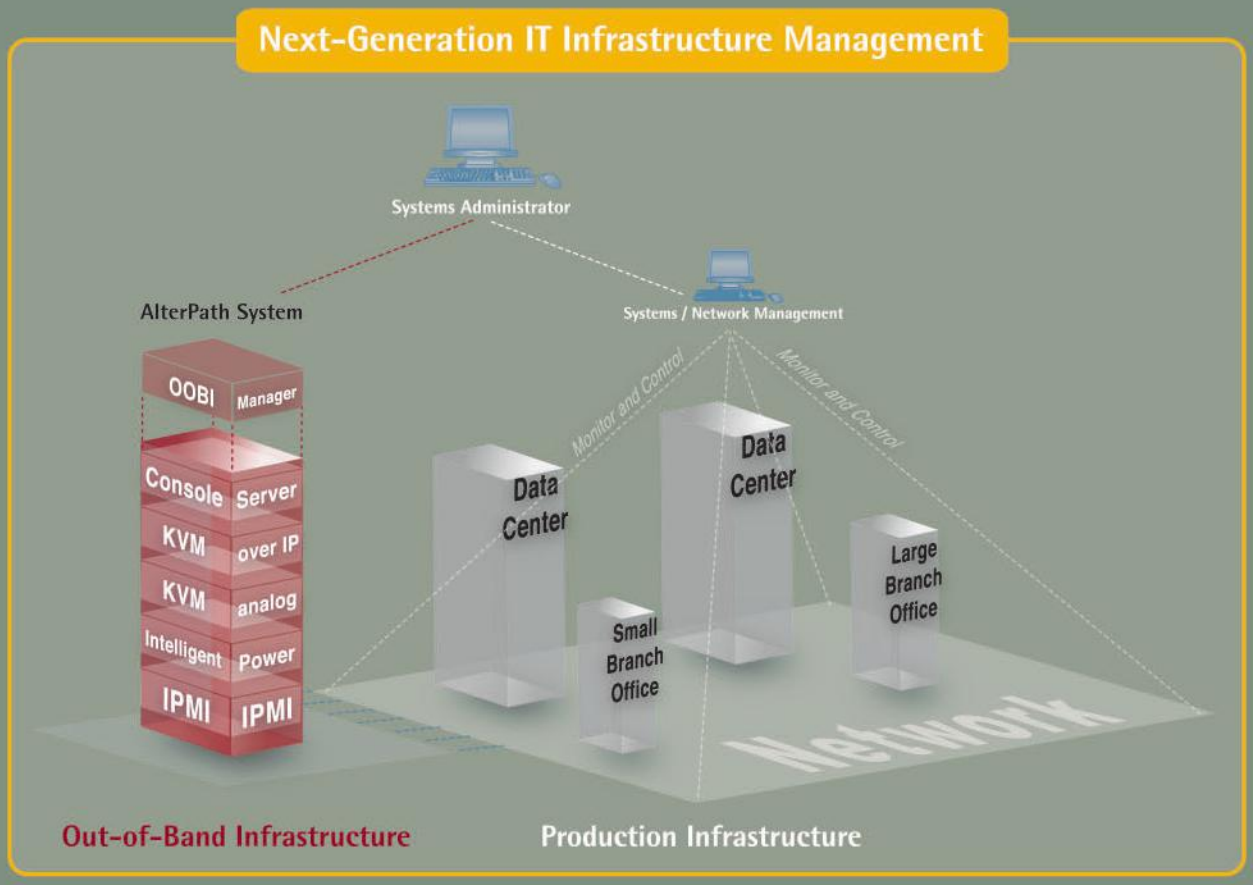
LINUX JOURNAL · COOKBOOK · GpsDrive · bash · Unison · UpStage · LifeKeeper · APRIL 2005 ISSUE 132

Let's talk OOBI

Next-Generation IT Infrastructure Management

OOBI, or Out-of-Band Infrastructure, is Cyclades' unique approach to enhance your IT infrastructure management capabilities. It integrates management of serial ports, KVM, KVM over IP, intelligent power distribution and IPMI devices in a secure, consolidated management solution for remote IT infrastructure administration. It complements system management tools like HP OpenView, IBM® Tivoli®, BMC PATROL® and CA Unicenter®.

An OOBI can cut costs and boost operational efficiency and productivity by minimizing the need for redundant equipment and personnel and, in most cases, eliminating the need for crash cart runs or remote visits to restore IT assets that have become disconnected from your production infrastructure.



To receive a **FREE** white paper on OOBI, Out-of-Band Infrastructure, visit us at **www.cyclades.com/oobiwp**

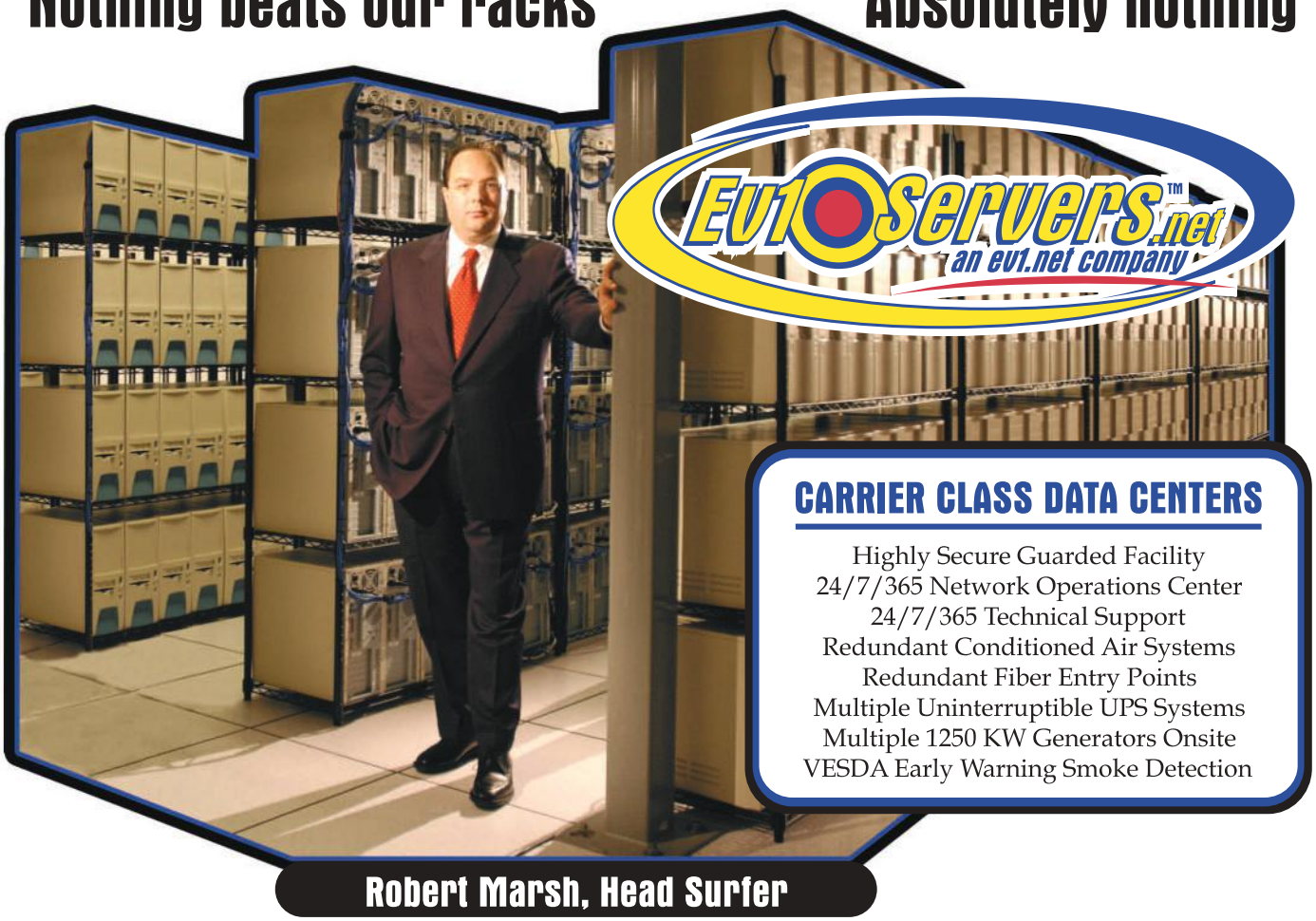
Over 85% of Fortune 100 choose Cyclades.

www.cyclades.com/lja

1.888.cyclades · sales@cyclades.com

cyclades®

Nothing beats our racks Absolutely nothing



Robert Marsh, Head Surfer

CARRIER CLASS DATA CENTERS

Highly Secure Guarded Facility
24/7/365 Network Operations Center
24/7/365 Technical Support
Redundant Conditioned Air Systems
Redundant Fiber Entry Points
Multiple Uninterruptible UPS Systems
Multiple 1250 KW Generators Onsite
VESDA Early Warning Smoke Detection

START YOUR OWN WEB HOSTING BUSINESS TODAY!

from **$299*** *Instant Activation!*

Dedicated Server

Dual Xeon 2.4 GHz
2 GB RAM • 2 x 73 GB SCSI HD
Remote Console • Remote Reboot
2000 GB Monthly Transfer Included

Over 20,000 Servers!

1-800-504-SURF | ev1servers.net

PLESK7 RELOADED
Preferred Control Panel

IP Compliant. Price subject to change. Quantities Limited.
*Per month. Set-Up fees apply. See web site for complete details.

LINUX JOURNAL
APRIL 2005 ISSUE 132

COVER STORY

FEATURES

INDEPTH

EMBEDDED

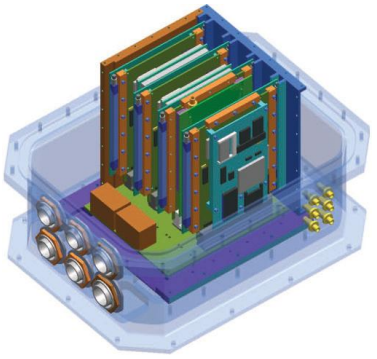
TOOLBOX

COLUMNS

REVIEW

COVER IMAGE COURTESY OF CHRISTOPHER HUFFINE

DEPARTMENTS



This sensor system started life on an unmanned airplane but became part of the TacSat-1 satellite project (page 40).

NEXT MONTH

DEVELOPMENT

Cryptography means a lot of number-crunching—but one processor vendor is putting support for the Advanced Encryption Standard in hardware. Michal Ludvig will discuss the code that uses processors with VIA PadLock technology to speed up kernel and OpenSSL crypto by a factor of 60.

Mozilla Sunbird is a calendar application that lets you share calendars company-wide using the Internet-standard iCalendar. Reuven Lerner will take us on a tour of the standard, the application and how to make it work for you.

Thinking about inventing a new programming language? Tom Tromey will cover how to develop a new front end to the GNU Compiler Collection. Make your favorite language use the optimizations and hardware support of GCC without re-inventing the whole compiler.

The Linux of Satellites

A hardware design from an unmanned aircraft project, along with Linux and other free software, got this project done quickly at a bargain price. **BY DON MARTI**

By the time you read this, TacSat-1 might already be in orbit. We're all in suspense as our cover project prepares to ride the first launch of the new SpaceX Falcon-1 launch vehicle from Vandenberg Air Force Base in California.

TacSat-1 aims to do for task force commanders what commodity hardware and open-source software can do for business managers. With the new satellite capability, commanders in the field will be able to track individual enemy radars and transmitters, and get visual and infrared imagery, with minimal bureaucracy.

It's a high-profile space version of what's been happening on Earth for a long time. Information technology is becoming faster and more responsive to real business needs. Road maps, customer-hostile business models, and anything else that gets in the way are obsolete. In this issue, we're celebrating the projects that don't merely get the job done more cheaply and reliably, but those that open up new information technology avenues for people who otherwise would be locked out by pointless restrictions.

Have a look at Charles Curley's "Finding Your Way with GpsDrive" on page 50. Unlike a monolithic GPS mapping product, you can combine your choice of maps with public GPS data to get the navigation you need. Yes, you can cruise for wireless Net access and plot it. Please be nice. Meanwhile, if you're worried about other people getting on your wireless network, Mick Bauer has some good news for you in the form of a new security standard and a way to integrate Wi-Fi security with your existing infrastructure. Get started with WPA on page 36.

Paul Barry had a problem converting his data into the promised Microsoft PowerPoint slides. Fire up the "productivity" application? No thanks—not enough time. Run everything through a script and OpenOffice.org, and the job's done and the carpal tunnels in Paul's mouse hand are safe, see page 58. Keeping up with vendors who try to lock in customers with undocumented formats is tough. Thanks, OpenOffice.org.

Sometimes you need to convert a system to Linux, or to a special-purpose Linux distribution, temporarily. On page 54, Daniel Barlow gets you started with modifying Knoppix to create your own personal live CD. Render Farm? BZFlag Zone? The choice is up to you.

Our Web columnist, Reuven Lerner, is celebrating his 100th column (page 22). Thanks, Reuven, for breaking through the wild and woolly mess of the Web to bring us ideas and technology that really work, for Linux users and everyone else. There's plenty of other great technical stuff in this issue, too. But even if you don't use any of the specific advice—which I doubt, considering we could all use a couple more shell tricks, as Prentice Bisbal brings us on page 76—remember the reason why all this stuff is so great. With Linux and the other software we cover, you have the freedom to make your project happen the way you want. See you at the launchpad.

Don Marti is editor in chief of *Linux Journal*.

LINUX JOURNAL

APRIL 2005
ISSUE 132

SUBSCRIPTIONS

E-MAIL: subs@ssc.com • URL: www.linuxjournal.com

PHONE: +1 206-297-7514 • FAX: +1 206-297-7515

TOLL-FREE: 1-888-66-LINUX • MAIL: PO Box 55549, Seattle, WA 98155-0549 USA • Please allow 4–6 weeks for processing address changes and orders • PRINTED IN USA

USPS *LINUX JOURNAL* (ISSN 1075-3583) is published monthly by SSC Publishing, Ltd., 2825 NW Market Street #208, Seattle, WA 98107. Periodicals postage paid at Seattle, Washington and at additional mailing offices. Cover price is $5 US. Subscription rate is $25/year in the United States, $32 in Canada and Mexico, $62 elsewhere. POSTMASTER: Please send address changes to *Linux Journal*, PO Box 55549, Seattle, WA 98155-0549. Subscriptions start with the next issue. Back issues, if available, may be ordered from the Linux Journal Store: store.linuxjournal.com.

Rackspace — Managed Hosting backed by Fanatical Support.™

Servers, data centers and bandwidth are not the key to hosting enterprise class Web sites and Web applications. At Rackspace, we believe hosting is a service, not just technology.

Fanatical Support is our philosophy, our credo. It reflects our desire to bring responsiveness and value to everything we do for our customers. You will experience Fanatical Support from the moment we answer the phone and you begin to interact with our employees.

Fanatical Support has made Rackspace the fastest-growing hosting company in the world. Call today to experience the difference with Fanatical Support at Rackspace.

LINUX JOURNAL READERS' CHOICE 2004

Thanks for
honoring us with the
2004 Linux Journal
Readers' Choice Award for
"Favorite Web-Hosting Service"

rackspace™
MANAGED HOSTING

1.888.571.8976 or visit us at **www.rackspace.com**

Ssh! Everybody Look Professional!

People "in the know" understand that Linux is perfectly appropriate for the enterprise. However, there are many circles that still think of Linux as a hobbyist project. Consultants such as myself face an up-hill battle when pushing Linux-based solutions. I believe our job is made more difficult when one of the few Linux-focused periodicals actually to make it to the local magazine racks prominently displays the powerful operating system's ability to act as a Web-based cat feeder. I appreciated the article, but did it have to go on the cover?

--
Jeremy Cherny

Are you trying to get us in trouble with the "where's the fun" guy? Fun technology attracts the new developers and projects, and non-fun technology dries up and blows away.—Ed.

Happy Birthday, Patrick

As a subscriber of *Linux Journal* since 1994, I have been following all the great pictures of newborns, who get their first introduction to Linux on the pages of your great publication. When my son Patrick arrived on October 23, 2004, I knew I had to start him off with penguins as soon as possible. So, on his first-month birthday, my wife snapped this picture to show that a new Linux hacker is on his way to help in the Open Source community.

--
Piotr Trzeciak

64-Bit Porting, Please

As I am currently hacking my way through this myself...I would really like to see an article on building software (compiling) for the AMD64. There are certain pointer symantics and sizing issues that need to be dealt with, and I have yet to find a good source on "porting" to 64 bits.

--
Peter

There's a bunch of 64-bit wisdom scattered around the Net and in project source code. We'll look for someone to write the article for you.—Ed.

Laptop Comparisons, Please

In the January 2005 issue, you have a nice review of the HP laptop. I do not mean to be too critical, but it seems to me that you have given us half a loaf. We do not buy such things in a vacuum. There are other Linux boxes out there, such as from Emperor or even Lindows, Wal-Mart, sub300 (ugh), etc.

It would help me a great deal if a review would describe not only the object under discussion, but also include some comments about whether it is "better than", in almost any way you choose to evaluate it, some other machine. Is is a better buy than the equivalent box from Emperor? Where does it fit in, in the long scale of very cheap to very expensive, versus quality. I think the reader would be better served with such information, even if it is only your best guess. Because (I hope) you have a lot better data base to go on than I do. Many thanks for a good magazine.

--
tony

Get Your Pre-Ban HDTV Cards

Thanks for the heads up on the DRM fiasco for HDTV. I believe pcHDTV is now shipping version 3000 and will continue to do so until the 30th of June, 2005, without the DRM flag. Can you confirm this or do you already have confirmation of this?

--
Kevin R. Battersby

Watch for an update item on this next issue.—Ed.

Enough Kids—Puppy Break!

After hinting for the last four and a half years of our marriage my wife finally conceded, and Charlie is the result. Immediately upon agreeing we would take the pup, my wife went to work using some fabric endowed with a Tux look-alike and made a few goodies for our new puppy. Hopefully this finds you the editor in good health, as well as the rest of the *LJ* staff. Here is to what may become of Linux in 2005, cheers!

--
James

31337 m1773nz!

My son Graeme is quite a Linux fanatic. One of his good friends made him these mittens for Christmas this year. I thought you might want to see this upcoming fashion trend for what every "cold" Linux user should be wearing.

--
Eric

Unicode Question

Thanks for your article in the December

How Stable is Your HPC Solution?

Aspen Systems Inc.

You can't afford downtime, in your business downtime means lost data and lost revenue.

An Aspen Systems' high performance computing cluster built on the stability and flexibility of the AMD Opteron™ Processor consistently connects you to the power you need to run your 32-bit and 64-bit applications faster and more efficiently than ever before.

We offer 100% custom solutions designed from the ground up to meet the most demanding applications: yours.

To design your custom cluster solution contact Aspen Systems at:

303.431.4606

sales@aspsys.com

or visit:

www.aspsys.com

2004 issue of *Linux Journal* about aggregating feeds, I really enjoyed it. It's probably months since you wrote the article, but I've only just got around to reading it over the Xmas break! I don't know Python, but managed to tinker with your code and get my own feeds page going (snowfrog.net/myfeeds.html).

I'm getting an error when I syndicate some sites, such as safari.oreilly.com/rss, and I don't know how to fix it. Any pointers? Obviously, it involves stripping out non-ASCII chars, or changing the codec to Unicode, but I don't know how to do that (yet):

```
UnicodeEncodeError: 'ascii' codec
can't encode character u'\xae' in
position 66: ordinal not in
range(128)
```

This occurs when (for example) I do a `sys.stderr.write(mystring)`. Thanks for any help you can give.

--
Sonia Hamilton

Reuven Lerner replies: *I'm glad that you enjoyed the article! And yes, I normally write columns about 3–4 months before they are printed—but I do remember writing about feedparser and aggregating feeds.*

Hmm, I'm a bit surprised that something is choking on Unicode characters. That shouldn't happen, should it? And for feedparser to be choking is even weirder, because I was sure that it could handle Unicode just fine. But the problem isn't the Unicode string. Rather, it has to do with the fact that the Unicode string isn't being translated into a non-ASCII codec, which is what you guessed. For example, consider the following:

```
>>> print u'\xae'
Traceback (most recent call last):
  File "<stdin>", line 1, in ?
UnicodeEncodeError: 'ascii' codec
can't encode character u'\xae' in
position 0: ordinal not in
range(128)

 >>> print u'\xae'.encode('utf-8')
®
```

So you (or the feedparser source; it's not clear if the problem is in code that you wrote or in the feedparser code) probably should include a call to encode, indicating the resulting codec.

I haven't read it very carefully, but the feedparser documentation includes a description of encoding systems. It might well be that you're being bitten by something there (www.feedparser.org/docs/character-encoding.html). I hope that this helps! Please let me know if you have any further questions.

Hey, Puffins Don't Count!

I have been using Linux since 1998 and Red Hat 5.2. My son has liked penguins since 1996. He has quite a collection of stuffed penguins including a few Tuxes. Please excuse the occasional puffin. Here he is pictured with our ThinkPad running Fedora 3.

--
Stuart Boreen

GPG Fingerprints

I read you rolled out GPG for everybody at SSC. How about adding the key fingerprints in the journal itself as an example to give validity to the keys. You list all the collaborators on page 4 with the e-mail addresses, this would be a nice spot to add the fingerprints. The only drawback would be the space it takes. Better: create a master key, sign all the keys with the master, and print the fingerprint of the master. Keep up the good work with the magazine, I am now the owner of 50cm of magazines. And my best wishes for 2005 to you and the whole team.

--
Erik Ruwalder

Great idea. We'll ask our IS department to create a company master key and sign all our keys with it.—Ed.

Dependency Hunting, S'il Vous Plaît

I appreciate Marcel Gagné and his monthly column, particularly because it is the only one in *LJ* that I can consistently understand. But, why does he feel it necessary (or useful) to repeat the same five-step build process for every piece of software? It's only useful if it compiles with no problems—and we all know that never happens. Anyone capable of hunting down dependencies certainly knows the build process.

I have a proposal for Marcel and François: how about a column devoted specifically to the compile process? I am particularly interested in knowing about common dependency issues, common paths to specify, and why and how to install dependencies in a different directory so they can coexist with other, default versions of the same software. I am running Xandros 2.0, which uses an older version of KDE (and many other things). I would love to be able to install software that requires KDE 3.3, but upgrading to that would certainly wreck my OS. There must be a way to install dependencies in parallel, with the more advanced versions to be used only by the programs that I specifically point to them, but I have no idea how to go about this.

--
Derek Croxton

Letters to the Mainstream Media

I was looking in the local rag, the *New York Daily News* and saw this in the editorial column:

> Microsoft Windows is a terrible product. If Windows were a commercial aircraft, the FAA would ground it. If it were a prescription drug, the FDA would ban it. If it were a horse, you'd shoot it. Every new Windows release is miles worse than the one before it. Every fresh patch and tweak crashes your system more and more desperately. Microsoft Windows wants to kill you.
>
> But yet we're all stuck with it. We all depend on it, completely and absolutely and utterly.

Designed for dependability.

The Intel® Xeon™ Processor offers enterprise-proven reliability, plus performance features like Processor redundancy, enhanced server management and higher bandwidth network redundancy. ZT Servers with the Intel Xeon Processor can keep your business up and running –and productive. All day, every day.

ZT OPTIMUM 5U Storage Server X9407

Intel® Xeon™ Processor 3 GHz
Upgradeable to Dual Intel® Xeon™ 3.60 GHz
(2MB L2 Cache Processor)
Intel® E7520 chipset Server Board
1.0GB ECC Registered DDR333 SDRAM (Up to 16GB)
24 x Seagate® 300GB SATA/150 7200RPM (8MB Cache) Hard Drive (Total 7.2 TB Storage)
24 x 1" SATA Hot-Swappable Drive Bays
Dual 12 Port 3ware 24 Channel Serial ATA Controller (RAID 0, 1, 5, 10 , JBOD Support)
1.44MB Floppy Drive
16x DVD±RW & CD-RW Combo Drive
ATI Rage XL SVGA 8MB PCI Graphics
Dual Intel® 10/100/1000 Gigabit Network controller
5U Rackmount Server Chassis w/950W Redundant Power Supply
3-Year Limited Warranty

$9,149

- Onsite service available
- Please call for latest pricing and discounts on quantity orders
- Your Ultimate Solutions Provider
- Aggressive Price

ZT Optimum 1U Server X9403

Intel® Xeon™ Processor 2.80 GHz
Upgradeable to Dual Intel® Xeon 3.60 GHz
(2MB L2 Cache Processor)
Intel® E7320 chipset Server Board
1.0GB ECC Registered DDR333 SDRAM (Up to 8GB)
2 x Seagate® 200GB Serial ATA/150 (8MB Cache) 7,200rpm Hard Drive
(IN RAID 1 CONFIGURATION)
2 x Hot-swap Serial ATA/150 Drive Bays
1.44MB Floppy Drive
52x32x52 CD-RW & 16x DVD-ROM Combo Drive
ATI Rage XL 8MB PCI Graphics Controller
Intel® 10/100/1000 Gigabit Network Controller
1U Rackmount Chassis W/350W Power Supply
3-Year Limited Warranty

$1,369

ZT Enterprise 2U Server X6653

Intel® Pentium® 4 Processor 3.20 GHz
(Upgradeable to Intel® Pentium® 4 Processor 3.60 GHz)
Intel® E7221 chipset Server Board
512MB DDR2 533MHz SDRAM (Up to 4GB)
2 x Seagate® 200GB Serial ATA/150 (8MB Cache) 7,200rpm Hard Drive
(IN RAID 1 CONFIGURATION)
1.44MB Floppy Drive
16x DVD-ROM
Onboard Integrated High Quality Server Graphics
Dual 10/100/1000 Gigabit Network Controller
1U Rackmount Chassis W/350W Power Supply
3-Year Limited Warranty

$1,099

ZT Optimum 2U Server X9405

Intel® Xeon™ Processor 3 GHz
Upgradeable to Dual Intel® Xeon™ 3.60 GHz
(2MB L2 Cache Processor)
Intel® SE7320VP2 (E7320 chipset) Server Board
1.0GB ECC Registered DDR333 SDRAM (Up to 12GB)
12 x Seagate® 300GB SATA/150 7200RPM (8MB Cache) Hard Drive (Total 3.6 TB Storage)
12 x 1" SATA Hot-Swappable Drive Bays
3Ware 12 Channel Serial ATA Controller (RAID 0, 1, 5, 10 , JBOD Support)
ATI Rage XL SVGA 8MB PCI Graphics
Dual Intel® 10/100/1000 Gigabit Network Controller
2U Rackmount Server Chassis w/460W Redundant Power Supply
3-Year Limited Warranty

$4,999

ZT Optimum 3U Storage Server X9406

Intel® Xeon™ Processor 3 GHz
Upgradeable to Dual Intel® Xeon™ 3.60 GHz
(2MB L2 Cache Processor)
Intel® E7520 chipset Server Board
1.0GB ECC Registered DDR333 SDRAM (Up to 16GB)
15 x Seagate® 300GB SATA/150 7200RPM (8MB Cache) Hard Drive (Total 4.5 TB Storage)
15 x 1" SATA Hot-Swappable Drive Bays
Dual 3Ware 8 Ports 16 Channel Serial ATA Controller (RAID 0, 1, 5, 10 , JBOD Support)
ATI Rage XL SVGA 8MB PCI Graphics
Dual Intel® 10/100/1000 Gigabit Network Controller
3U Rackmount Server Chassis W/760W Triple-Redundant Power Supply
3-Year Limited Warranty

$5,999

(984-7687)

866-ZTGROUP

http://www.ztgroup.com/go/linuxjournal

Please contact our system specialists for any customize systems, latest pricing & quantity orders via email to Shopper@ztgroup.com or call (866)984-7687

Special Discount for Education, Government & Corporate

Quality Assured • 24x7 available • Lifetime Tech Support

*** Price subject to change without notice**

So, I wrote a reply to the editor:

> What to do about "broken" Windows?
>
> Plenty. There is a whole family of free operating systems such as GNU/Linux and BSD available to all on the Internet or from your local computer users group.
>
> Try Knoppix on CD. It comes with everything you need including word processing and e-mail software, real games, self-installing network software and you don't even have to install it on your computer.
>
> Microsoft does not want you to know about that.
>
> Wonder why? It is simply better. Try it (www.knoppix.net).

--
Adam Vazquez

Kernel IPSec History

The article "Linux VPN Technologies" [February 2005] discusses IPSec and its availability in the 2.6 series kernel. It states that FreeS/WAN is available in the kernel, when in fact the 2.6 kernel uses a port of the KAME IPSec stack (www.kame.net). The KAME stack was originally developed for the BSD variants and is very mature. The utilities for interacting with this stack, called ipsec-tools, can be found at ipsec-tools.sourceforge.net. I'm successfully using the 2.6 IPSec stack for a custom wireless access point using hostap. Thanks for the excellent work.

--
Peter Johanson

FreeS/WAN and OpenS/WAN were never official parts of the kernel; some distributions did include them.—Ed.

More Innovative Apps, Please

I have been a Linux advocate for many years and continue to marvel at the progress it has made. Several leaps have brought Linux much more into the mainstream of business and even home users in recent years.

It seems that many applications are not innovative, but just copies of other ideas from other platforms. While it is important that critical areas be filled with appropriate applications in order to make Linux viable for users, it is also important to innovate. That being said, will users move to Linux for the same applications that they can get on other systems? Probably, because of cost savings. But, more users would move faster to Linux if there are applications that are innovative.

I remember in the mid-1980s when a small company was able to capture nearly 25% of the PC market despite having more expensive products. A simple change to the user interface that used graphics in place of menus made all the difference.

So, as I flip thought the last several issues of *Linux Journal*, I have yet to see many innovative applications. As more people become interested in Linux for its reliable and highly customizable features, will there be an incentive to switch other than cost? Without innovative applications, it could relegate Linux to remain in the back office in the hands of the techies.

--
John Irey

NLD as Seen by a Novell User

Just before reading the latest issue [February 2005], I was thinking to myself that *LJ* really hasn't yet acknowledged that Novell is now one of the major players in the Linux world. And then there was your review of Novell Linux Desktop (which as I'm sure others have told you by now is NLD not NDS). As a longtime Novell user and a longtime Linux user I was happy that Novell took the steps it did. Like everyone else I was keeping my fingers crossed that they wouldn't screw up like they did when the acquired WordPerfect and sold UNIX to SCO. So far they haven't made any major blunders.

Initially, the emphasis was on a good server kernel to replace NetWare, which although still quite capable, isn't as good as Linux in many areas. Of course Novell just couldn't resist competing with Microsoft by pushing Linux on the desktop. NLD isn't bad, but it offers little advantage over any of the other distributions. Novell is positioning NLD as a business desktop.

In their rush to get NLD out the door, they didn't get all the pieces in place to integrate NLD into an existing Novell network, so most established NetWare shops aren't finding it very useful either. I don't even think ncpfs (needed to mount NetWare volumes) came installed by default. I know GroupWise didn't, even though they have a pretty good Linux version. Evolution 2 is the default mail program but none of the GroupWise hooks are working yet. Those are waiting on the next version of GroupWise due out mid-year. The reset of the integration part is waiting on Novell Open Enterprise Server (OES), now in beta. Supposedly, it will have a true NetWare client akin to the the Win32 client. Time will tell.

Novell is a major player in the Linux world now, and we should accept that fact and work with them. They are very open-minded right now and will benefit from interaction with people that have been around Linux a lot longer then they have. I'd encourage you to attend Brainshare this year. Linus was there last year. As always, *LJ* is great and continues to get better. Thanks for the good work.

--
Paul

Cruelty-Free Advertising, Please

I used to be a subscriber to *Linux Magazine* until the Microsoft ads started appearing a few years ago. I was sickened and stunned. I stopped dead still in my tracks with feelings of anger and realizations of betrayal. I really used to look forward to the monthly delivery of *LM*, but I was left staggered and emotionally confused at the sight of Microsoft's ugly and sudden appearance. There was a sinister happiness about the ad, and it felt very much as if the magazine I was holding contained a plague that began infecting my hands. I could feel the hate and cruelty making its black way up both my arms...going for my soul...trying to turn me into its Golem. Even as the magazine slammed against the wall across the room, I still felt sick, and angry, and sad, and betrayed. It was a really tough day for me.

Linux Journal is the only Linux magazine I subscribe to now. If something should happen to you guys...where else could I go?

--
Tony Freeman

Security Blanket?

My home office is in the basement of my Wisconsin home. It gets rather chilly here in the middle of the winter, so my wife

Cyclades AlterPath™ Manager for

Cyclades IPMI Management

Enjoy the magic

Cyclades' AlterPath™ Manager is the first secure enterprise IPMI manager to boost operational efficiency and productivity with industry leading features:

- **Vendor-independent IPMI management**
- **Enterprise security and authentication (SSH v2, LDAP, RADIUS, TACACS+, Kerberos)**
- **Centralized access, control and auditing**
- **Event notification and alarms**
- **Scalable to 5000 devices and 256 simultaneous connections**

- **Remote IT infrastructure administration**
- **Centralized OOBI™* management**
- **Remote incident resolution**
- **Web-based access**
- **Remote logging and monitoring**
- **Supports IPMI 1.5 and 2.0**

We've worked our magic. Now you can work yours.

To receive a **FREE** white paper on IPMI, visit us at www.cyclades.com/ipmiwp

*OOBI, or Out-of-Band Infrastructure, integrates management of serial ports, KVM, KVM/IP, intelligent power distribution and IPMI devices in a secure, consolidated management solution for remote IT infrastructure administration.

Over 85% of Fortune 100 choose Cyclades.

www.cyclades.com/ljb

1.888.cyclades · sales@cyclades.com

cyclades

surprised me with a blanket she made to help keep me warm as I use my Linux workstation.

A mid-winter kite flying event called "Kites on Ice" was held in Madison, Wisconsin on Lake Monona. My son captured this view of some penguin kites flying high above the snow and ice-covered lake.

I have been enjoying *Linux Journal* since Issue 36 (April 1997) and will continue for a long time to come (my subscription runs until February 2012. (I bought into the 100 issues for $100 offer a while back.) Keep up the good work. Linux forever!

--
paul

root Password Management

May I interject a thought regarding the comments made in the January 2005 *LJ* [Best of Technical Support, "Distributing /etc/shadow", p. 68]? While possibly more than was originally requested, one of the other options available instead of constantly changing the root password would be to use the SecurID system (and ACE Server) that is sold by RSA Security. It gives you a variation on "one-time passwords" and in a lot of cases can satisfy the MIL-Spec that requires rotating the root passwords. However, in practice, locking out the root password and using `sudo` for everything (which can also use SecurID) is a much smarter idea. It provides auditing as a side benefit.

--
Michael C. Tiernan

Open Access to Archives, Please

Please put my vote in the "liked open access better" category. I subscribe to make *Linux Journal* possible and to have a hard copy in my hand every month. It doesn't bother me that someone else may be getting it free off the Web. I think that requiring a subscription to get full content is passing up the opportunity to provide a service. Let me give you an example.

I've been asked to do a piece for IEEE Software. So while researching past issues to find the appropriate tone for the audience, I found that there's a lot of good stuff there that I (and others like me) don't have access to because we don't get the subscription. As a result, we're less informed than we might be. Publications that do that may be protecting their copyrights and business model at the cost of their community being less informed. Even as a subscriber I'd be happier if the average Linux user was better informed.

The last time I checked, the European Linux magazines offered free access to content that was over a year old. If allowing access to older content (for some small value of "older") will satisfy the original objectors, I can live with that.

I will keep my subscription whether you choose to change the subscriber-only policy or keep it. In either case, I get what I want and what I paid for. I think that open Web access to content is a valuable extra to me

Photo of the Month: Penguin Visit

My wife and I returned from Antarctica in December 2004. You'll be happy to know that the penguin population is thriving and (at least while we were there) gentoos were the most populous distribution!

--
William E. Shotts

Photo of the month gets you a one-year subscription or a one-year extension. Photos to ljeditor@ssc.com.

Your corporate colors and logo

Engineered to your specifications

"From testing and design to nationwide deployment, the service from MBX has been first class."

- Ray S., Crate and Barrel

MBX is the leader in custom appliances. Many premier application developers have chosen MBX as their partner because of our experience, flexibility and accessibility. Visit our website or better yet, give us a call, our phones are personally answered by experts ready to serve you.

MBX RP-1010 LCD / Custom Branded Appliance

- Intel® Pentium 4 Processor® at 2.4E GHz
- 1U Short Depth Rackmountable Chassis
- 256MB PC2700 DDR Memory
- 512MB Solid State Flash Drive
- Dual Onboard Gigabit NIC's
- Front Panel LCD with 6 Button Control
- Custom Branded With Your Logo
- 3 Year Warranty
- Worldwide Deployment and Support
- Custom Branded Packaging Available
- Configurations in 2U and 4U Available
- Custom OS and Software Install
- Custom Chassis Color Available
- No Minimum Quantity Required

$999 or lease for **$31**/mo.

MBX™ MOTHERBOARDEXPRESS

www.mbx.com/oem

1.800.939.6191

intel inside pentium 4

and the community.

--

George Koharchik, Speaking only for myself

Trip to Thailand

I took this picture of my wife, Ja (center), and her two twin sisters, Apple (left) and Cherry (right), on Christmas Day 2004, in Rayong Thailand. The T-shirts were from the Picn*x 13 Linux picnic in Sunnyvale, California last August (donated by Google).

We were on nearby Koh Samed (Samed Island) the next day when the tsunami hit. Luckily, both Rayong and Koh Samed are in the Gulf of Thailand, not on the Bay of Bengal. We noticed nothing more than a 3–4 foot surf, slightly larger than normal.

--

Drew Bertola

What Was the Bug?

I am very curious about the "Horrible Bug" referenced in the diff -u section of the February 2005 *Linux Journal*. I have just purchased SuSE Linux Professional, Release 9.2, which contains the 2.6.8 kernel. A description of the bug would help to determine if my specific system would in any way be impacted.

--

Richard Hathaway

Zack Brown replies: *The bug was with NFS. Entering a mounted NFS directory would result in an OOPS under the 2.6.8 Linux kernel. Only folks using NFS would experience a problem.*

Regarding SuSE Pro 9.2, you can put your mind at ease, the fix is included. All Linux distributions, SuSE, Debian, Red Hat and the rest, apply various patches to their kernels before release. In fact, in recent days the kernel developers have come to rely on vendor patches more explicitly, as a crucial element of the stabilization process. The 2.6.8 kernel included in the SuSE Pro 9.2 release is not a "true" 2.6.8 kernel, it is more like a 2.6.9-rc2 kernel with further additions. One of these additions clears up the NFS oops problem found in the official 2.6.8 kernel.

Question on Serial Ports

I particularly enjoyed Chris McAvoy's article in the January 2005 issue entitled "How I Feed My Cats with Linux". I do have one question though. He makes the point that the BASIC Stamp uses a nonstandard serial port and specifically points out that Parallax's method makes two-way communication difficult. This seems a valid reason for replacement, except that I can't find any instances in the sample code where two-way communication is actually used. Did I miss something? For the purpose of the given example, wouldn't the onboard serial suffice? I do appreciate that this is but one example of the possibilities of this kit and can see where two-way communication would be useful, just not in this case.

I would like to commend this article and ask for more like it, as I am interested in data acquisition and digital I/O controls for some future projects that I am planning, and currently *LJ* is my only link to the computer world until 2008. Which brings me to a final question for the subscription department. I am a longtime subscriber, but the last couple of years, I have been receiving my *LJ* while incarcerated in a California State prison, and I wonder, are there any other inmate subscribers? I've run into very few computer geeks like myself in prison, and not a single Linux enthusiast, so my curiosity is piqued.

Thanks to the entire *LJ* staff for their hard work in putting out a fine publication.

--

Jason Shelton

Chris McAvoy replies: *thanks for writing. You're right about not necessarily needing the MAX232 for one-way serial communication to the STAMP. Given the way we're using the STAMP, we could have just used the built-in serial port. That said, it was nice during testing to be able to run the DEBUG command in my PBASIC code, and see the output live on the console. If we used the built-in port, it would be more difficult to debug. Plus, the MAX232 kit is really slick, and relatively inexpensive.*

Yes, there are other subscribers in prison, but we can't give out the exact number.—Ed.

An Epistle for General Release

The New Bedford Monthly Meeting of Friends met this second day of January 2005 and resolved to declare our recognition of the good that free software is doing in the world and to thank those who have shared the fruit of their labor.

We single out this activity for the following reasons.

That our Meeting uses these products for administrative purposes and that we hope to soon use them to help others. This is our thank-you note.

That those who are doing this work might better realize their own Light. We see Godliness in their actions and by drawing their attention to that Godliness may we let them feel it more strongly.

That people generally may know of and use this software and save their resources for other needs.

The society at large, and especially those who regulate, legislate, or adjudicate, may note the public good done by such sharing of intellectual property. Society should look kindly on this sharing, a sharing which its laws seem ill suited to promote.

In using the words free software we mean software which is put in the public domain or is released with conditions that ensure that any interested person may have, use, improve, and redistribute the software.

We welcome your letters. Please submit "Letters to the Editor" to ljeditor@ssc.com or SSC/Editorial, PO Box 55549, Seattle, WA 98155-0549 USA.

EmperorLinux

...where Linux & laptops converge

The Meteor: 3lb Linux

- Sharp Actius MM20/MP30
- 10.4'' XGA Screen
- X@1024x768
- 1.6 GHz Transmeta Efficeon
- 20-40 GB Hard Drive
- 512-1024 MB RAM
- CD-RW/DVD (MP30)
- 802.11b/g Wireless
- Ethernet/USB2
- ACPI Hibernate
- 1'' Thin

The SilverComet: 4 lb Linux

- Sony VAIO S170
- 13.3" WXGA+ Screen
- X@1280x800
- 1.5-2.0 GHz Pentium-M
- 100 GB Hard Drive
- 256-1024 MB RAM
- CDRW/DVD or DVD-RW
- 802.11b/g Wireless
- 10/100/1000 Ethernet
- ACPI Hibernate
- USB/FireWire

The Toucan: 5 lb Linux

- IBM ThinkPad T Series
- 14.0" XGA+/15.0'' UXGA
- X@1400x1050/X@1600x1200
- ATI Radeon Graphics
- 1.6-2.1 GHz Pentium-M 7xx
- 40-80 GB Hard Drive
- 512-2048 MB RAM
- CDRW/DVD or DVD-RW
- 802.11b/g Wireless
- 10/100/1000 Mbps Ethernet
- APM Suspend/Hibernate

The Rhino: 7 lb Linux

- Dell Latitude D800
- 15.4'' WUXGA Screen
- X@1920x1200
- NVIDIA GeForce graphics
- 1.7-2.1 GHz Pentium-M 7xx
- 30-80 GB Hard Drive
- 256-2048 MB RAM
- CDRW/DVD or DVD+RW
- 802.11b/g Wireless
- 10/100/1000 Ethernet
- USB/FireWire

Since 1999, EmperorLinux has provided pre-installed Linux laptop solutions to universities, corporations, and individual Linux enthusiasts. We specialize in the installation and configuration of the Linux operating system on a wide range of the finest laptop and notebook computers made by IBM, Dell, Sharp, and Sony. We offer a range of the latest Linux distributions, as well as Windows dual boot options. We customize each Linux distribution to the particular machine it will run upon and provide support for: ethernet, modem, wireless, PCMCIA, USB, FireWire, X-server, CD/DVD/CDRW, sound, power management, and more. All our systems come with one year of Linux technical support by both phone and e-mail, and full manufacturers' warranties apply. Visit www.EmperorLinux.com or call 1-888-651-6686 for details.

Custom Configurations

Linux Pre-Installed

Technical Support

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

YOUR HIGH PERFORMANCE COMPUTING SOLUTION HAS ARRIVED.

VXRACK™ with the Intel® Xeon™ processor helps you simplify computing operations, accelerate performance and accomplish more in less time.

1 Choose one of the 3 convenient rack sizes

VXR-96
Rack accomodating up to
96 VXBlades/**192 Processors**
36TB of aggregated **Storage**
1.15TB of Global Memory
Power Distribution Included
Patented Architecture
Advanced Cooling System
Integrated InfiniBand Cable Mgnt.

$ 1,750.00*

VXR-128
Rack accomodating up to
128 VXBlades/**256 Processors**
48TB of aggregated **Storage**
1.5TB of Global Memory
Power Distribution Included
Patented Architecture
Advanced Cooling System
Integrated InfiniBand Cable Mgnt.

$ 2,190.00*

VXR-72
Rack accomodating up to
72 VXBlades/**144 Processors**
27TB of aggregated **Storage**
864GB of Global Memory
Power Distribution Included
Patented Architecture
Advanced Cooling System
Integrated InfiniBand Cable Mgnt.

$ 1,590.00*

CIARA TECHNOLOGIES

For more Information call
or visit us at

VXB-7221B
Intel SE7221B Motherboard
800MHz Front Side Bus
Intel® Pentium® 4 3.2GHz
1GB DDR2 400 Memory
Single 40GB 7200RPM ATA Drive
One **PCI/Express Slot Available**
Dual 10/100/1000 Intel Lan Port
350W Power Supply

$ 985.00

VXB-7501W
Intel SE7501W Motherboard
533MHz Front Side Bus
2 x Intel® Xeon™ 3.06GHz
2GB DDR 333 ECC Reg.Mem
Single 40GB 7200RPM ATA Drive
One **PCI/X Slot Available**
Dual 10/100/1000 Intel Lan Port
350W Power Supply

$ 2,355.00

VXB-7520J
Intel SE7520J Motherboard
800MHz Front Side Bus
2 x Intel® EM64T Xeon™ 3.2GHz
2GB DDR2 400 ECC Reg.Mem
Single 40GB 7200RPM ATA Drive
One **PCI/Express Slot Available**
Dual 10/100/1000 Intel Lan Port
500W Power Supply

$ 2,950.00

2 Choose one or more type of VXBlade

3 Add, Mutiply,That's it. Easy as 1, 2, 3...

For example you choose the following: One VXR-96 with 48 Dual Intel® EM64T Xeon™ and 40 Single Intel® Pentium®4. You take 1 (VXR-96) + 48 (VXB-7520J) + 40 (VXB-7221B)...That's it

VXRACK

THE FUTURE OF CLUSTER TECHNOLOGY

CIARA TECHNOLOGIES...A GLOBAL SOLUTION PROVIDER.

Ciara Technologies is a world-class computer systems manufacturer. Ciara designs, develops, manufactures, markets, services, and supports a variety of computer systems including graphic workstations, rackmount and tower servers, networked storage and the newly acclaimed VXRACK™ Cluster Technology. The company's state of the art supercomputer cluster is based on the Intel IA32 and IA64 architectures and utilizes Linux operating systems. We are proud to be recognized by Intel as an "Intel Premier Provider". Choosing Ciara is choosing a single point of contact for all your IT requirements. All our products are built under the ISO 9001 standards and regulations. The growth of Ciara enabled the company to move its 300+ employees,in February 2003, to an ultra-modern plant of 576,000 sqft.. Ciara now has the capability of producing more than 500,000 units per year.

866-7VX-RACK (866-789-7225)
WWW.VXRACK.COM

intel inside XEON

On the WEB

Being a publication ourselves and a part of a company that has published books, magazines and Web sites, we're always interested in what is happening in the world of publishing. Based on the following Web articles, available on the *Linux Journal* Web site, one of the newest trends is moving publications toward an open-source model:

» **This issue's EOF discusses how the open-source paradigm is being applied to scientific publishing to encourage equal access to information published in scientific journals. Author Christopher Frenz explains how this growing movement led to the National Institutes of Health (NIH) asking that published research results funded by NIH be released without cost after a period of six months. In his Web follow-up article, "Voice Your Opinion to the NIH" (www.linuxjournal.com/article/8061), Frenz outlines how you can let the NIH know your thoughts about open access for science.**

» **As Clay Dowling explains, however, open-source publishing models are not being investigated and used only for science. "Publishing Open-Source Documents with Open-Source Tools" (www.linuxjournal.com/article/8062) describes the entirely open-source process and tools used to produce *The Shadow of Yesterday*, written by Clinton Nixon of Anvilwerks (www.anvilworks.com). Dowling and Nixon also discuss "the practical business impact of publishing an open-source document with open-source tools".**

diff -u

What's New in Kernel Development

Linus Torvalds and **Andrew Morton** are still trying to find the best way to continue developing Linux. With the death of the idea of a stable/unstable series, there is still a push toward stability for each actual point release, such as 2.6.9 and 2.6.10. However, many users are reluctant to test the 2.6 kernels because of the tremendous amount of development going into them. Linus, Andrew and others have been giving thought to how to attract more testers to the now unpredictable official tree. One idea has been to bring back the stable/unstable concept for alternating versions. So 2.6.11 would be a stabilization kernel, with only bug-fixes for a couple of months, while 2.6.12 would be a new-feature kernel for a couple months, and so on. Another possibility would be to add a fourth number to the version, with numbers like 2.6.11.2 and 2.6.11.3, and these releases would be used for bug-fixes, while more development takes place on 2.6.12. So far nothing is certain, and Linus and Andrew are still trying to figure out the impact of abandoning the original stable/unstable development system. Stay tuned.

An interesting copyright question arose when **Adrian Bunk** noticed that **ReiserFS** files included a notice implicitly transferring copyright of all additions to **Hans Reiser**. The authors of the code explicitly could retain copyright by including text with their contributions, but Adrian felt there was something fishy about it. Linus Torvalds has given his support to Hans' copyright handling, and Hans himself also makes a point of asking all contributors directly, for the copyright assignment. According to Hans, the text is only in the source files in order to cover his backside from the likes of **The SCO Group**. And as **Christoph Hellwig** has pointed out, **SGI** makes the same request for copyright assignment from anyone contributing to the **XFS** filesystem. With precedent, politeness and an affirmation from the top Linux dog, it's possible this practice may spread to other areas of the kernel as well.

Marcus Metzler noticed that **iRiver** had released a binary-only product based on Linux and had refused to release any source code along with it. They certainly have made no secret of the fact that their multimedia player is Linux-based in their publicity and manuals, but no copy of the GPL, nor any offer to provide sources, have been found on their site or in their product.

The **SquashFS** compressed filesystem hovers on the brink of acceptance into the official kernel tree. **Phillip Lougher**'s code is self-contained, functional and clean. Folks like **Greg Kroah-Hartman** have been urging him to submit the code, but Phillip is reluctant. He has many new features to add, and whether it would be best to implement these before or after acceptance into the official kernel is not clear to him. I think it is a safe bet that SquashFS will have no trouble getting into 2.6, whenever Phillip decides the time is right. The kernel dudes eagerly await his submission.

FUSE, on the other hand, a user-space filesystem actively trying to be accepted into the main kernel tree, is running into serious problems. Linus Torvalds, in particular, believes that filesystems simply are not supposed to be user-space creatures. Divorcing a filesystem from the kernel, he says, is the same as microkernels' attempt to split the guts of a system into discrete pieces. For the same reason that Linus believes in a monolithic kernel structure, he believes that a user-space filesystem is a bad idea. On the other hand, Linus has said he'd be willing to accept FUSE, with a restricted feature set, if it avoided certain ugly behaviors that he feels should not be the province of a user-space filesystem anyway. He had a similar set of restrictions with **DevFS** long ago. The DevFS situation turned into a mess, partly because the /dev directory is so central to Linux. A single filesystem probably will be nowhere near as controversial.

—ZACK BROWN

Your Network.

Your Network...

...on TOE

Enabling networks to cover more ground faster.

Plug SBE's TCP/IP Offload Engine (TOE) into your system to see the performance boost for yourself...

All TOEs should process TCP/IP at network speeds, provide full segmentation and assembly, terminate multiple simultaneous sessions, and minimize transaction latency, without host intervention. However, not all TOEs are alike...depending on the individual manufacturer's target market, the devices vary in their ability to fully handle these essential criteria.

Adding a TOE to your existing system is only a cost-effective option if it can truly heighten your network performance at a fraction of the cost of purchasing an additional server. Only one TOE board has proven to provide peak performance across all four metrics of TOE effectiveness. While other TOE vendors offer solutions capable of satisfying one, two or maybe three of the critical TOE performance metrics,

SBE is today's only source to deliver Gigabit Ethernet throughout at line rate, over 70% reduction in CPU utilization, 32 microsecond transaction latency, and support for high session count applications, all on one PCI or PMC-based board.

Skeptical? Burned by inferior TOE solutions in the past? Well, now you can take advantage of our risk-free offer to see the results of SBE's TOE for yourself...

Contact us to qualify for a free test drive of the SBE TOE board in your application. Plus, ask us about how to win a free iPOD® mini.*

Phone: 800-214-4723
Email: info@sbei.com
Web: www.sbei.net/tryTOE.htm

SBE®

Linux On Demand

flexibility on demand | 925-355-2000 | info@sbei.com | www.sbei.com

* Restrictions apply. While supplies last. iPOD mini is a registered trademark of Apple Computer, Inc.

A Hot New Linux PXA

The coolest Linux product I saw at CES 2005 (the giant Consumer Electronics Show in Las Vegas—for more, see Linux for Suits on page 46) was the new Archos PMA430 Pocket Media Assistant. Because you can stick just about any noun you want between "Personal" and "Assistant", let's call it a PXA. It's less than an inch thick, 3.1" wide, 4.9" long and just under 10 ounces.

Because it's Linux, open source and a member of nobody's media management silo, it's free to do all kinds of stuff that Apple, Sony and other handheld makers with lock-in agendas will never support on their own devices. For example, it will *record* digital audio as well as play it back, which it can do in Ogg Vorbis as well as MP3 and other formats. It will record and play back digital video (MPEG-4 SP on a 3.5" 320×240 screen). It's a full-featured PDA, using Qtopia software, and a photo viewer with a 30GB hard drive that also serves as a peripheral storage volume through USB 2 or USB 1. It has ten hours of battery life playing audio and about half that playing video. It runs games. It has built-in Wi-Fi and an Opera browser. Best of all, it's open to anything written for its Linux OS. To that end, the company plans to have a software development kit released by the time you read this.

—DOC SEARLS

Ten Years Ago in *Linux Journal*: April 1995

Kurt Reisler wrote that the Digital user group DECUS was planning a half-day seminar led by Linus Torvalds at its May 1995 conference, plus a full day of other Linux activities. Looking forward to the Digital Alpha port of Linux, he wrote, "Imagine your Linux system running at 300+ MIPS."

The transition from a.out to ELF shared libraries was in progress, and the issue covered both. Eric Kasten wrote a shared library tutorial, including how to create the then-current a.out format. "The current a.out shared libraries will probably need to be supported for some time", he wrote. Meanwhile Eric Youngdale contributed an introduction to ELF, including the reasons we were all switching to ELF.

Joesph Brothers wrote a tour of hardware architectures with Linux ports. At the time, only x86, Motorola 68k and Alpha would run a shell. Others in progress were MIPS, SPARC and PowerPC. Alpha was the BogoMips champion at 149.49. The "bogo-fastest" x86 listed was a 486DX4/100 at 50.08.

Pacific Hi-Tech advertised the "Linux Run-Time System 1.0", a live CD distribution that booted and ran without installing to the hard drive, for $29.95 US.

—DON MARTI

They Said It

Only understanding for our neighbors, justice in our dealings, and willingness to help our fellow men can give human society permanence and assure security for the individual.

—ALBERT EINSTEIN
www.empyrean.ca/words/quotes/einstein.html

Powerful, reliable software and improved technology are useful byproducts of freedom, but the freedom to have a community is important in its own right.

—RICHARD STALLMAN
gnu.planetmirror.com/philosophy/gpl-american-way.html

The GPL is the most popular license for free software....As of April 2004 the GPL accounted for 74.6% of the 23,479 projects with an OSI-approved open-source license listed on Freshmeat. The GPL also accounted for 68.5% of the 52,182 free or open source software projects listed on SourceForge.

—WIKIPEDIA
en.wikipedia.org/wiki/GNU/GPL

Column 100

We've gone from rolling our own CGI scripts for everything to a profusion of Web tools and frameworks. What's next for Web development?

BY REUVEN M. LERNER

Welcome to the 100th installment of At the Forge! Yes, that's right, this is the 100th column that I have written for *Linux Journal* and before it, SSC's *Websmith*, starting in the spring of 1996. For many years now, I have enjoyed having the monthly opportunity to explore Web- and server-side technologies.

This month, I want to look back at some of the history of server-side and Web/database programming, so we can gain some appreciation for where things currently stand. We then explore the Web as it stands today and consider where things will go in the coming years.

Looking Back

Today it's easy to take the Web and Internet for granted. I keep track of my bank accounts on the Web; I buy books from on-line bookstores; I read Weblogs using a Web-based RSS reader; I access newspapers more current than their printed counterparts; I chat with friends and relatives by using instant messenger programs, and I even receive payments by way of PayPal. It often has been said that residents of Manhattan never need to leave their homes, because everything can be delivered. For better or worse, the Internet is making that a reality for a growing number of people all over the world.

The Internet's maturation for business and pleasure has been a result of a dramatic transformation. Originally, Web servers were mechanisms for sharing stored plain-text and HTML-formatted text documents. But soon after it became popular to explore the relatively limited number of documents on the Web, someone realized that HTTP's inherent client-server nature made it possible to create documents dynamically in response to a request. An HTTP client requesting a document from a server had no way of knowing if the document had been sitting on the server's filesystem for several months or if it was created on the spot in response to this request. This insight transformed the Web forever, turning it into a platform for real-time document generation and application development, rather than a simple, shared repository for static documents.

The beginnings of this dynamic revolution were fairly primitive. The first dynamically generated content was little more than a wrapper around traditional UNIX command-line programs such as mail and finger. One of the first programs that my friends and I wrote, for example, was a simple program that made it possible to search through the content of our newspaper's on-line archives. Of course, my friends and I could have created specialized HTTP servers with this functionality. Luckily for us and for all Web developers, the designers of NCSA httpd, the forerunner of Apache, made it possible for any program on the server to communicate by using HTTP through its common gateway interface, otherwise known as CGI. CGI meant that any program on our server could be accessible on the Web, merely by wrapping it inside of a CGI program.

Things still were rough in those early years. We all assumed that the Web was inherently stateless and were pleasantly surprised when Netscape announced the creation of cookies, making it possible for servers to keep track of user-specific information. No programs yet existed to report on Web traffic, let alone libraries that took care of the low-level details associated with Web programming. Debugging consisted of watching the Web server's error log. And using anything more complicated than a simple text file was considered a sophisticated data-storage technique.

Here and Now

Today, of course, Web development is a far cry from what it was back then. Downloading and installing the latest version of Apache is a trivial act; within several minutes of visiting www.apache.org, you can have a state-of-the-art Web server running on your favorite computer. Relational databases are an unstated requirement for nearly any sophisticated Web application that you might want to create. But much of the time, you don't even have to create your own programs—the number of libraries, applications and frameworks now available for creating Web/database applications has become overwhelming. It used to be that you needed to search high and low for an open-source application that would suit your needs. Nowadays, it still takes time to find the right application, but that's because you need to sort through so many bad or inappropriate ones before finding the one that is right for you.

Moreover, the community of developers has matured tremendously over the past few years. There never was a lack of goodwill or help for newcomers to the server-side programming world, but there often was a lack of experience, because so little had been tried. In some ways, the early days of Web programming resembled a network of research labs, each of which would share its experiences with the rest of the community. Today, there is a great deal of experience, both in the Open Source community and behind corporate doors. A young programmer interested in creating new applications has an almost endless supply of books, magazines, Web sites and source code to look and learn from.

It's also true that the most popular programming languages used to create Web/database applications—Perl, Python, PHP and Java—have matured significantly over the past few years. But improvements to these languages and their libraries have impressed me less than the trend toward high-level languages in the computer industry.

Back when the Web was coming into its own, most people developed software in C and C++. People who programmed in high-level languages, such as Perl and Python, were seen as glorified tinkerers or people who were somehow less serious than their compiled-language counterparts. The Web has changed all of this; it now is possible to be seen as a serious application developer even if you're only working in PHP. Of course, compiled C code still executes faster than the equivalent high-level code. But, the corresponding difference in

development and debugging time generally are so great that almost no one writes Web applications in C.

Increasingly, we see that mainstream companies are moving toward high-level languages in general and toward many open-source programs in particular. Many companies, from Amazon to eBay, have discovered that their programmers are more productive when using high-level languages. The fact that Java and C# are the lowest-level Web development languages in mainstream use says a lot about where the industry is going. Languages that make it possible for programmers to concentrate on high-level ideas rather than get their hands dirty with individual bits and bytes have become mainstream. Java largely has failed as a desktop application language, but C# seems to be gaining some speed as a result of Microsoft's .NET initiative—which means that within the next few years, most desktop applications might be running in languages that lack pointers and include garbage collection.

Obviously, there are many reasons, both technical and financial, why programmers are moving toward such languages. I have no doubt, though, that the Web has helped to push this issue to the forefront. High-level languages such as Perl are suited perfectly to the Web, with its ambiguous data types, its need for database connectivity and the need for easy-to-use, powerful text strings and string-manipulation libraries. The Web is nothing more than a bunch of text strings being hurled over the network, and no one can hurl text faster or farther than a high-level open-source language.

Dramatic growth also has occurred in the number of frameworks available for the creation of server-side applications. Even if you have an easy-to-use programming language, you still need to implement your own systems for managing users, groups, permissions, content and messages. By using an existing framework, you can avoid that work and take advantage of someone else's experience. Frameworks have moved in two different general directions—content management systems, which perform just-in-time assembly of newspapers and magazines, and application servers, which provide developers with a toolkit for the creation of applications.

On the surface, you might think that application frameworks such as HTML::Mason, Zope, OpenACS and Java servlets/JSPs have little in common. But anyone who works with more than one of these systems quickly discovers that although each framework has its own approach, they share many commonalities. Moving from one framework to another still can be difficult, but once you have enough experience with several application frameworks, trying others becomes relatively easy.

Yes, being a Web developer is 2005 is quite pleasant compared with what we had to endure ten years ago. The software is increasingly mature, the community is large and helpful, we are no longer re-inventing the wheel every other week and the number of organizations moving sites to the Web means that there is some demand for our work in the marketplace.

The Future

Given such a rosy description of the present day, where are we going in the future? What trends will pick up speed

as we pass through 2005? To begin with, it is clear that the Web, by which I generally have meant the combination of HTTP, HTML and URLs, is slowly breaking apart into separate constituent parts. I always thought that the Web was unusually powerful because it combined three simple, powerful technologies—HTTP, HTML and URLs—that worked well together. But I now see that each is useful in its own right and is branching out into other uses.

Particularly interesting are Web services, which represent a new, rich and open communications protocol for programs other than Web browsers. When they were first revealed, I though that Web services were some simple ideas piggybacking on the Web's success and name recognition. Although this might be true regarding the poor name choice and although they might be simple in theory, Web services are quite powerful indeed. The idea that one application can connect to another without regard for operating system or programming language is nothing short of amazing. And although truly good uses for Web services remain relatively rare, Amazon, Google and Bloglines are demonstrating that it is possible to expose your internal API to customers and other outsiders without giving up the store.

A similar trend is the use of the Web browser as an integral component in desktop application development. Help systems now are built with HTML and miniature Web browsers, and there are some full-fledged applications, such as ActiveState's Komodo, that are based on the underlying Mozilla engine. I often have said that Mozilla is the new Emacs. Although Mozilla development significantly is harder than Emacs customization ever was, the fact that Mozilla provides a cross-platform, programmable environment for rich desktop applications is impressive and is likely to improve further.

One promising application is Sunbird, the Mozilla calendar program, which I have been using for several months on my own desktop. Sunbird still has a number of problems and bugs, but one of my favorite features is its use of the iCalendar standard to retrieve various calendars from the Internet using HTTP. Yes, that's right—I'm running a desktop application based on Mozilla that retrieves URLs by way of HTTP, but it's not a Web browser!

On the server side, collaboration is an increasingly important watchword. Although it might not meet the rigorous standards of a commercial encyclopedia, Wikipedia is where I first turn when I'm curious about a topic. And thanks to thousands of contributors, it is more than good enough for my day-to-day use. Managing that sort of collaboration is no mean feat, and the WikiMedia Foundation's MediaWiki software, based on PHP and MySQL, quietly is turning into a top-notch package for collective writing and editing.

Finally, there always is a need for better debugging and testing frameworks. The growing trend on this front is more testing and even test-based programming. Unit tests are never going to provide a complete measure of whether software works correctly—but wouldn't you rather know that all of your procedures are working correctly before you start trying to integrate them? Test-driven development has been identified as one of the key methodological changes of the last few years, and I believe that it will continue to grow in popularity as software becomes increasingly complex.

Conclusion

It has been my pleasure to write 100 installments of At the Forge so far. But as you can tell from my above enthusiasm, many new challenges await Web/database developers, which means it'll take at least 100 more columns to cover them all. Over the coming months, we are going to look at a number of the ideas mentioned in this column, including iCalendar, Wiki software, Web services and test-driven development.

It might be more than ten years old, but the Web continues to be a fun, exciting and intriguing medium in which to work. Drop me a line at reuven@lerner.co.il telling me where you think the Web is headed—and what projects, technologies and trends you would like to see me cover in the coming months and years.

Reuven M. Lerner, a long-time Web/database consultant and developer, now is a graduate student in the Learning Sciences program at Northwestern University. His Weblog is at **altneuland.lerner.co.il**, and you can reach him at reuven@lerner.co.il.

Why buy more Servers when all you need is more Disks?

Linux + Disks + Ethernet = EtherDrive®

Disks go inside servers... right? If you run out of disk space you get another server... right? Well, that used to be the case, but not any more. Now you can expand the disk space on any server with EtherDrive Storage Blades.

EtherDrive Storage Blades are simple and easy to use. And the best part is, you already know how. An EtherDrive Storage Blade is a disk drive mounted on a very small server attached directly to your network. Each blade is a nanoserver, with firmware that puts the disk's storage right on your network. No IP addresses. Just disks on the network accessible by your servers.

Just Disk Drives on Ethernet

The open protocol, ATA-over-Ethernet, allows the most in flexibility and operation simplicity. Since EtherDrive blades just look like local disks, you already know how to use them. Use any file system software. Use any RAID software, or Coraid's open source RAIDBlade appliance. Use any volume management software. It's all up to you to decide how to organize your disks. And, since the protocol is open, you know everything about how it works. The protocol is simple, only 8 pages. The open source device driver means you never have to look at the protocol. But isn't it good to know you can?

Complete Control

You have complete control over the contents of the disk. EtherDrive doesn't store anything on your disks that you don't want. You can take a disk from a running system, install it on an EtherDrive Storage Blade, and mount it. That means you are always in control. No data reformatting. No captive data. Just disk drives on the network. You never have to worry about getting your data off of an EtherDrive Blade if it fails. Just mount the disk on a system or another Blade and you're back in business.

EtherDrive Storage Blades insert into a shelf of 10 slots. Using 400GB ATA disks you can have 4TB in one 3U rack space. You can add up to 4,095 shelves on a single network. That means you can have servers sharing more than 16 Petabytes.....Imagine that.

40,000 Disks on Your Servers

A system that can go from a couple of disks, all the way to 40,000 disks, in whatever increment you want. That's probably more than you'll ever need, but isn't that the idea of scalability? Since our shelves mount in simple relay racks, just like your switches, you never run out of room. Never have your data captive inside one server's chassis. Never have to fork lift obsolete systems. Never have to buy more servers when all you need is more disks.

Processing Power with Each Blade

EtherDrive Storage Blades can go fast, too. Since each blade has its own cpu, memory and Ethernet interface, they can all work independently or in unison. Striping software can read/write blades in parallel. The wider the stripe, the faster the I/O.

Each Blade isn't limited to a single server, either. Many servers can access the same group of EtherDrive Storage Blades. They can share read-only file systems or use available software like Red Hat's GFS to share read/write file systems.

Using EtherDrive Storage Blades you only add pennies to the cost of the raw storage. **Less than $0.65 per Gigabyte.**

CORAID

www.coraid.com
info@coraid.com
1-877-548-7200

Dynamic Interrupt Request Allocation for Device Drivers

Interrupts are how hardware gets software's attention. Here's how they work. BY DR B. THANGARAJU

A computer cannot meet its requirements unless it communicates with its external devices. An interrupt is a communication gateway between the device and a processor. The allocation of an interrupt request line for a device and how the interrupt is handled play vital roles in device driver development. As the number of interrupt request lines in a system is limited, sharing an interrupt between devices is a must to access more devices. Any attempt to allocate an interrupt already in use, however, eventually crashes the system. This article explains the basics of the interrupt and the fundamentals of interrupt handling and includes an implementation of an interrupt request (IRQ) allocation for a character device.

The purpose of any device is to do some useful job, and to do so it should communicate with the microprocessor. When a processor wants to communicate with a device, it sends instructions to the device controller. A device controller controls the operation of a device. Similarly, if a device wants to reply to a processor that says new data is ready to be retrieved, the devices generate an interrupt to capture the processor's attention. An interrupt is a hardware mechanism that enables a device to communicate with a processor.

Until version 2.6, Linux had been non-preemptive, meaning that when a process is running in kernel mode, if any higher-priority process arrives in the ready-to-run queue, the lower-priority process cannot be preempted until it returns to user mode. But, an interrupt is allowed to divert CPU attention even though it is executing a process in kernel mode. This helps to improve the throughput of a system. When an interrupt occurs, the CPU suspends the current task and executes some other code, which responds to whatever event caused the interrupt.

Each device in a computer has a device controller, and it has a hardware pin that is used to assert when the device requires CPU service. This pin is attached to the corresponding interrupt pin in the CPU, which facilitates communication. The pin in the processor connected to the controller is called the interrupt request line. A CPU has several such pins so that many devices can be serviced by the processor. In a modern operating system, a programmable interrupt controller (PIC) is used to manage the IRQ lines between the processor and the various device controllers. The number of free IRQs in a system is restricted, but Linux has a mechanism to allow many pieces of hardware to share the same interrupts.

Interrupt servicing can be compared to a programmer's job. The programmer opens a mailbox and does his routine programming work. When new mail arrives, he is interrupted by a beep or by some other notification at the corner of the screen. Immediately, he saves the program and switches over to the mailbox. He then reads the mail, sends an acknowledgement and resumes his earlier work. A detailed reply listing the steps he has taken is sent later.

Similarly, when a CPU executes a process, a device can send an interrupt to the CPU regarding some task, for example, data is ready for transfer. When an interrupt comes, the CPU instantly saves the current value of the program counter in the kernel mode stack and executes the corresponding interrupt service routine (ISR). An ISR is a function situated in the kernel that determines the nature of the interrupt and performs whatever actions are needed, such as moving a block of data from hard disk to main memory. After executing the ISR, the CPU resumes the earlier process and executes.

A device driver is a software module in the kernel that waits for requests from the application program. Whenever an application wants to read data from a device, the corresponding device driver is invoked immediately, and the respective device is open for reading. If the system is waiting for slow hardware, it cannot do any useful job. One of the prime aims of kernel developers is to utilize system resources effectively. To avoid waiting for data from the hardware, the kernel gives this job to the device controller and resumes the stopped process. When reading completes, the device notifies the CPU through an interrupt. The processor then executes the corresponding ISR.

Interrupt Classification

Interrupts are divided into two broad categories, synchronous and asynchronous. Synchronous interrupts are generated by the CPU control unit when it is executing an instruction. The control unit issues an interrupt after terminating the instructions, hence the name synchronous interrupt. Asynchronous interrupts are created by hardware devices at random times with respect to the CPU clock. In the Intel context, the first one is called exceptions and the second is interrupts. Interrupt is identified by an unsigned one-byte integer called a vector. The vector ranges between 0 to 255. The first 32 (0–31) vectors are exceptions and non-maskable interrupts, which was explained in my article "Linux Signals for the Application Programmer", *LJ*, March 2003. The range from 32–47 is assigned to maskable interrupts and is generated by IRQs (0–15 IRQ line num-

bers). The last range, from 48–255, is used to identify software interrupts; an example of this is interrupt 128 (int 0X80 assembly instructions), which is used to implement system calls.

IRQ Allocation

A snapshot of interrupts already in use on the system is stored in the /proc directory. The `$cat /proc/interrupt` command displays the data related to the interrupts. The following output was displayed on my machine:

```
CPU0
  0:    82821789          XT-PIC  timer
  1:         122          XT-PIC  i8042
  2:           0          XT-PIC  cascade
  8:           1          XT-PIC  rtc
 10:      154190          XT-PIC  eth0
 12:         100          XT-PIC  i8042
 14:       21578          XT-PIC  ide0
 15:          18          XT-PIC  ide1
NMI:           0
ERR:           0
```

The first column is the IRQ line (vector ranges from 32–47), and the next column is the number of times the interrupts are delivered in the CPU after booting the system. The third column is related to the PIC, and the last column is the list of the device names that have registered handlers for the corresponding interrupt.

The simplest way to load a device driver dynamically is first to find the unused IRQ line in the system. A request_irq function is used to allocate a specified IRQ line number for a device. The syntax for the request_irq follows and is declared in linux/sched.h:

```
int
request_irq (unsigned int irq,
             void (*handler) (int, void *,
                              struct pt_regs *),
             unsigned long flags,
             const char *device, void *dev_id);
```

The details of the arguments in this function are:

- unsigned int irq: interrupt number, which we want to request from the system.
- void (*handler) (int, void *, struct pt_regs *): whenever an interrupt is generated, we have to write ISRs to handle the interrupt; otherwise, the processor simply acknowledges it and does nothing else for that interrupt. This argument is the pointer to the handler function. The syntax for the handler function is:

  ```
  void
  handler (int irq, void *dev_id,
           struct pt_regs *regs);
  ```

  The first argument is the IRQ number, which we already have mentioned in the request_irq function. The second argument is a device identifier, using major and minor numbers to identify which device is in charge of the current interrupt event. The third argument is used to save the process' context in the kernel stack before the processor starts executing the interrupt handler function. This structure is used when the system resumes the execution of the earlier process. Normally, device driver writers need not worry about this argument.
- unsigned long flags: the flags variable is used for interrupt management. The SA_INTERRUPT flag is set for fast interrupt handler, and it disables all the maskable interrupt. SA_SHIRQ is set when we want to share the irq with more than one device, SA_PROBE is set if we are interested in probing a hardware device using the IRQ line, and SA_RANDOM is used to seed the kernel random number generator. For more details of this flag, see /usr/src/linux/drivers/char/random.c.
- constant char *device: a device name that holds the IRQ.
- void *dev_id: the device identifier—it's a pointer to the device structure. When the interrupt is shared, this field points to the particular device.

The request_irq function returns 0 on success and -EBUSY when the allocation has failed. EBUSY is the error number of 16, which is described in the /usr/src/linux/include/asm/errno.h file. The free_irq function releases the IRQ number from the device. The syntax for this function is:

```
free_irq (unsigned int irq, void *dev_id);
```

The explanation for the arguments is the same as above.

An ISR is invoked whenever an interrupt occurs. The operations to be performed on the cause of the interrupt are described in the ISR. The kernel maintains a table in memory, which contains the addresses of the interrupt routines (interrupt vectors). When an interrupt occurs, the processor checks the address of the ISR in the interrupt vector table and then executes. The task of the ISR is to react to the device according to the nature of the interrupt, such as read or write data. Typically, the ISR wakes up sleeping processes on the device if the interrupt signals the event for which they are waiting.

The amount of time the processor takes to respond to an interrupt is called interrupt latency. Interrupt latency is composed of hardware propagation time, register saving time and software propagation time. Interrupt latency should be minimal to improve the system's performance; for this reason, the ISR should be short and disable interrupts only for a brief time. Other interrupts can occur while interrupts are disabled, but the processor does not allow them until interrupts are re-enabled. If more than one interrupt is blocked, the processor allows them in priority order when it is ready for interrupt service.

Device driver developers should disable interrupts in driver code only when necessary, because the system does not update the system timers, transfer network packets to and from buffers and so on during the interrupt disabling. Driver developers should write ISRs to release the processor for other tasks. In real-world scenarios, however, ISRs handle lengthy tasks. In such situations, the ISR can do only the time-critical communication with the hardware to disable the interrupt and use the

tasklet to perform most of the actual data transfer processing. The tasklet is the advanced feature in the latest Linux kernel that does certain operations related to the interrupt during safe times. The tasklet is the software interrupt, and it can be interrupted by other interrupts. The internals of the interrupts have been explained in detail by Bovet and Cesati (see the on-line Resources), and the implementation of the interrupts in device driver perspective is presented by Rubini and Corbet (see Resources).

Simple Implementation

Any kernel module includes a device driver that can be loaded with the existing kernel, even when the system is running. I explain the basic dynamic IRQ allocation procedure in a simple module shown in Listing 1. The following simple character device driver code describes the dynamic allocation of an IRQ line for a device named OurDevice. When you insert the module, the init_module function is executed. If it is allocated successfully, an unused major number and register for the given IRQ number for the device and the corresponding printk message then is printed. From here, we could check the IRQ allocation in the /proc directory. The given IRQ is released at the time the module is removed. The best place to register an IRQ number is an open entry point of a driver code, which subsequently frees the IRQ in a release function.

The my_module.c file is compiled with the 2.6.0-0.test2.1.29 kernel. The kernel-2.6.0-0.test2.1.30.i586.rpm was downloaded along with all the dependent RPMs and installed. The RPM was downloaded from people.redhat.com/arjanv/2.5/RPMS.kernel, and the device driver program was compiled as follows:

```
gcc -Wall -O3 -finline-functions \
-Wstrict-prototypes -falign-functions=4 \
-I/lib/modules/2.6.0-0.test2.1.29/build/include \
-I/lib/modules/2.6.0-0.test2.1.29/build/include/
➥asm/mach-default
-I./include -D__KERNEL__ -DMODULE -DEXPORT_SYMTAB \
-DKBUILD_MODNAME=my_module -c my_module.c -o \
my_module.o
```

After inserting my_module.o, if the major number and the IRQ allocation for the device are successful, the corresponding printk statement output can be seen. If the IRQ number already is in use by another device, the kernel unregisters the device and releases the major number. The `$cat /proc/interrupt` command displays the following output:

```
CPU0
  0:    82887219          XT-PIC  timer
  1:         122          XT-PIC  i8042
  2:           0          XT-PIC  cascade
  7:           0          XT-PIC  OurDevice
  8:           1          XT-PIC  rtc
 10:      154769          XT-PIC  eth0
 12:         100          XT-PIC  i8042
 14:       21636          XT-PIC  ide0
 15:          18          XT-PIC  ide1
NMI:           0
ERR:           0
```

Listing 1. my_module.c

```
#include <linux/init.h>
#include <linux/fs.h>
#include <linux/module.h>
#include <linux/sched.h>
#include <linux/interrupt.h>

static struct file_operations fops;
static int Major, irq = 7;

static void OurISR (int irq, void *device,
                    struct pt_regs *regs)
{
  /* important and immediate time critical tasks */
}

static int __init my_init_module(void)
{
    int status;
    Major = register_chrdev(0, "OurDevice", &fops);

    if (Major == -1) {
        printk (" Dynamic Major number "
                "allocation failed\n");
        return Major;
    }

    status = request_irq(irq,
                         (void *)OurISR,
                         SA_INTERRUPT,
                         "OurDevice", &fops);
    if (status == -EBUSY) {
        printk ("IRQ number allocation failed\n");
        unregister_chrdev(Major, "OurDevice");
        return status;
    }

    printk ("The module is successfully loaded\n");
    printk ("Major number for OurDevice:   %d\n",
            Major);
    printk ("IRQ number for OurDevice:     %d\n",
            irq);
    return 0;
}

static void __exit my_cleanup_module (void)
{
    printk("Major number %d  IRQ number %d "
           "are released\n", Major, irq);
    free_irq(irq, &fops);
    unregister_chrdev(Major, "OurDevice");
    printk("The Module is successfully unloaded\n");
}

module_init (my_init_module);
module_exit (my_cleanup_module);

MODULE_LICENSE("GPL");
```

The simplest way to load a device driver dynamically is first to find the unused IRQ line in the system.

An entry of OurDevice along with the IRQ line can be seen in the output. When we remove the module, the kernel frees the IRQ number, unregisters the device and releases the major number.

Conclusion

Hopefully, this article makes clear the fundamental concepts of interrupts and the interrupt handling routine. The discussion of the request_irq and free_irq function is useful when we use these concepts in device drivers. The dynamic IRQ allocation procedure has been explained with the simple character device driver code.

Acknowledgements

C. Surest is acknowledged for his help during the preparation of this article.

Resources for this article: www.linuxjournal.com/article/8064.

Dr B. Thangaraju received a PhD in Physics and worked as a research associate for five years at the Indian Institute of Science, India. He is presently working as a tech lead in the Embedded Systems Focus Group at Talent Transformation, Wipro Technologies, India. He works in the areas of Linux internals, the Linux kernel, Linux device drivers and embedded and real-time Linux. He can be reached at bt_raju@vsnl.net.

The Cook's Collection

Organize your books with an application that takes the ISBN and fills in the rest of the data for you, or catalog a collection of anything.

BY MARCEL GAGNÉ

So that is where my 2005 wine encyclopedia has gone! *Mon Dieu*, François, I've been looking for that everywhere. Wait a minute. That's my Parisienne cookbook, my Tuscan creations cookbook and my Provencal herbs reference. How many of my books do you have here? Non, *mon ami*, I am not suggesting anything other than I have been looking for these for some time now. Yes, you are right, at least they weren't lost. I think you had better prepare the tables, *mon ami*, our guests will be here any moment.

Too late, they already are here. Welcome, *mes amis* to *Chez Marcel*, where fine Linux fare is always on the menu, and the wine cellar is always among the greatest in the world. Please sit and make yourselves comfortable while François fetches the wine. Please, *mon ami*, head down to the north wing of the cellar and bring back the 2000 Bordeaux we were, ahem, subjecting to quality control earlier. It's next to the Margaux labeled "don't open until 2010". *Vite*, François. *Vite*!

While we wait for my faithful waiter to return with the wine, let me tell you about today's menu. As you know, *Chez Marcel* has served up a great number of recipes in the years we have been here. We've also served up a great deal of wine. Much as I would like to think that I can remember all of this information, the truth is somewhat more realistic. That's why there are shelves of books on Linux, cooking and wine in the kitchen, cellar and office. The problem becomes one of management, and that's why we need a database.

But what kind of database? How about something easy and extremely flexible. Meet Tellico. Robby Stephenson's Tellico is billed as a collection manager, but I like to think of it as a versatile personal library system. It's a great tool for keeping track of your many cookbooks as well as Linux books, science-fiction books, mysteries and so on (Figure 1). That in itself would make it an extremely useful tool for keeping track of what books various friends and family have borrowed. I don't know about you, *mes amis*, but I have lent out numerous books over the years that have never come back. The people who borrowed them forgot whom they borrowed books from, and I forgot whom I lent them to—with the exception of François. I keep a special list for him.

Tellico has templates to track other forms of collections as well, including videos, music, coins, stamps and more. There's even a template for your wine cellar. You also can create your own collections or modify existing forms. I show you more

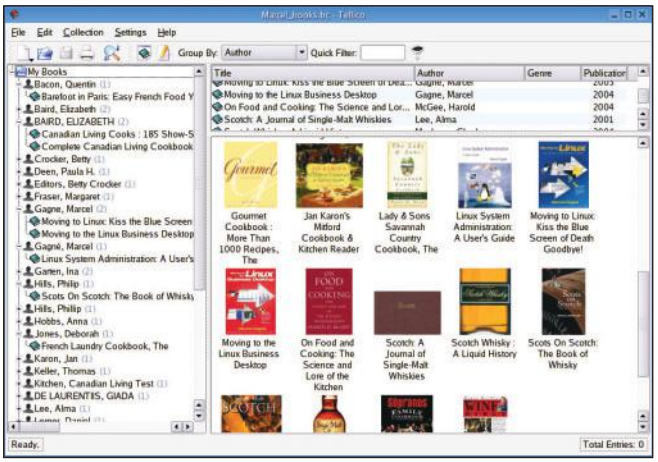


Figure 1. Tellico makes a great personal library system, and it looks good doing it.

and tell you how to work with it shortly. Prebuilt packages are available for a number of the major distributions, such as Fedora, SuSE, Mandrake, Slackware and others. You also can download the source (see the on-line Resources) and build it using our famous extract and build five-step:

```
tar -xzvf tellico-0.13.1.tar.gz
cd tellico-0.13.1
./configure --prefix=/usr
make
su -c "make install"
```

Tellico is a KDE 3.1 or greater package and requires the associated Qt and KDE development libraries. If you are working from source, you may want to consider building with a couple of additional but optional libraries. The taglib development libraries are the first option, which lets you read information from audio files—more on this shortly. Another optional library is yaz. Build Tellico with that and you have access to Z39.50 searches.

When you start Tellico—by running the command `tellico`—you start with the proverbial clean slate. Expand the program window to a comfortable size and start defining a collection. To create a book collection, click File on the Tellico

Figure 2. Entering a New Title into Your Book Collection

menu bar, then New and then select New Book Collection. I mentioned that Tellico is a great personal library system to let you record your books and keep track of when and where you got them, as well as who has borrowed them. Before we get to that stage, however, we need to enter the information from our collection (Figure 2).

Under the various tabs, you can enter the obvious title and author information as well as publisher, publishing date, edition, genre, series number, condition, whether the book is signed, whether it is currently loaned out and whether you have read it. Many more fields are available for you to explore yourself, but I must mention that you can enter a cover image too, as you saw in Figure 1.

If this seems like a lot of work and you don't feel like adding all this information yourself, there is another way. No, you don't have to hire anyone. All you need is a connection to the Internet, because Tellico offers the ultimate in convenience. Simply click Edit on the menu bar and select Internet Search.

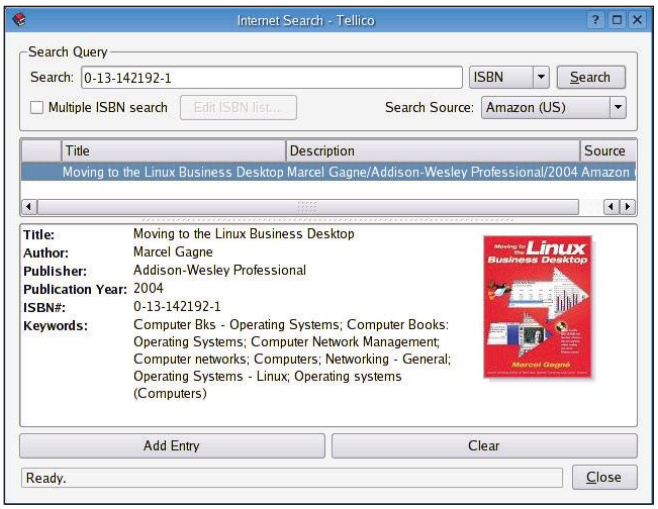


Figure 3. With an Internet connection, entering book information is a breeze.

When the Internet Search dialog appears (Figure 3), you can enter the book's title, author, International Standard Book Number (ISBN) or any keyword you wish. Searches are done on Amazon.com's database, although you can search on UK, Japan and Germany sites too. If you searched by ISBN, you likely will have only one entry returned, but other searches probably will return more than one title. Click to select the one you want, then click Add Entry. Your database automatically is updated along with a nice cover image.

Tellico provides an intelligent search dialog to find a particular title or range of titles. You also can access this information at a glance by adding or removing columns reflecting the various fields from the listings on the right-hand side. For instance, if you always want to know what is out, simply right-click on the fields bar and add Loaned. Titles with that field checked have a green check mark in that position.

Other options exist for bringing data into your Tellico collections besides the ones described here. Click on File and look under the Import submenu. There, you can find options to use data from simple CSV files, Alexandria, Bibtex and more.

The export function is even more interesting because this is where we enter into reporting. You can print an entry at any time,

but the export function is somewhat more powerful than this. For example, by selecting HTML export, you can generate an HTML page of all your books with whatever display fields your particular view uses. You then are asked for an HTML filename, whether you want to format all fields or selected entries only and so on. The result is a clean HTML-formatted page (Figure 4).

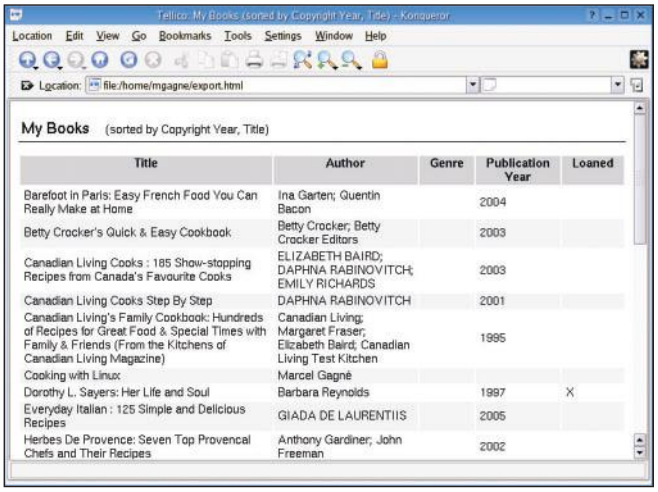


Figure 4. An HTML-Formatted Report

Before we move on, I want to tell you about one other export function of which I am particularly fond. Choose Export to PilotDB from the list, and you can generate a PDB format report readable by your favorite Palm document reader. Simply hotsync to install your document, and you have everything your need at your fingertips.

When I started telling you about Tellico, I mentioned that templates for other collections exist, including music collections. If you don't have a collection on the go already, click File and select New Music Collection. Entering a new CD title is a process similar to that of entering a book, except the fields are different. That said, adding your CD collection to your library is easy if you have the taglib extensions on your system. Simply enter a music CD into your CD-ROM or DVD drive, click File, select Import and choose Import Audio CD Data (Figure 5). The program reads the information from your CD and imports it to your collection.

Once your titles are entered, you can go back and fine-tune any information that might be missing. Of course, if your collection is on vinyl or tape, you have to enter everything manually. As with the book collection, you can enter that an album has been loaned to a friend. Knowing what books and music you have and where they are at the moment, you can sit back with a glass of wine and relax.

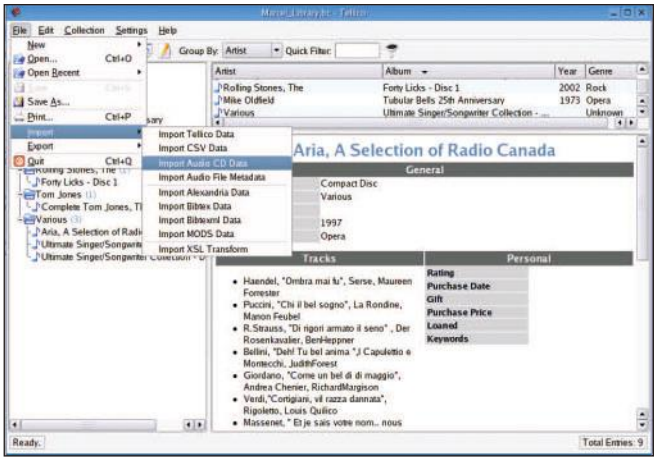


Figure 5. Tellico can read the title, artist and track information directly from your CDs.

And now, we find ourselves back at wine, which is not a bad place to be. What about your wine cellar? Incredibly, Tellico has something for the home wine cellar as well. In the same way that you created a book and music collection, you also can create a wine collection. Click File, then New and select New Wine Collection. Now, click Collection, then New Entry and start adding your wines, one by one (Figure 6).

Unfortunately, there is no magical entry system for building a database of your wine collection, no fanciful way to scan the labels and have all the information magically appear. Each bottle must be entered manually (Figure 5). Still, spending a little time in the wine cellar, studying and recording your collection should not be seen as chore but a labor of love.

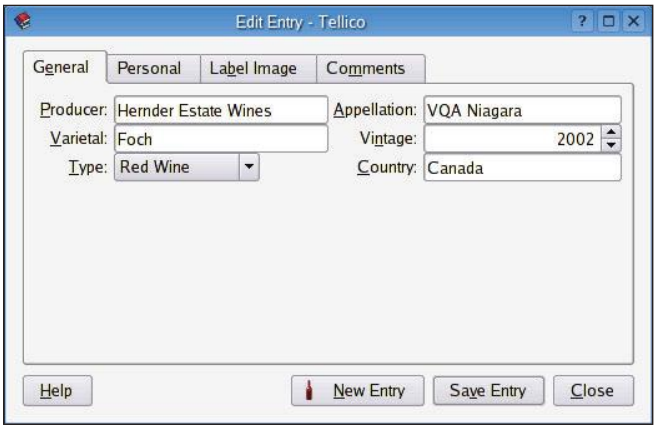


Figure 6. But of course, we can build a wine cellar database as well.

Finally, for those who have been asking themselves for a package that would allow them to create simple, custom databases, Tellico is also for you. Instead of using one of the predefined templates, choose to create a custom collection. The default collection fields are extremely simple here—title only—so you will want to modify it. After creating your custom collection, click Collection on the menu bar and select Collection fields. Here, you can define additional fields, whether text, numeric or whatever your needs might be.

On that note, *mes amis*, I see by the clock that closing time once again has arrived. Now that our wine cellar is entirely up to date, François can give you his complete attention and happily will refill your glasses. Until next time, *mes amis*, let us drink to one another's health. *A votre santé! Bon appétit!*

Resources for this article: www.linuxjournal.com/article/8063.

Marcel Gagné is an award-winning writer living in Mississauga, Ontario. He is the author of the all-new *Moving to the Linux Business Desktop* (ISBN 0-131-42192-1), his third book from Addison-Wesley. He also is a pilot, was a Top-40 disc jockey, writes science fiction and fantasy and folds a mean Origami T-Rex. He can be reached at mggagne@salmar.com. You can discover a lot of other things, including great WINE links from his Web site at **www.marcelgagne.com**.

Polywell First Class Linux Systems

The Most Reliable 64-bit Computing

2200A 1U 1TB Dual Opteron

- Dual AMD® Opteron™ 242 64-bit Processors
- 2GB 400MHz ECC DDR (Expandable to 16GB)
- 1TB (4x250G) SATA HD 7200RPM 8M
- 1 x PCI-X 133MHz Solt
- On-board Dual Gigabit Ethernet
- On-board ATI Graphics
- Slim CD-ROM Drive, Optional DVD-RW or CDRW
- Slim Floppy Drive
- 1U 24" Depth Rack Chassis with 460W P/S
- 4 x Swappable Drive Bays (SATA or SCSI)
- Supports Linux, FreeBSD or Windows
- Custom Configuration Available
- Please call for other Options

Order# 2200A242LX02A

$2,888

Linux Appliance PCs

- Custom Made Odd Size Chassis
- AC or DC Power Supply
- Low Power Voltage Processor
- or High Performance AMD® Opteron™ Processor
- Diskless or Flash OS Boot Drive
- Swapable Hard Drive, CD-ROM, FDD
- Integrated Graphics, Ethernet, USB
- Optional LCD LED Control Module
- IS2 Audio, MPEG2/4 Hardware Video
- Upto 4 Ethernet Ports or 4 Serial Ports
- We have over 18 years OEM Experience in Set top Box, Digital Media Player, POS Kiros, Thin Client, Networking Appliance,

OEM Appliance starts at

$299

8422B 2U 4-Way Opteron

- Quad AMD® Opteron™ 846 64-bit Processors
- 4GB 400MHz ECC DDR (Expandable to 32GB)
- 2 x 133, 2 x 66MHz PCI-X, 1x32bit PCI Slots
- Dual Gigabit Ethernet, U320 SCSI Controller
- Dual Hot-Swap 36GB SCSI Drive for Mirror
- Slim CD-ROM Drive, Optional DVD-RW or CDRW
- 2U 28" Depth Rack Chassis with 700W PFC P/S
- On-board ATI Graphics
- Supports Linux, FreeBSD or Windows
- Custom Configuration Available
- Please call for other Options

Order# 8422B846LX02B

$8,350

1U Power Saving ISP Server

- AMD® Sempron™ or Opteron™ Processor
- 512M Memory
- 80GB Hard Drive
- 10/100Mbit Ethernet
- We provide Drive Image Service
- 1U 14" Short Rack, allow 2 x 1U per Rack
- Low Power Usage to save your Data Center Cost
- Perfect Entry Level ISP Server or Appliance System
- IDE Flash Drive is available
- Supports Linux, FreeBSD or Windows
- Custom Configuration is Available
- Please call us to discuss your specification

Order# 6231U800ISP02E

starts at **$399**

8400B 4U 2TB 4Way Opteron

- Quad AMD® Opteron™ 846 64-bit Processors with Hyper Transport Technology
- 4GB 400MHz ECC DDR (Expandable to 32GB)
- 2 x 133, 2 x 66MHz PCI-X, 1x32bit PCI Slots
- Dual Gigabit Ethernet, U320 SCSI Controller
- On-board ATI Graphics
- 2TB 8 x 250GB SATA 7200RPM Hard Drive 8M
- CDRW-DVD Combo Drive, Floppy
- Optional Tape Back Up, Media Reader
- 4U 26" Depth Rack Chassis
- 950Watt 3+1 Redundant Hot Swap Power Supply
- Supports Linux, FreeBSD or Windows
- Custom Configuration Available
- Please call for other Options

Order# 8400B46LX02C

$9,999

SANAS 4U 9.6TB Storage

- SAN Ready NAS Storage Server with upto 24 x 400GB Hard Drive
- Dual Processor FreeBSD/Linux NAS Appliance
- upto 4 x 2G Fibre Channel Ports for SAN
- upto 8 Gigabit Ethernet Ports for NAS
- Supports UNIX, Linux, FreeBSD, Windows
- 24 x Hot Swap Drive Bays for SATA or SCSI Drives
- 950W 3+1 Redundant Hot Swap Power Supply
- Custom Configuration is available
- Remote Support Available

Order# SANAS6TBLX02F

6TB NAS starts at **$9,999**

18 Years of Customer Satisfaction
5-Year Warranty, Industry's Longest
First Class Customer Service

AMD 64 Opteron™

888.765.9686

www.Polywell.com/us/LJ

POLY SYSTEMS

AMD64 architecture reduces I/O bottlenecks, increases bandwidth, and reduces memory latency. Critical information gets to those who need it quickly and efficiently.

Polywell Computers, Inc 1461 San Mateo Ave. South San Francisco, CA 94080 650.583.7222 Fax: 650.583.1974

Securing WLANs with WPA and FreeRADIUS, Part I

Upgrade your wireless network from the old, insecure WEP to the new standard—and integrate the authentication with your Linux network. BY MICK BAUER

Are you worried about the security of your 802.11b wireless local area network (WLAN) because you're using plain-old wired equivalent privacy (WEP)? If you're still relying on WEP alone, you *should* be nervous: venerable and well-known vulnerabilities in WEP make it simple for eavesdroppers to crack your WEP keys simply by capturing a few hours' worth of WLAN packets and brute-forcing the flawed encryption used by WEP.

But there's hope! Wi-Fi protected access (WPA) adds new authentication mechanisms and improved encryption key generation to 802.11b, and WLAN products supporting WPA have become readily available. Better still, Linux tools are available for WPA supplicants (client systems), authenticators (access points) and servers (RADIUS authentication servers).

In the next couple of columns, I describe WPA and its component protocols, how they interoperate and how to build a Linux-based WLAN authentication server using the FreeRADIUS server-software package.

Overview

So, what's wrong with 802.11b security in the first place? In a nutshell, 802.11b's WEP protocol has two fatal flaws. First, cryptographic-implementation flaws make it impossible to achieve encryption key strength effectively higher than 40 bits, even if your gear supports higher key lengths. Second, a weakness in WEP's encryption key derivation implementation makes it possible for an attacker to derive a WEP-protected network's WEP secret key—the encryption key used by all clients on the entire WLAN—after capturing a sufficient number of packets.

The pending 802.11i protocol will provide a complete, robust security framework for WLANs. Even after it's finalized, however, it will be some time before this protocol is available widely in commercial products or free software packages.

Enter WPA. WPA adds two crucial components of 802.11i to 802.11b. First, it adds the 802.1x authentication protocol, which provides flexible and powerful authentication capabilities. Second, it adds the TKIP protocol, which provides mechanisms for assigning unique WEP keys to each WLAN client and then dynamically re-negotiating them, such that WEP's key derivation vulnerability effectively is mitigated.

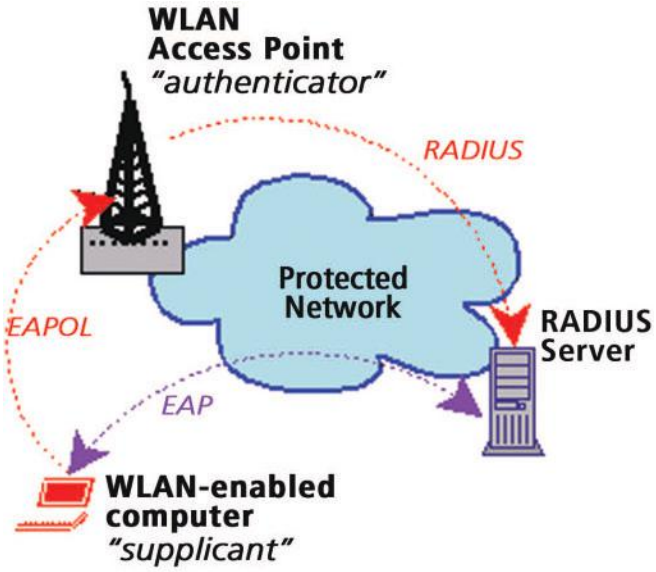


Figure 1. WPA Topology

Figure 1 shows how the various pieces of a WPA system interact. First, we have a WLAN-enabled client system, whose WPA client software is called a supplicant. The client/supplicant connects to a wireless access point (AP), which serves as an authenticator, effectively proxying authentication between the supplicant and a back-end authentication server. In Figure 1, this back-end server is portrayed as a RADIUS server, but TACACS also can be used.

Besides proxying authentication between supplicant and server, the AP/authenticator also feeds data from the authentication server through the Temporal Key Integrity Protocol (TKIP) to obtain a WEP session key. It then pushes the key back to the supplicant. The supplicant periodically is prompted to re-authenticate itself, at which time its WEP key is replaced by a new one.

The authentication (RADIUS) server is optional. Another option is to use pre-shared key (PSK) mode, in which shared keys unique to each WPA supplicant system manually are entered into the AP and used for authentication in lieu of RADIUS. This is better than WEP by itself, because this shared key is not used as an encryption key itself. Rather, it is used to seed TKIP transactions, which in turn provide dynamic WEP keys.

WPA already is supported by a wide variety of new commercial WLAN adapters and access points. It's even been back-ported to some older 802.11b products, thanks to firmware upgrades. In the Linux world, it's supported on the client side by wpa_supplicant (hostap.epitest.fi/wpa_supplicant), on Linux access points by hostapd (hostap.epitest.fi/hostapd) and on the authentication server side by FreeRADIUS (www.freeradius.org).

Before we narrow our focus to building a WPA-ready FreeRADIUS server, which mainly will be covered in my next column, let's look more closely at the authentication and encryption portions of WPA.

WPA Authentication: 802.1x, EAP and RADIUS

Are you following me? Because WPA actually is a bit more complicated than Figure 1 implies. To review: in WPA, your client system (supplicant) must authenticate itself to the network before being allowed to connect, at which point it's provided with a session encryption key that changes periodically.

The reason this gets complicated is the 802.1x protocol used for WPA authentication allows for a variety of methods to authenticate supplicants, which is a good thing. By using a modular, extensible authentication mechanism, the odds are reduced that WPA—or 802.1x or 802.11i—will be made obsolete as particular authentication protocols go in and out of favor. 802.1x's modularity and extensibility is provided, appropriately enough, by the Extensible Authentication Protocol (EAP), of which a number of variants exist. Let's talk about a few of the most popular ones.

EAP-MD5 uses a simple MD5-hash-based credentials exchange. The supplicant provides a user name and MD5-hashed password to the server, and the server compares these to its own database. Unfortunately, an eavesdropper can capture the hash transmitted by a WPA supplicant and run an offline dictionary attack against the hash to deduce the password used to create it. Also, although EAP-MD5 authenticates the supplicant to the server, it doesn't do anything to authenticate the server to the user, for example, with server certificates, à la SSL. EAP-MD5 therefore is a poor choice for 802.1x authentication in WPA contexts.

EAP-TLS uses the TLS encryption protocol, a descendant of SSL, as a basis for authentication. On the one hand, this is a strong authentication method: it requires both the authentication server and its users to have digital certificates, which are the basis of authentication transactions. Issuing digital certificates to a large number of users and managing those certificates, however, can be complex and time consuming. Consider, for example, the time required to revoke certificates of people who leave your organization. Also, EAP-TLS generally requires a complete public key infrastructure (PKI) environment, which few small-to-medium organizations are comfortable supporting. Also, when authentication is initiated, user names are transmitted in clear text, a small but noteworthy exposure.

PEAP (Protected EAP) was developed primarily by Microsoft as a means of using TLS encryption to protect weaker but simpler authentication methods, such as MD5 and MS-CHAP. With PEAP, an encrypted channel is established between supplicants and the server before any credentials are exchanged. This is consistent with the way most Web applications use TLS. That is, they use TLS to establish an encrypted tunnel over which simple user name-password authentications safely can be performed, without going so far as to use TLS's

SO MANY PROTOCOLS!

One of the reasons I'm devoting an entire column to describing how WPA works, rather than simply diving into how to configure FreeRADIUS for WPA, is the myriad protocols and sub-protocols that comprise WPA can be confusing. If you're having trouble keeping all this straight, maybe Figure 2 can help; it shows WPA's protocols in hierarchical form.

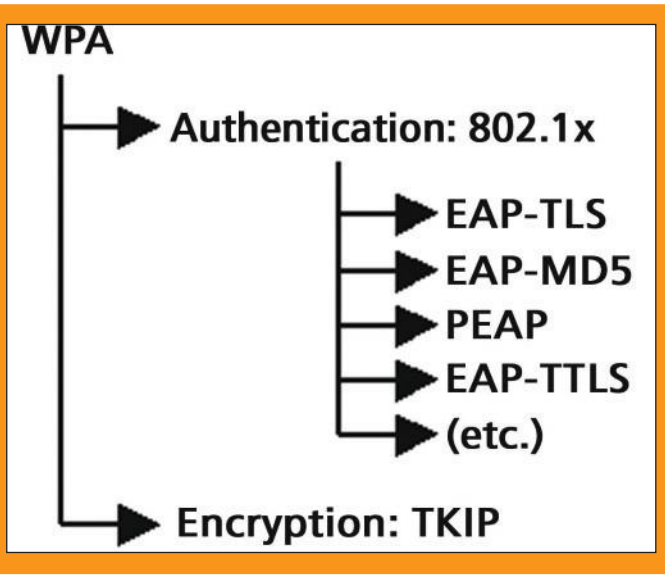


Figure 2. WPA Protocols

more secure but more complicated client-certificate authentication mechanism. The main disadvantage of PEAP is its Microsoft-centricity. Although some free software tools do support PEAP, many people see no incentive for Microsoft to ensure interoperability with other vendors' WPA products or platforms.

EAP-TTLS is, essentially, a non-Microsoft-driven alternative to PEAP. It involves establishing an encrypted TLS tunnel over which either TLS-based or other (weaker) forms of authentication are conducted. Its main advantage over PEAP is being less subject to the whims of one large corporation. It also presently supports a slightly wider range of authentication methods, although PEAP is designed to support more methods than have been implemented thus far. Lacking Microsoft's muscle, some people see EAP-TTLS as not having as much momentum as PEAP.

Other EAP variants include EAP-SIM, Microsoft's EAP-MSCHAPv2 and Cisco's Lightweight EAP (LEAP).

At this point, you might be wondering, "Hey, isn't RADIUS an authentication protocol, too? How does that fit in?" RADIUS is the protocol your authenticator (AP) speaks to your authentication server. In the context of 802.1x and WPA, you can think of RADIUS as the transport over which your authenticator forwards EAP messages to your server. Put another way, your end-user's supplicant speaks EAP to your authenticator; your authenticator forwards those messages within RADIUS packets sent to your server.

There's still *another* protocol at play here, playing a similar role in supplicant-authenticator communications: EAPOL, or EAP Over LANs. This protocol is completely transparent, however, because it's built in to supplicant and authenticator software and requires no configuration of its own. Therefore, there's nothing specific you need to know or understand about EAPOL unless you write WPA software.

From the time a supplicant initiates its connection attempt to the AP, your AP allows *only* EAP traffic. Only after authentication has completed successfully, based on the server's response, is your supplicant system given a DHCP lease and permitted to connect completely to the WLAN. Another consequence of successful authentication, however, is the assigning of a WEP key to the supplicant.

TKIP and WEP Keying

If a supplicant is authenticated by way of EAP-TLS or some other encrypted version of EAP, that authentication traffic also is encrypted. But the wireless LAN frames themselves are not; that can't happen until WEP is enabled on the connection between the supplicant system and the access point. As it happens, from the implementor's standpoint, this is the simplest part of WPA. Upon successful authentication, the server, authenticator and supplicant use the Temporal Key Integrity Protocol (TKIP) to negotiate and transmit WEP keys securely for use between the authenticator and the supplicant system. This process largely is transparent: you do not need to configure anything on the server or supplicant for this to work. However, most access points, including hostapd on Linux, can be configured with custom settings for things such as WEP-re-keying interval.

The other thing to remember about TKIP is, as I mentioned earlier, the server is optional. If you've configured your supplicants and authenticator to use pre-shared key (PSK) mode, TKIP still is used to key and re-key WEP encryption dynamically between your supplicant and access point.

Conclusion (for Now)

That's WPA in a nutshell. Next time, we'll apply these concepts of using FreeRADIUS to create a Linux-based authentication server for WPA. If you can't wait until then to get started, check out the on-line Resources for more information. Be safe!

Resources for this article: www.linuxjournal.com/article/8017.

Mick Bauer, CISSP, is *Linux Journal*'s security editor and an IS security consultant in Minneapolis, Minnesota. O'Reilly & Associates recently released the second edition of his book *Linux Server Security* (January 2005). Mick also composes industrial polka music, but has the good taste seldom to perform it.

servers DIRECT™

More products, better service, guaranteed.

GO STRAIGHT TO THE SOURCE!

SERVERS DIRECT - YOUR DIRECT SOURCE FOR SCALABLE, COST-EFFECTIVE SERVER SOLUTIONS.

1U Entry Level Server
SD1-1103S

Excellent general purpose server for organizations with the need for a low, entry level price

- Intel® Pentium® 4 Processor 2.40GHz 533 FSB
- 1U chassis with 260W power supply
- Supermicro P4SCi board with Intel E7210 chipset
- Kingston 256MB PC2100 DDR ECC RAM
- Seagate 80GB SATA 7200RPM hard drive
- 2 x 1" Hot-swap SATA drive bays
- Integrated ATI Rage XL SVGA PCI video controller
- 1 x Intel 82547GI CSA, 1x Intel 82541 Gigabit Ethernet controller
- RAID 0, 1 supported

$699

1U Application Server
SD1-1302S

Designed for high-performance servers clustering and business critical applications with EM64T solution.

- Intel Xeon Processor 2.8Ghz with 800FSB 1MB Cache (Dual Processor Option)
- Intel Extended Memory 64 Technology
- 1U chassis with 420W power supply
- Supermicro X6DVA-EG board w/Intel E7320 Chipset
- Kingston 512MB PC2700 DDR ECC Reg. RAM (2x256MB)
- Seagate 80GB SATA 7200RPM hard drive
- 2 x 1" Hot-swap SATA drive bays
- ATI Rage XL SVGA PCI video controller with 8 MB of video memory
- Intel® 82541GI Dual Port Gigabit Controller

$999

2U Database Server
SD2-1204C

Cost effective storage in a convenient 2U form

- Intel Xeon Processor 2.80GHz 533 FSB
- 2U Chassis with redundant 400W power supply
- Supermicro X5DP8-G2 server board with E7501 chipset
- Kingston 512MB PC2100 DDR 266 ECC Reg. RAM
- Seagate 36GB SCSI 10K RPM U320 SCA hard drive
- 6 x 1" hot-swap Ultra320/160 SCSI drive bays
- ATI Rage XL SVGA PCI video controller with 8 MB of video memory
- Dual Intel 82546EB Gigabit Ethernet ports
- Support for Zero-Channel RAID with Adaptec 2010S ZCR card (card NOT included)

$1899

3U Storage Server
SD3-1205S

Specially engineer for high-availability and maximum performance for SAN solution up to 3.2TB

- Intel Xeon Processor 3.0Ghz with 800FSB 1MB Cache (Dual Processor Option)
- Intel Extended Memory 64 Technology
- 3U Chassis with 760W Triple-Redundant Power Supply
- Supermicro X6DHT-G server board w/Intel® E7520 Chipset
- Kingston 1024MB PC3200 ECC Reg. RAM
- Western Digital 250GB 7200RPM hot-swap SATA drive w/8MB Cache
- 8 x 1" hot-swap Serial ATA Drive Bays
- ATI Rage XL SVGA PCI video controller with 8MB of video memory
- Dual Intel® 82541GI Gigabit Ethernet Controllers
- Include Zero-Channel RAID with Adaptec 2020SA (SATA) ZCR Card

$2599

4U Enterprise Server
SD4-1301S

Easily scalable storage solution with hot-swap functionality for growing businesses

- Intel Xeon Processor 3.0GHz 533 FSB
- 4U Chenbro RM414 SATA chassis with 460W redundant power supply
- Supermicro X5DPE-G2 server board with Intel E7501 chipset
- Kingston 512MB PC2100 DDR 266 ECC Reg. RAM
- 3Ware 8-port SATA controller card
- Western Digital 250GB 7200RPM hot-swap SATA drive with 8MB cache
- 16 x 1" hot-swap SATA drive bays
- Dual Intel 82546EB Gigabit Ethernet ports

$3099

5U Advanced Storage Server
SD5-1201A

Storage server with 24 hot-swap hard disk bays suitable for 6TB of pure data storage capacity

- Intel Xeon Processor 3.0GHz 533 FSB
- 5U Rackmount chassis with 950W triple redundant power supply
- Supermicro X5DPE-G2 server board with Intel E7501 chipset
- Kingston 1024MB PC2100 DDR 266 ECC Reg. RAM
- 3Ware 12-port SATA controller card
- Western Digital 250GB 7200RPM hot-swap SATA drive with 8MB cache
- 24 x 1" hot-swap SATA drive bays
- Dual Intel 82546EB Gigabit Ethernet ports

$4299

PROVEN TECHNOLOGY FROM YOUR TRUSTED SERVER SOURCE

Your business requires solid server solutions. With Servers Direct server systems based on the Intel® Xeon™ Processor, you can count on high availability, maximum efficiency and proven performance to help you meet your business reliability requirements.

1.877.727.7127 | **sales@serversdirect.com**

Linux on a Small Satellite

With less than a year to design and build a satellite, this team used existing sensor hardware, industry-standard parts, shell scripts and our favorite OS to make the project come together.

BY CHRISTOPHER HUFFINE

The Department of Defense (DoD) Office of Force Transformation (OFT) approached the Naval Research Laboratory (NRL) with an opportunity to build and launch a micro-satellite, of the 100-kilogram class, to provide a platform for a host of technology and operational experiments. A key challenge posed to the Laboratory by OFT was to build this capability in less than one year. Bringing this first TacSat vision together required the development of new partnerships and methods as well as the leveraging of existing hardware, software and facilities.

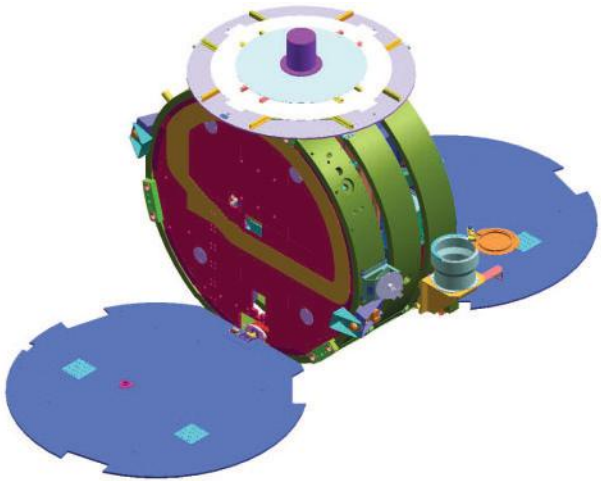


Figure 1. TacSat-1 Spacecraft, Solar Arrays Deployed, Nadir (Earth-Facing) Side Facing Up

Copperfield-2, a sensor system developed by the author's team for the Navy, became the cornerstone of the TacSat-1 payload infrastructure. The Copperfield-2 sensor system (Figure 2) originally was designed for use on unmanned aerial vehicles (UAVs)—a good match for adaptation to a space mission, as many of the design requirements are similar.

A satellite bus can be thought of as the spacecraft vehicle. It provides the physical and electrical infrastructure to support the payload. The satellite payload is the sensor or experiment being carried by the bus. TacSat-1 used a bus originally designed for use in the ORBCOMM constellation of small communications satellites. If Copperfield-2 was flown on an aircraft or UAV, that platform would serve as a bus, providing infrastructure to the payload.

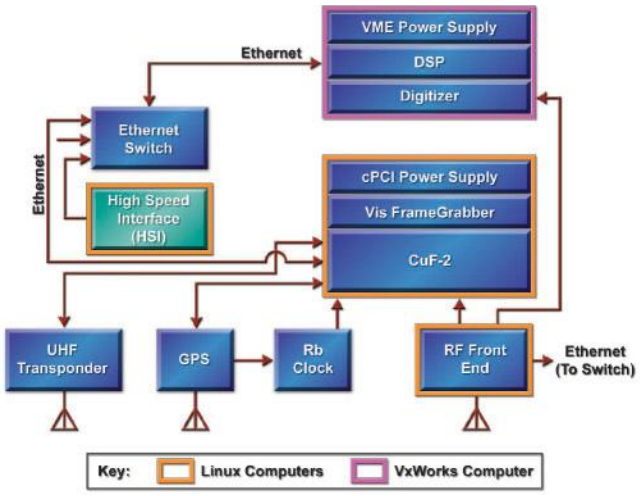


Figure 2. TacSat-1 Copperfield-2 Payload Block Diagram

Modular Payload Hardware Design

The first hardware version of the Copperfield payload was designed from legacy hardware systems and was adapted to allow the original hardware to operate through an Ethernet-connected TCP/IP interface. When trades were made before designing the second-generation experimental capability, various bus standards, commercial off-the-shelf (COTS) emerging capabilities and other factors were considered. We decided to

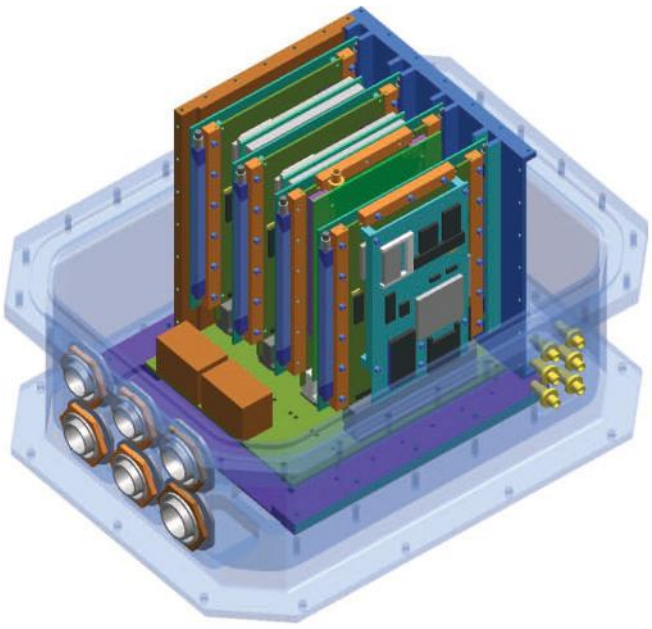


Figure 3. TacSat-1 Copperfield-2 CompactPCI Cardset and Chassis

In fact, multiple general-purpose processors are part of the Copperfield-2 payload, each communicating by way of an Ethernet network.

pursue a 3U CompactPCI architecture to allow maximum flexibility of the physical form factor (Figure 3). However, we decided to use a custom PCI motherboard so the CompactPCI user-defined P2 connector pins could be used for our own purposes. This results in a motherboard with slots that support our custom-designed hardware, slots that support COTS Ethernet switch cards and slots that can accommodate cards built to the PXI standard. The resulting architecture blends standard CompactPCI with Ethernet connectivity available by way of the P2 backplane.

Modular Standards-Based Payload Architecture

Few satellite programs have the latitude or the ability to take the risks that the TacSat-1 experiment has. The TacSat-1 experiment allows innovative leveraging of both government off-the-shelf (GOTS) and COTS hardware components, as well as novel approaches to creating payload software that provide maximum flexibility and standards-based operation. The risk philosophy allowed the utilization of a modular payload hardware. Identically, a modular software and communication system was expanded for TacSat-1, extending the role of standards-based open-source software such that it provides reusable software infrastructure suitable for flexible command and control of the TacSat-1 payload.

The Copperfield-2 payload architecture was intended to provide as much flexibility as possible. It is a testament to the flexibility of the architecture that extension of the UAV payload to a space application was possible. Because the payload software components are not space-flight critical, meaning the health and safety of the spacecraft does not depend on its reliability, much of the software can be leveraged across air and space platforms.

Linux Kernel as the Foundation

From the beginning of Copperfield-2 development, it was our desire to capitalize on the momentum, capability and availability of Linux source code. With the development of the processor card with its PowerPC PowerQuicc II, the hardware infrastructure was in place to support a robust embedded system. The accessibility of source was a paramount feature that allowed us to recover from various situations we encountered, including board layout errors. Although the board design was made to look similar to the Motorola reference design—MPC8620ADS-PCI, which no longer is available—some ambiguities, hardware limitations and other issues necessitated changes to the kernel.

When TacSat-1 development began, many seasoned veterans questioned the choice of Linux as host to the payload control software. Proprietary real-time operating systems typically have been used for space systems developed at NRL. During the architectural design process, no hard real-time requirements were discovered, revalidating the original choice of Linux for Copperfield-2 and, thus, also for TacSat-1.

Beyond the tweaks necessary to get Linux working correctly with our hardware, only three device drivers were written—one to support the sensor data format; one to interface with the Xilinx SystemAce, a CompactFlash interface device that can be used to load FPGAs and also be used for OS storage; and one on the PowerPC 823 HSI interface box communicating with the FPGA. Due to the large Xilinx Virtex-II mapped to the

Table 1. TacSat-1 Copperfield-2 Ethernet-Connected Embedded Systems

Component	Vendor	OS	Processor
High-Speed	Bright Star Engineering (custom adapter board)	Linux 2.4 custom distribution	PowerPC MPC823 Interface (HSI)
IDM UHF Modem	Innovative Concepts	Proprietary	PowerPC 860
Copperfield-2	Aeronix/NRL	Linux 2.4 custom distribution (DENX ELDK-based)	PowerPC PowerQuicc II 8260 MR.DIG Card
RF Front End	Bright Star Engineering (custom adapter board)	Linux 2.4 custom distribution	StrongARM SA1110 Controller

memory space of our PowerPC processor, some innovation was required to handle device driver development in the face of changing FPGA designs. Don Kremer at Aeronix developed a series of utilities that can read Verilog source files and create myriad macros, C code and even HTML documentation that allow the Verilog hardware specification essentially to write the majority of the necessary drivers.

Networking Architecture of COTS Processors

The core Copperfield-2 payload processor provides two key functions for the mission. First, it is a sensor system that receives sensed data, processes the data and interacts with onboard communications equipment to transmit the results to other sensors and ground stations. Secondly, it serves as a general-purpose computer system that provides the infrastructure for storage and data handling. In fact, multiple general-purpose processors are part of the Copperfield-2 payload, each communicating by way of an Ethernet network. A COTS Ethernet switch serves as the center of the star Ethernet architecture.

Gateway to the Bus Legacy Equipment

To capitalize on the Ethernet, TCP/IP, standards-based architecture of the UAV payload while remaining compatible with the satellite bus' legacy OX.25 interfaces—which provide a means for downlinking science data and state-of-health telemetry—a different embedded computer module was designed specifically to serve as the bridge. This module is called the high-speed interface (HSI) and provides a 2MB synchronous serial bus connected to the spacecraft communication controller. The HSI hardware is implemented as a combination of FPGA hardware and a BSE ipEngine general-purpose PowerPC 823 embedded processor.

In the HSI, the FPGA provides the hardware necessary to meet timing requirements for the data link, decoupling the processor from the synchronous data link. The PowerPC runs a Linux 2.4-based kernel, and the HSI FPGA interface is implemented as a standard Linux device driver. No special real-time extensions are used, and a Linux-based application provides the interface between the TCP/IP networking stack, using standard protocols and the device driver implementation. The HSI system allows multiple processes and Ethernet-connected computers to access the data stream sent to the spacecraft. The PowerPC communications controller on the Copperfield-2 processor easily could have handled the HSI tasks on TacSat-1. However, due to the extremely limited availability of hardware and the desire to increase parallel development opportunities, this interface was developed independently.

Rapid Payload Software Development with Existing Tools

The most "custom" part in any satellite program often is the payload control software. Because many of the Copperfield-2 payload components with processors run Linux, interesting software options are available. Much of the payload software was implemented as bash (Bourne again shell) scripts. During the rapid development of the payload software, the philosophy was to attempt to divide the software development into two parts, custom and reused software modules. This philosophy called for minimizing custom code to limited functions and programs with specific purposes. Occasionally, we did find that existing utilities did not quite fit the requirements, and these were modified or replacements were written.

These specific custom programs and drivers allowed for control of payload elements through small command-line utilities that could be tested completely and easily in their limited functionality. These programs were developed with the UNIX command-line functionality in mind, along with data input through standard in (STDIN) and data output through standard out (STDOUT). Developing software utilities with interfaces such as these in mind has been the standard for many legacy operating system concepts from the earliest UNIX developments. We intended to continue that strategy and build upon it, as it provides an amazingly flexible way of constructing thorough capabilities with simple although powerful utilities.

GNU and Open-Source Utilities

The first step in designing the software architecture was to examine what tools already were available to the developers—in this case, parts of the Linux distribution and other GNU and open-source utilities with well-defined pedigrees that provided needed capabilities. Time and time again, as we were developing the payload control software, we were amazed at the flexibility and amazing number of options that various commands provide.

One example is the GNU compression utility gzip. During a ground contact event, the payload streams data in real time

Listing 1. Downlink Pipeline Demonstrating tar, gzip and netcat

```
# Configure the file download pipeline
tar -cf - ${downloadFileList} | gzip -c -l | \
     file_downloader -tqid ${target_qid} -rlp \
                          ${return_link_path} \
                     -dri ${dump_request_id} \
                     -fmt ${dataFormat} | \
     netcat localhost ${!returnLinkService}
```

through a series of software pipes. It originates in a file located on the Flash filesystem and then makes its way through various utilities, including a compression stage, and into the satellite bus. We found that it was necessary to tune gzip to select a compression ratio/performance curve that would ensure that the 1MB downlink was filled completely with data packets. gzip inserted into the downlink stream was a relatively late addition, and it allows us to make maximum use of the available downlink bandwidth. The design of command-line utilities using STDIN/STDOUT interfaces allows capabilities such as this to be integrated transparently into the data stream, within the performance capability of our computer system.

Payload Control Subsystem—with bash

Choosing a scripting language is a difficult task—indeed, in the Open Source community, many competent options are available. Perl may have been a good choice, but we were not comfortable with the size of its installation and memory footprint. Python also would have been a great choice, but the development team did not have experience with it. The most powerful shell-scripting language appeared to be bash, although it also is the heaviest in terms of footprint. Our smallest embedded systems could not handle the entire footprint of bash, but the Busybox lightweight shell-scripting interpreter, ASH, proved almost as capable for the tasks that had to be monitored and controlled on those smaller targets.

Although space here does not allow for a complete architecture discussion of the payload control software design, at its core the software is a series of bash scripts designed to support various functions of the payload. The system is designed to take advantage of POSIX-style filesystem security. Upon boot, the first processes run as root as the system starts. As the payload control software begins to come on-line, it starts up as user BOOT. The system can stay in BOOT and provide a certain number of critical system capabilities, including providing binary telemetry streams, file transfers and direct commands. When a sensor mission is about to begin, the system moves to a state of TRANSITION, and all further data collections take place as the OPS user, who has a different set of permissions.

Listing 2. Sensor Data Processing Pipeline

```
# Start the data processing pipeline
# (with cpf ignoring SIGINT,SIGTERM)
   eval "cat $dig_data_stream | \
      tee $raw_file | \
      cpf -i -v$cpf_verbosity $cpfparams \
      > $output_file &"
   # Enable the dig channel
   set_hardware 'echo $dig_channel \
     channelEnable ena | mapper 2>&1'
```

Listing 3. Example Data Output Pipeline with Conversion to Proprietary Data Format as the Last Step before Sending Out the Data

```
# Start the pipeline
format_event -severity $severity_level \
             -status $status_code \
     -failcmd $fail_cmd \
                -text "${event_text}"  \
                -debug $debug_level  2>> $logFile \
        | ox25  -tbox  ${tbox}  -tque    ${tque}   \
                -sbox  ${sbox}  -sque    ${sque}   \
                -cflgs ${cflgs} -seq     ${seq}    \
                -func  ${func}  -subfunc ${subfunc} \
                -debug ${debug_level} \
                2>> $logFile  \
        | netcat $ncVerbose localhost
${!returnLinkService} \
                2>> $logFile
```

Time and time again, as we were developing the payload control software, we were amazed at the flexibility and amazing number of options that various commands provide.

At the conclusion of the data collection, OPS is commanded to shut down. Multiple redundant copies of the BOOT directories are designed into the system to provide backup capability in the case of filesystem corruption or other significant error.

bash scripts launch every payload control system function. They create complex filenames we use to keep track of configuration, date and time and other information. They un-gzip and untar commands and files that are uplinked to the satellite. Commands themselves also are bash scripts with simplified functionalities. They call other bash scripts to do the actual data collection or to set environment variables that change the behavior of other scripts.

This combination of the bash scripting language, GNU and open-source utilities and custom command-line applications is unique in satellite programs. For TacSat-1, most of the custom code involves the conversion of data from the TCP/IP world to proprietary OX.25 formats to handle sensor data.

Distributed Development and Collaboration

The extensive use of TCP/IP-based systems and the common Linux operating system provided unique opportunities for a distributed development environment. Early in TacSat-1, our custom PowerPC 8260 development hardware had limited availability. The design cycle for much of the payload software began on Intel x86-based computer systems, migrated to generic PowerPC embedded processors and eventually made its way to the final target. The software design team was distributed spatially and tied together through a virtual private network (VPN) architecture. Remote power control devices allowed developers who were operating off-site to cycle power on hardware components. A Web-based collaboration tool allowed the posting and dissemination of critical communications and interconnection control documents (ICDs). Some developers also used instant messaging technology to stay in contact with one another. Recent additions to the collaborative working environment include the use of E-Log to maintain an on-line database of lessons learned. We also are working to integrate Bugzilla capability into the system to replace our relatively crude Message Forum-based problem report (PR) tracking.

The TCP/IP nature of the payload data network allowed developers to test communications between payload elements at each step in the design process, from developing on a standard PC to final communications before inserting the custom hardware required to communicate with the bus. Even after complete integration of the payload into the bus, an Ethernet test port allowed network access to the satellite, which was invaluable for collaborative debugging of the system. Test ports also allow access to serial consoles for most of the payload components and, in some cases, JTAG or other hardware debugging ports.

The payload software design team consisted of experienced satellite and ground station software experts, as well as team members accustomed to the TCP/IP data transport and Web/CGI application development, plus embedded systems experts. Although quite different from the typical satellite software design team, this combination provided nearly the perfect balance of skills and innovative methods to maximize the use of existing software designed for aircraft applications. The extensive remote collaboration, interface testing and networking capability provided a smooth bus-payload integration.

The core of the payload control software, including many of the command and control scripts, were developed in a span of less than four months, from start to finish. Additional scripts were inserted into the core payload control software infrastructure to bring on-line additional sensor capabilities as those sensors became available. New capabilities and patches may be uploaded to the satellite as requirements dictate.

Conclusion

Few satellite programs have the sponsor-supplied latitude or the ability to take risks that the TacSat-1 initiative provides. In this context, the TacSat-1 program allows innovative leveraging of both GOTS and COTS hardware components, as well as novel approaches to creating payload software that provides maximum flexibility and standards-based operation. The modular nature of the Copperfield-2 allowed rapid hardware integration, proving the concept of a modular payload that scales from UAV applications to a spacecraft application, all using Linux and GNU software as a foundation. At the time of this writing, TacSat-1 was scheduled to launch in February 2005.

Acknowledgements

The author acknowledges the essential contributions to the TacSat-1 effort by Stuart Nicholson, consultant, and Eric Karlin, Mike Steininger and Brian Davis of SGSS, Inc., for core Payload Control Software. Brian Micek of Titan Corp., Chris Gembaroski, Don Kremer, Tim Richmeyer and the Copperfield-2 team at Aeronix, Inc., Jeff Angielski of the PTR Group for Linux porting, device drivers and sensor support software. Also, thanks to Wolfgang Denx and the Linux PowerPC community for their contributions toward making PowerPC Linux stable and robust.

Resources for this article: www.linuxjournal.com/article/8066.

Christopher Huffine is an electronic engineer at the US Naval Research Laboratory, working for the Naval Center for Space Technology. He has been using Linux since college on various platforms, from desktop workstations to embedded control computers.

Appro HyperBlade AMD Opteron™ Clusters

Designed for High-Density, High-Performance Technical Computing Applications

- Powerful, modular and scalable architecture
- Blade servers fit all cabinet sizes
- Dual AMD Opteron™ processors per node
- Up to 16GB ECC DDR memory per node
- One PCI-X slot per node
- Up to two internal HDDs
- Remote Management Options
- High-Speed Interconnect Options: Myrinet, Dolphin, Quadrics and Infiniband
- Support for Windows® and Linux Operating Systems

For an overview of how to plan and build a high-performance Linux cluster, get a white paper, sponsored by Appro at www.appro.com.

Appro 1U & 4U - Quad AMD Opteron™ Servers

Cutting-Edge Performance

Appro 1142H : Power-to-Density Server

- Quad AMD Opteron™ Processors
- Up to 32GB ECC DDR 333/400 RAM
- Up to two 500GB IDE, SATA or 292 SCSI
- Slim FDD and DVD / CDRW
- Support for Windows® and Linux Operating Systems

Appro 4148H : Multi-Purpose Server

- Quad AMD Opteron™ Processors
- Up to 32GB ECC DDR 333/400 RAM
- Up to eight 2TB SATA or 1.2TB SCSI
- Four full size PCI-X slots
- 3.5" FDD & 5.25" bay for CD / DVD-ROM or tape devices
- Support for Windows® and Linux Operating Systems

AMD Opteron™ Processors
- Integrated AMD HyperTransport™ technology allows for concurrent multiple processors in a single system.
- Shorten run-time cycles and increase bandwidth for processing computing requests.
- 32 bit applications while you migrate to 64 bit computing for long-term investment protection.

APPRO
HPC Cluster Solutions

Appro has everything you need to create a network Blade cluster-ready.
On-site maintenance and installations services are available.

For more information, please visit www.appro.com
or call Appro Sales at 800.927.5464, 408.941.8100.

AMD 64 Opteron™

The No-Party System

Why Linux isn't really a "platform" and "third party" is a misnomer. **BY DOC SEARLS**

I always have been fascinated by the expression "third party" as it is used in business. What do we mean by that? And, why don't we talk about first party and second party, except in legal documents? Third party clearly labels its members as subordinates to first and second parties. In technology, third parties inhabit a business ecosystem defined by a large vendor (the first party) and its relationship with customers and users (second parties). Wikipedia says, "In computer programming, and particularly in Microsoft Windows programming, 'third-party software component' refers to a reusable software component developed to be either freely distributed or sold by an entity other than the original vendor of the development platform."

By that definition, third-party software plays a value-adding role in a market ecosystem defined by the vendor. In architectural terms, the vendor's role is to provide a platform that supports both second and third parties. Platforms in turn serve as foundations for silos: locked-in market spaces controlled by the vendor.

Linux doesn't work that way, nor does the software we call free or open source. Yet Linux seems to have third parties. Look up "third-party software" on Google and the top result—at this moment, at least—is Third-party Quickcam, a table of Linux resources that's maintained by Patrick Reynolds of the Computer Science Department at Duke University. Here's a fun digression: a search for "Linux" on the Department Web site returns 1,140 results, and a search for "Windows" returns 2,360 results, the first of which is "Emacs/G++ on Windows Machines". It begins, "You'll need three things to make your Windows machine work like a Linux/Unix machine."

The third-party Quickcam software on Patrick Reynolds' list may work with Linux, but it isn't controlled by Linux in the way third-party applications for Windows and OS X are controlled by Microsoft and Apple. That's because Linux isn't a company. As one Linux programmer once told me, "Linux can't sue anybody." In fact, Linux isn't even a platform of the sort defined by Windows and Mac OS. Instead, Linux is a form of building material that grows in the wild and naturally is suited for making foundations and frameworks. The wild in this case is fertile human mentation, which is why it evolves and improves in the course of being put to use.

The limits to Linux's usefulness also are natural ones. No company restricts anybody's right to use it. Because Linux embodies and expresses the GNU General Public License (GPL), Linux is not only free as in beer and free as in freedom, but free as in marketplace. Both Linux and the software that runs on it are unconstrained by formalized business relationships defined primarily by one party. The primary practical purpose of free software is to be useful, not to serve as a platform for a silo—even if platform vendors build silos on it anyway, as, for example, Apple's OS X does on FreeBSD. Hey, it's a free market.

For years I've been carrying around a discomfort with the platform label for Linux and the third-party label for software that runs on it. That discomfort verged on pain when I wandered around CES (Computer Electronics Show) in Las Vegas, January 2005. Shows such as CES and MacWorld, which followed it, provide an interesting contrast to Linux events, because they gather exhibitors that seem to operate on a set of principles exactly opposite of those Linux holds.

At CES, for example, proprietary is a *good* word, and digital rights management (DRM) is a *good* feature. At CES, I lost count of the times somebody reciting a scripted pitch on a vendor's stage bragged about "our proprietary technology". Red Hat, Novell, IBM and Sun—none of whom were at CES, for whatever that's worth—all have proprietary technologies, some more than others, but you never hear them brag about it at LinuxWorld Expo.

I was fascinated to hear about one vendor or another "owning" a market or "dominating" a category. What does this mean for third parties in owned or dominated categories? It seemed to me that their role, in spite of whatever success they might achieve, still is a captive one, like a prisoner or a slave.

In the central halls of CES, it seemed as though every product category—audio/video, satellite systems, mobile electronics, home theater, HDTV, digital cameras and camcorders, to name only a few—were collections of silos that I couldn't help but think of as prisons. For audio recording, Sony had ATRAC. For digital IBOC (in-band, on-channel) AM/FM radio, Ibiquity had HD Radio. For "digital lifestyle" home PC/TV integration, Microsoft had Windows Media, the Digital Media Edition of Windows XP and a raft of other closed and proprietary products. Microsoft partners all over the floor carried the Microsoft PlayForSure logo, which serves two purposes: 1) labeling third parties as members of Microsoft's branded ecosystem and 2) sugar-coating the DRM in Windows Media Player. The satellite radio (XM, Sirius) and television (DishTV, Voom, DirecTV) vendors were silos in themselves. Third-party antennas, receivers and other devices all are built precisely to specifications provided by the vendors.

Linux was all over the show, however, although few vendors were willing to talk about it, much less brag about using it. When I went looking for Linux stories at one name-brand network equipment company, the head media relations guy was summoned to tell me, with practiced precision, "We can't talk about that." When I pressed him, his answers made it clear that the company's publicity ports for Linux and open-source information were blocked by the legal department. When I pressed harder, the guy finally said, "Okay, I'll tell you this much. You can't throw a stick at anything in this booth and not hit something that runs on Linux."

The biggest booth at the show was a collection of large rooms off the Central Hall in which Sony showed off its latest and greatest. If there was any Linux in those rooms, you wouldn't know it from Sony's literature or hear about it from Sony staffers. The official Sony policy on Linux and open

source appeared to be stony silence—in spite of a pro-Linux keynote given by Sony COO and President Kunitake Ando at CES two years ago. “There’s no Linux here”, one Sony guy told me, as if I had showed up at Tiffany’s asking for whiskey.

When I asked another Sony guy if the company ever would make a portable digital audio device that could record and play back Ogg, MP3 or formats other than the company’s own highly proprietary ATRAC, he said “Oh no. There are copyright issues with those.” When I asked him to explain those issues, he mumbled something about people “stealing music”. I told him Apple’s iPod was not only kicking Sony’s butt in the portable audio market but was capable of recording in MP3. He said there was nothing he could do about that. He did assure me that Sony had no plans to make an MP3 player.

The reason behind this stance, of course, is Sony isn’t merely a consumer electronics company: it’s a music company as paranoid about “piracy” as the rest of the tired old recording industry. But, it doesn’t need to be. Sony is crippling its legacy business—electronics—to keep one of its acquired businesses—music—from getting hurt. This allows innovative and unconstrained competitors, such as Apple, to clean up in a category Sony probably would dominate if it wasn’t simply a collection of battling business units whose conflicts are settled by lawyers.

The largest presence at CES was the absent exhibitor whose own show followed the next week in San Francisco—Apple. In October 2004, the NPD group said Apple’s iPod accounted for 92.1% of the market for hard drive-based music players. In online music retailing, Apple’s iTunes Music Store is equally dominant. Thanks to iTunes and iPod, the hardware extension of iTunes software, Apple is becoming the Microsoft of Music. And without music, there wouldn’t be a consumer electronics business. iPods were everywhere at CES. And although Apple is far less supportive of third parties than is Microsoft, it does have a few. One is Motorola, which plans to come out with an iTunes phone. Others are Belkin, BMW, Mercedes and countless makers of cases, attachments and various iPod accessories. More than one exhibitor told me that many of the new Microsoft PlayForSure partners were motivated by fear of Apple’s success with iPod and iTunes and the relative exclusivity of Apple’s partnership requirements.

All of which is interesting, but beside the point. The point is how Linux and the pioneering values of its companions quietly are changing the world.

While everybody else watches battles among market fortresses, pioneering developers quietly open and settle the wide open spaces where freedom reigns. We see it happening in embedded operating systems; TiVos, Replay TVs and countless network appliances at CES all run on Linux. We see it happening with radio, in podcasting and with music recording, thanks to Creative Commons-licensed artists and music. What’s next?

The day after CES, I ran into Alan Graham, a programmer and author. His latest book is *Never Threaten To Eat Your Co-Workers: Best of the Blogs*, for which I wrote the foreword. He told me we can expect the same progress to happen with television:

> Television is just information and information wants to be free. All it’s going to take to free it is one guy with one new inven-

> tion, one new cool implementation.
>
> Go back 75 years to the early days of radio. People complained, "You can't do music on radio!" But guess what? Radio sold music.
>
> Technology *builds* markets. Point to any technical breakthrough in media, and you can point next to a market that got created by that breakthrough. Look at videotape. Netflix. Blockbuster. Little independent video rental places. Did the VCR kill the movie market? No. It created a new market for movies. The same thing will happen to television.

To seek relief on the last day of the show, I went over to the newly renovated Alexis Park Hotel, which for the last several years has been the home of the High Performance Audio corner of CES. In the old days, exhibitors suffered exhibiting in the cavernous and noisy main halls. In the Alexis Park, each exhibitor has its own small hotel suite. Sound isolation is remarkably good, considering.

I went there looking for Linux stories and also because, many years ago, I was an audiophile. This was back when vacuum tubes were going out of fashion; they're back with a vengeance now. I built Dynaco pre-amps and power amps from kits and knew the virtues and failings of countless brands of turntables, amplifiers and tuners. I could never afford to be a high-end customer and still can't, so I did the next-best thing—retailing. I worked as a salesman and a manager at several audio "salons", as they called them back then.

Although I expected to see and hear some far-out and high-priced audio gear at the Alexis Park, I didn't expect to it to be a delightfully silo-free zone. As with the freelance Linux hacker ecosystem, high-end audio is inhabited mostly by smart and resourceful do-it-yourself builders, all making whatever they feel like making, any way they want to make it, without restrictions by any "platform" vendors. Instead, they all regarded the big-name vendors, Sony, Technics, Bose—everything sold in Circuit City and Best Buy—with disdain.

What's more, these gear hackers all were pursuing perfection—they call it that—with products built mostly from standards-based components and in a mostly open way. They bragged and argued about approaches, implementations and results, in large measure because their materials and building methods are open to inspection and discussion. Not surprisingly, this included their use of Linux. Rodomir "Boz" Bozovic, PhD, of Tact Audio Labs told me his shop uses Linux in its pursuit of "acoustical room correction, measurement and monitoring". Mark Doehman, Chief Designer at Continuum Audio Laboratories in Victoria, Australia, told me the company's radical-looking turntable benefited from software that did "wave shaping" and other stuff that sounded cool but I don't remember. With luck they'll make it into a future story in *Linux Journal*.

My favorite component was the RCA 833A vacuum tube, which is the size of a pickle jar and was a workhorse for decades in radio transmission and industrial heating applications. A number of speaker makers drove boxes the size of coffins that cost more than luxury cars with WAVAC HE-833A single-ended monoblock amplifiers, which sell for $38,000 US. One speaker maker told me, with pride and admiration for the WAVAC, that the 833A tube costs less than $50. Like every other amplifier I saw at the Alexis Park, it was differentiated by the quality and uniqueness of design, construction and, especially, by the unique personalities behind the products. Sound familiar?

So, what does this say about Linux and third parties? I asked Jeff Wiegand, a veteran independent Web developer now working for the St. Louis City Government, if the term third party makes any sense to him. "It's only manufacturers and clients now", he replied. Then he went on to define manufacturer as "anybody who makes anything that's useful."

I did find some other examples back in the main halls at CES. For example, I had long conversations with several executives at Frey Technologies, which makes SageTV media centers. Among other things, they were launching a new Linux version of the company's media center that "offers the reliability and affordability of Linux without Windows licensing fees or the more expensive hardware required to deploy Windows-based systems". CEO Dan Kardatzke told me the company started out working with Microsoft but decided there was far more room to grow and compete outside the Windows silo. "It was an economic decision to begin with. The OEM cost of MC—the Media Center edition of Windows XP—is $89 US. But there are also these really high hardware costs, for processors and graphics chip sets and so on. We can run on a 600MHz Pentium III. There's also stability, reliability, networkability...."

Alan Graham also told me home entertainment battles will be won, eventually, by the most open systems. He finds hope, for example, in the relatively open ecosystem surrounding the Linux-based Replay TV:

> Replays are just wonderful—far more flexible and capable than TiVos. They have lots of inputs on the back and lots of ways they let you control them, rather than vice versa. You can have several Replays in your house, plug them all into a 100baseT network or a wireless one, but you want wired for speed. You can swap out or add bigger drives. And you can hack the whole thing into one big system with DVArchive, which is a free Java program you can run on Linux or anything else. You can set all your Replay recording schedules for whatever you want to record. You can set DVArchive to move videos off the Replays and onto your central server to archive there.
>
> You can also use the VLC media player to play them. VLC is free open-source software. It runs on every platform you can name, including all the Linux distros and even little handheld Linux devices. It recognizes the Replay format, pulls off two reference files and the video file. The beauty is you can take these Replay files and play them in the VLC player, on the go. If you can plug in an Ethernet cable and do minimal command-line work, you can do home entertainment automation. You can build a video server system. Today. So much stuff is already here. Not just Replay, DVArchive and VLC, but proximity through Bluetooth and presence through XMPP. Consider the possibilities.

It's a lot easier to consider those possibilities if you're a pioneering member of the No-Party system.

Resources for this article: www.linuxjournal.com/article/8067.

Doc Searls is Senior Editor of *Linux Journal*.

Appro XtremeBlade Solution

Built in Class. Built to Scale.

The Most Powerful 2-Way, 4-Way and/or 8-Way Infiniband™ Enabled Blade Solution

✔ **Designed for High-Availability and High-Performance Data Center Applications**
✔ **Fits up to 12 Blades in a 7U Height Subrack**

- **Scalable** - 2-way, 4-way and/or 8-way configuration
- **Flexible** - Ability to mix blade configurations in one rack cabinet
- **Connected** - Integrated Infiniband™ and Gigabit switches
- **Powerful** - AMD Opteron™ based server with leading 32-bit performance with 64-bit capabilities
- **Reliable** - Hot swappable blades, redundant switches, power supplies and cooling fans
- **Smart** - BladeDome II – Centralized remote blade server and system management
- **Balanced Architecture** - Memory, I/O and communication bandwidth match CPU bandwidth

AMD Opteron™ Processors
- Integrated AMD HyperTransport™ technology allows for concurrent multiple processors in a single system.
- Shorten run-time cycles and increase bandwidth for processing computing requests.
- 32 bit applications while you migrate to 64 bit computing for long-term investment protection.

APPRO
HPC Cluster Solutions

Appro has everything you need to create a network Blade cluster-ready.
On-site maintenance and installations services are available.

For more information, please visit www.appro.com
or call Appro Sales at 800.927.5464, 408.941.8100.

AMD 64 Opteron™

Finding Your Way with GpsDrive

Lots of tools can plot your position on a map, but this one displays your friends' positions, enables multiple map sources and more. **BY CHARLES CURLEY**

The Egyptians invented geometry, the mathematical basis of surveying. The Nile's annual floods removed markers and forced those tidy bureaucrats to re-measure roads, fields and other features of the landscape.

Gunpowder came to western hands, and long-range artillery was invented. This required precisely locating naval and artillery guns, as well as their targets. So, the military has had a longtime interest in the art of locating things, and they have refined the techniques that the Egyptians first pioneered.

In the 1970s, the US Department of Defense (DoD) started work on the Global Positioning System (GPS). This put a constellation of 24 satellites in low-Earth orbit. GPS allowed instantaneous fixes accurate to within a few tens of meters. The Soviets launched a similar system, Glonass, which Russia still maintains. And, the EU has begun work on an improved system of its own, Galileo, to be deployed in 2008.

The military is happy; they now can locate targets with much greater accuracy. However, as with another DoD project, the Internet Protocol, the civil spinoffs may far outweigh any military benefits. We can now use GPS to locate errant hikers, help distressed vessels and search for oil wells far more precisely and cheaply than with previous techniques. Indeed, the EU sees Galileo primarily as a commercial venture.

All three systems are based on atomic clocks aboard the satellites. The receiver uses time signals to tell its distance from each satellite. Spherical geometry tells us that three satellites give a fix in two dimensions. A fix in three dimensions requires a minimum of four satellites. Modern GPS receivers can track as many as 12 satellites, the most they can see at any one time.

Because of the frequencies and signal strengths at which GPS operates, the major constraint on GPS receivers these days is that one must be outdoors, or nearly so, or have a remote antenna, in order to track satellites.

What Is GpsDrive?

GpsDrive is a program licensed under the GNU General Public License (GPL) for displaying one's position in real time. It operates on most laptops running Linux, and on Linux-driven PDAs, such as the Yopy and Zaurus. Currently, 12 languages are supported.

Before we begin, a word of warning: never consider GPS as anything but an adjunct or supplement to other tools of navigation. The advent of GPS is not occasion to dump your copy of Bowditch.

Getting It Running

GpsDrive requires the Gnome Toolkit plus (GTK+), version 2.2 or higher, which comes with most Linux distributions. Anti-aliasing fonts are nice but not required.

MySQL can store waypoints, and GpsDrive will automatically use it if possible.

Kismet is a wireless sniffer, a tool for detecting Wi-Fi access points. As Kismet detects them, GpsDrive automatically turns the contact information into waypoints and stores them in MySQL. This turns GpsDrive into an excellent tool for wardriving.

Festival is a voice output program for Linux. GpsDrive uses it for voice delivery of comments as you approach waypoints. It is an excellent safety feature for mobile GpsDrive users. Flite is a stripped-down version of Festival.

Installation

Installing GpsDrive is straightforward for those familiar with typical package installation.

Get GpsDrive from its home page or mirrors indicated on its Web site (see the on-line Resources). You can get tarballs, md5sums and RPM packages for the latest stable versions. You also can get the latest work-in-progress quality version from anonymous CVS. The tarball version is the more flexible, as you can remove some of the components you don't plan to use.

To install a tarball, copy it to a suitable location. Then do the following:

```
tar -xvzf gpsdrive*tar.gz
cd gpsdrive
./configure
make
```

If you are using only the NMEA protocol and don't need the GARMIN protocol, configure GpsDrive with:

```
./configure --disable-garmin
```

You can append `--enable-auto-optimization` for optimized compiler flags.

Then, as root, install the program, the gpsd dæmon and the language files. Run:

```
make install
```

RPM installation is the usual:

```
rpm -ivh gpsdrive*.rpm
```

Once installation is complete, you should be able to read the man page, which has the latest information.

The first thing to do is to see if GpsDrive works with your GPS receiver. To test the system, fire up gpsd, a dæmon that serves the raw GPS data. It will listen on /dev/gps, unless you tell it otherwise on the command line with the -p option:

```
gpsd -p /dev/ttyS1
```

Because you should run GpsDrive and gpsd as a non-root user, make sure that user has read and write permission on the device.

Once gpsd is running, run:

```
telnet localhost 2947
```

When you get the connect message, press the R key, and gpsd will start feeding you raw NMEA sentences, like so:

```
[ccurley@charlesc ccurley]$ telnet teckla 2947
Trying 192.168.1.32...
Connected to teckla.
Escape character is '^]'.
r
GPSD,R=1
$PRWIRID,12,01.05,07/29/96,0003,*46
$GPRMC,235947,V,4333.1694,N,10812.0068,W,0.000,0.0,120895,13.3,E*42
$PRWIZCH,00,0,00,0,00,0,00,0,00,0,00,0,00,0,00,0,00,0,00,0,00,0*4D
ASTRAL
ASTRAL
$GPRMC,235949,V,4333.1694,N,10812.0068,W,0.000,0.0,120895,13.3,E*4C
....
GPSD,R=0
^]

telnet> quit
Connection closed.
```

This works even when the receiver can't get any signal, because the receiver will send data indicating that it doesn't have any signal.

Once you know which device your GPS receiver is on, make a symlink (as root) to /dev/gps so that gspd or gpsdrive can use the default:

```
ln -s /dev/ttyS0 /dev/gps
```

You can set the device name in the GpsDrive GUI, but gpsd won't use that setting.

If you are going to use MySQL for waypoint storage, which is required for Kismet, see the file README.SQL. You need to feed the file create.sql into MySQL's command-line client, so you must have appropriate permissions in MySQL. You can use any reasonable MySQL client to edit your waypoints, including OpenOffice.org.

Firing Up GpsDrive

Once you have GpsDrive and any optional software you want installed, and you know the GPS receiver is working, try GpsDrive. You will see a splash screen, then the main window. Then you will see one nag screen for the first and last time. The author, Fritz Ganter, pays for the server for the Web page out of his own pocket and would appreciate your contribution.

Once you close the nag box, you should see an image in the map section of the GpsDrive window. This is a placeholder until you get a map for yourself. The first thing to do is turn off simulation mode in the Preferences menu. While you are there, if you want statute or nautical miles, select that option.

To get your first map, determine the latitude and longitude of the center of your new map. Then put the program into position mode (lower-left area of the menu). Next, create a waypoint with the X key, and enter the lat and long of the map center. Use minus signs to indicate south and west (Figure 1).

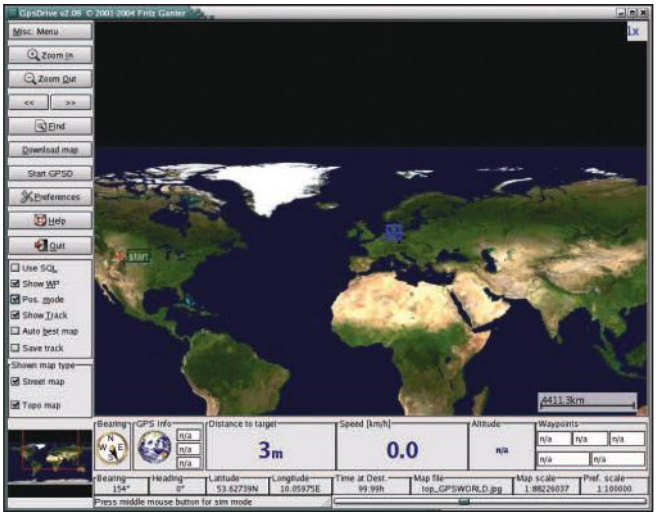


Figure 1. The Main Window on the First Use of GpsDrive

Use the find tool (upper-left menu) to go to the waypoint. Now, click the Download Map entry on the left side of the main window. You will notice that your lat and long are the defaults. Select your scale and source, and grab a map. Bingo! The new map is displayed immediately. If this is a location you use a lot, you may want to download several maps at different scales.

GpsDrive Modes

GpsDrive has three modes: position, normal and simulation.

Use position mode to move around on your maps. Enter position mode by checking Pos. mode on the lower-left side of the main window. Once you are in position mode, as you jump around by clicking on the map, GpsDrive shows you the distance and bearing from the current position (marked with a blue square) to the target (indicated by an alternating red and blue cross).

For example, once you have a small-scale map of a large area, you can move around and download selected large-scale

maps for interesting locations. You also can define waypoints using position mode.

In normal mode, GpsDrive has a fix from a GPS receiver and is tracking the position indicated by the receiver. As the position changes, GpsDrive pans across its supply of maps. GpsDrive comes up in normal mode.

In simulation mode, GpsDrive generates a path from a starting point to one or more waypoints. To enter simulation mode, bring up Preferences, go to the first settings tab and check Simulation. This is a fun mode, as you get to watch an imaginary vehicle move at high speeds across the countryside.

Getting Maps

You will want several maps in different scales. I recommend you get a very small-scale map that covers all of your normal travel area. With this in place, you won't fall off your map if you accidentally click outside your area in position mode. The NASA maps (if you have the disk space) or the default map do this nicely.

In the GUI, you simply select the parameters for the map you want, and the server, and then get it. That's the easy way. However, the results may not tile well. You can get US Geological Survey maps from Topozone.com or street maps from Expedia.com.

If you know the latitude and longitude of the center point and the scale you want, enter these into the download map dialog and go. You also can enter position mode and click on existing maps until you get to the center of a new map you want and then download it.

Then, there is NASA topographical data. See the file README.nasamaps for details and Figure 2 for an example.

For a more systematic map collection, see the accompanying gpsfetchmap.pl.

A Note on Copyrights

Some of these map sources provide copyrighted data. Be sure you use the maps in a manner consistent with the permissions granted on the Web site.

Importing Your Own Maps

You also can import your own maps. You need to know the latitude and longitude of the center point and the scale of the map. There is a druid to help you import maps under the Misc. menu in the top-left corner of the GpsDrive window.

Using GpsDrive

Now that you have some maps, it's time to play around with your new toy.

GpsDrive is well supplied with tool tips, so we only cover the highlights of the display here.

Right below the map in the main window, GpsDrive displays navigation data. Distance to the next waypoint and current speed are obvious. To the right of those is some information on waypoints, mobile targets visible on your friend's server, and the current time according to the GPS receiver.

To the left of the distance to waypoint display is GPS information. With no GPS, a rotating globe is shown. When a GPS is present, the globe is replaced by a signal strength meter for visible satellites. Its background is red if there is no fix; green if there is a fix.

To the left of the GPS data is a compass. The top of the compass indicates your current heading or the course you are sailing. The black pointer gives a bearing to the next waypoint.

A lot of settings are handled in the Preferences menu, which you can select from the left side of the main window. You already know about selecting your units of measure. If you are operating with an older computer, you may want to limit the amount of CPU time GpsDrive takes up, and turn off shadows, which require extra processing to draw.

In the second settings tab you will find some GPS-related settings. For example, you may elect to have GpsDrive access the receiver directly instead of through gpsd.

The SQL tab lets you select certain types of waypoints to include or exclude from the display. This lets you organize waypoints into categories and decide which ones to display. I use this with a set of waypoints for my preferred gas station chain. I can turn them on or off on the display, depending on whether I am looking for gasoline or not.

Once you have maps in hand, there are several controls you can use to manipulate them. For areas where you travel a lot, you probably have maps of several different scales. There are several ways to select between them. The first is to check Auto best map in the lower part of the left menu. This tells GpsDrive to select the best (largest scale) map available for the current location.

Below that, right above the area map, you can check on street or topographical maps, or both. With both checked, GpsDrive moves between the two types, which gives you the most coverage for the maps you have.

Turn Auto best map off and you have several ways of selecting scale. In the upper-left area of the main window, you will find two arrows. Click on the left arrow to move to a larger-scale map,

Figure 2. Southern New England shown by GpsDrive using NASA topographical data.

on the right to move to a smaller-scale map. You also can move the slider on the very bottom-right side for the same effect. This sets the preferred scale, and GpsDrive stays as close to that scale as it can.

Within a given map, you also can zoom in and out. Use the two magnifying glass controls on the upper left of the main window. The current magnification is indicated in the upper-right corner of the main map. GpsDrive keeps the same level of zoom when it changes maps, which can be disconcerting.

First, make sure you have waypoints turned on and that you are using SQL or not, as appropriate.

There are several ways to set waypoints. You can hand-edit them into the text file or MySQL database, you can use the program gpsbabel to convert from other file formats or you even can download them from Wayhoo.com.

In position mode, you can enter a waypoint at the current position by pressing the X key, or you can enter a waypoint at the current mouse pointer with the Y key. You always can edit the parameters before you commit the waypoint.

Wardriving with GpsDrive

Wardriving is the sport of driving around searching for Wi-Fi access points. For more, see the article "Discovering Wireless Networks" in the September 2003 issue of *Linux Journal*.

Got Friends?

GpsDrive comes equipped with a friends server. This lets several people display each others' positions on their systems. You can run your own, or you can use any one you can find on the public Internet. This is real-time plotting of multiple vehicles' positions. This makes GpsDrive a great adjunct to a car rally or search-and-rescue mission.

If a user falls off the Net temporarily due to Wi-Fi signal loss, the user's last known position is displayed. Once he or she is back on the Net, displays are updated in seconds.

Missing from GpsDrive

About the only thing missing from GpsDrive is street-level routing. To do this, the program needs an open source of street-level data. Commercial data usually runs in the area of 10,000 Euros, which is a showstopper. If you know of such a data source, please let the author know.

Language Support

GpsDrive needs localization, especially for Festival. Volunteers?

Conclusion

GpsDrive is an excellent tool for displaying the positions of one or more GPS receivers in real time. It is suitable for several applications, from fun stuff like tracking a Sunday afternoon's exploration to serious work like search and rescue.

Resources for this article: www.linuxjournal.com/article/8068.

Charles Curley (www.charlescurley.com) teaches Linux at two Wyoming colleges. He also writes software and articles and books, using open-source software tools such as Emacs.

Building Your Own Live CD

Create your own special-purpose live CD distribution with these little-known secrets of bootable CDs.

BY DANIEL BARLOW

You've probably heard of Knoppix, the Debian-based distribution that squeezes 2GB of applications on a single standalone CD. It's been used as a Linux demonstration tool, a rescue disk and even as a Debian installer. It's inspired a small raft of related projects, ranging from CDs containing Knoppix, plus or minus a few extra packages, to complete re-architectures of the system.

I recently set out to produce a live CD for a product demonstration. I started by taking the Knoppix CD apart to see how it ticked, and I ended up with a Makefile and a few ancillary files that are clearly Knoppix-inspired but have little derived code. This is what I learned.

A Brief Tour

If you put the Knoppix CD in a CD-ROM drive and mount it, you soon notice that it doesn't look much like an ordinary Linux installation. There are a few graphic files and a free music track, but no init, no /dev and no /bin. The magic is in the big file called /KNOPPIX/KNOPPIX, an ISO9660 filesystem image compressed for the cloop device.

The standard loop device in the kernel allows you to access a file in some filesystem as if it were a device; requests for blocks of the device are mapped to requests for blocks in the underlying file. Because you can mount the device, this effectively means you can create images of filesystems and access them as if they were real hardware disks. If you downloaded Knoppix from the Net, you have an ISO9660 image that can be loop mounted to look at its contents:

```
# mkdir /tmp/knoppix-cd
# mount -o loop -r \
$HOME/KNOPPIX_V3.3-2003-09-24-EN.iso /tmp/knoppix-cd
```

The cloop compressed loop device takes this a step further. In this adaptation of the loop device, each block is compressed with gzip and transparently decompressed when it's accessed. /KNOPPIX/KNOPPIX is an image for this device that is mounted during startup—this is how Knoppix gets 2GB onto a 650MB CD.

You don't need to install cloop in your usual kernel if you simply want to look around the inner filesystem. Install the cloop-utils package and use extract_compressed_fs, as shown below. You need about 2GB of free space in /var/tmp or wherever you decide to put the image:

```
# mkdir /tmp/knoppix-cloop
# extract_compressed_fs \
   /tmp/knoppix-cd/KNOPPIX/KNOPPIX \
    >/var/tmp/KNOPPIX-cloop
# mount -o loop /var/tmp/KNOPPIX-cloop \
   /tmp/knoppix-cloop
# find /tmp/knoppix-cloop -print
```

You can look, but you can't touch—the ISO9660 filesystem is read-only. To modify the distribution, you first need to copy both filesystem images to ordinary directories:

```
# mkdir $HOME/my-knoppix-tree  \
    $HOME/my-knoppix-cd-tree
# tar -C /tmp/knoppix-cloop -cf - . | \
    tar -C $HOME/my-knoppix-tree -xvpf -
# tar -C /tmp/knoppix-cd -cf - . | \
    tar -C $HOME/my-knoppix-cd-tree -xvpf -
# umount /tmp/knoppix-cd /tmp/knoppix-cloop
```

Now, you can hack away to your heart's content. The most convenient way to do this is to change root into the Knoppix inner tree using the chroot command:

```
# mount -t proc none $HOME/my-knoppix-tree/proc
# cp /etc/resolv.conf \
$HOME/my-knoppix-tree/etc/resolv.conf
# chroot $HOME/my-knoppix-tree /bin/sh
```

From here, you can use all the usual Debian package management commands (dpkg, apt-get and so on) to install or delete whatever you like. When you're done, exit the chroot and unmount proc, unless you want your development system's process list immortalised on CD. Then, use `create_compressed_tree` and `mkisofs` to create the inner and outer images:

```
# mkisofs -L -R -l -V "KNOPPIX ISO9660" -v \
-allow-multidot $HOME/my-knoppix-tree | \
create_compressed_fs - 65536 > \
$HOME/my-knoppix-cd/KNOPPIX/KNOPPIX
```

```
# mkisofs -l -r -J -V "KNOPPIX with local stuff" \
-hide-rr-moved -v -b KNOPPIX/boot-en.img \
-c KNOPPIX/boot.cat -o knoppix.iso \
$HOME/my-knoppix-cd
```

Finally, burn knoppix.iso to a CD-ROM and boot it. If you prefer, you can test without burning by using Bochs or VMware.

Further In

This simple approach starts to break down, however, when you want more extensive customizations. For example, if you want X to start a particular window manager but don't want to use all of GNOME or KDE, you have to edit the script yourself. This isn't hard to do, but it means that you've essentially forked Knoppix. When a new Knoppix version comes out, you'll have to do it again. In addition, if you intend to sell your Knoppix-based CD commercially, you need to remain compliant with the licenses of all the software you distribute, which means knowing exactly what's on it. The Knoppix version I looked at contained some files that weren't from Debian packages, and sometimes they weren't even free software.

So, is there some other place we could start? Happily, yes. Between the efforts of Progeny, which donated its installer to the Debian Project; Klaus Knopper, the author of Knoppix and the creator of the cloop device; and other Debian developers who are working on adding his custom code into the main Debian repository—today we can put together a passable live CD system from scratch using only Debian packages. The rest of this article describes how.

Downloads

A tarball containing all the scripts and files referred to here can be found at ftp.linux.org.uk/~dan/livecd. Due to space limits, here, most of the code is not reproduced in the article itself. It's mostly Makefile-driven, with some shell scripts and some simple Perl, and it should be pretty easy to follow. You may hit a few snags if you're not using Debian. If you make it work with some other host distribution, be sure to send patches.

The debootstrap program provides the Debian base system from which you start. Given a Debian release name and a package mirror URL, debootstrap downloads and installs the base system into a subdirectory of your choice. This is pretty flexible; you can chroot into it, use it as a UML root or, if the subdirectory you chose was on its own filesystem, reboot your computer and use it directly. You even can burn it onto a CD, which is what we are going to do. We have some work to do first, though.

Expect to do quite a lot of debootstrap and package installation as you test your scripts. Before going much further, save yourself some time and bandwidth by installing a proxy package archive (such as apt-proxy) on a convenient machine.

Adding Packages

The fix_inner target in the Makefile adds packages to the base system. The first thing we do is replace start-stop-daemon with /bin/true to prevent post-installation scripts from running services in our chroot. With that done, we chroot into the system repeatedly and run such commands as apt-get and dpkg.

For testing and experimentation, we also have a Perl script, run-chroot.pl, that simulates a system boot in the chroot area. It doesn't start most of the services, because they're already running on the host and would conflict, but it does run an SSH server and the X startup script. This is a lot more convenient than writing a CD and rebooting whenever we want to test something.

autologin

There's no point in making people log in on a single-user demonstration system. You have to tell them the password anyway, and the CD is read-only so they can't change it beyond the current session. GDM has an autologin feature, but to keep the image size down we want to avoid dragging in all the GNOME dependencies. Instead, we simply use `su` to start X as a non-root user and run the .xsession script, which opens an xterm and Emacs and starts our application. The autologin-x script is installed as /etc/init.d.autologin-x, with appropriate symlinks to make it run at boot.

The script chooses which X server to run based on whether DISPLAY is set already; if so, it starts up Xvnc instead of XFree86. This is done to help with testing: when autologin-x is run by run-chroot.pl inside an xterm, we can connect to it with a VNC client to make sure all the usual X applications come up correctly. Of course, for X to work on the real CD-ROM, we need to know what video hardware the user has.

Hardware Detection

Hardware detection in Linux has improved a lot in the last ten years, helped by the improvements in hardware technologies. It's a lot easier to detect today's PCI and USB hardware reliably and safely than it was with the ISA devices we used to have.

Most Linux distributors have something that grovels through the PCI and USB devices in the system and loads appropriate modules. Knoppix uses Kudzu, originally written for Red Hat, but vanilla Debian uses the discover command. The two are pretty similar in coverage; as it's all open source, they can copy from each other's hardware databases. The Debian X server packages already use discover to provide defaults for X configuration questions, so we'll stick with it.

debconf

What do we do with the hardware we detect? Debian packages have human-editable configuration files, but they typically also come with post-installation scripts that create the initial versions of said files interactively. Where applicable, such as for X and network configuration, these scripts run the hardware detection tools.

The problem is we're installing the packages in a chroot on the host system, and detecting the host system's hardware is not going to help on the target. What we need to do is put the debconf database somewhere writable, so at boot time we can use debconf-communicate to unconfigure the package and run its .config script to make it think it's being configured for the first time. This is a more thorough approach than using dpkg-reconfigure, which sometimes asks questions such as, "Are you sure you want to reconfigure this package?" This can be confusing to the end user who hasn't even configured it once yet. See the debconf-communicate manual page and target/etc/init.d/configure-xserver in the tarball for details.

Persistent Storage: Hotplug

The CD-ROM is read-only, and a ramdisk goes away when the power is turned off. People want to save their files, though, or even have access to the files they've created already on existing hard disks or on removable devices, including USB keychains and Zip drives. Again, most of the hard work has been done for us; this time hotplug and autofs are our saviours.

Hotplug listens for new devices being added or removed. When it sees a new USB storage device, it loads any necessary modules and creates an emulated SCSI host. We still need to know what devices are available and mount them, and that's where autofs comes in.

autofs mounts and unmounts filesystems on demand. Using a program map, we can have a Perl script run whenever the user asks for /media/list; it creates a directory with links named after the attached devices. These links point to more autofs mount points to access the filesystems. In the tarball, look at target/etc/auto.master and target/usr/local/sbin/autofs-device-list.

The Kernel

We use basically the same kernel configuration as Knoppix (look at /usr/src/linux/.config in a running Knoppix system, or kernel-config in our tarball), but we remove support for a few obviously unused things, such as ZISOFS. The standard Debian make-kpkg tool patches, builds and installs the kernel. This is a Debian dependency on the host system (you need the cloop-src package), and as it's probably the only nontrivial such dependency, it might be worth moving into the chroot in a later version.

The Filesystem

Most of a UNIX filesystem is happy mounted read-only, but we do need to write files in some places. For example, the X server configuration file needs to be written at boot time according to the hardware in use, the debconf database must be updated and there are various log and lock files too.

We use the tmpfs filesystem to create a RAM-based filesystem. The system is arranged to use this ramdisk for root and expect the cloop image on /ro. Then for read-only directories, we create symlinks, for example, from /usr to /ro/usr.

We keep a list of read-only directories, and we check it twice. First, we create a tarball of the system that excludes all these directories, replacing them with appropriate symlinks. This tarball then is copied into the root filesystem of the running system. Second, when we're writing out the ISO9660 image to be cloop-compressed, this is the list of directories to include.

initrd

Before the system proper starts up, there are two important things we must do. First, we need to mount the cloop image, load whatever modules the CD-ROM needs, then find and mount the CD. Next, we install the cloop device and mount the inner filesystem on it. Second, we create a ramdisk for the root filesystem and copy the root_fs.tgz image from the CD into it.

We use the initrd (initial ramdisk) support to create a mini root filesystem that the kernel mounts and runs before the real init starts. This is a gzipped filesystem. When a kernel with initrd support is booted with the command line `initrd=filename`, it loads the contents of that filename and creates a ramdisk out of it. It then starts running the /linuxrc file in that ramdisk.

When linuxrc has finished, it uses the pivot_root call to change onto the real root directory, which was /ramdisk, and executes the real init.

The initrd and the kernel together need to be small enough to fit in 1.44MB of RAM with all the other files on the boot image. This is not a lot of space, as GNU libc alone is about 1,200K, we're going to have to be pretty creative.

dietlibc, BusyBox

Even if you've never wanted a Linux PDA or an in-car MP3 jukebox, you now have a reason to be grateful to embedded Linux hackers. We're going to use Busybox and dietlibc to get our quart into the proverbial pint pot. Busybox is a small shell that can be configured at build time to include many common utilities as built-ins, and dietlibc is an alternative C library optimized for small size. By happy coincidence there turns out to be a Busybox applet for everything we need on the initrd, and by statically linking with dietlibc we can get all this into about 100K. For comparison, the same Busybox options statically linked against glibc get a 500K executable.

Applets for Busybox are enabled using #defines in its Config.h file (in the tarball). Some of the disabled options may seem rather arbitrary, but when you already have a choice of `echo *` and `tar cvf /dev/null` to list the current directory, `ls` really is a luxury.

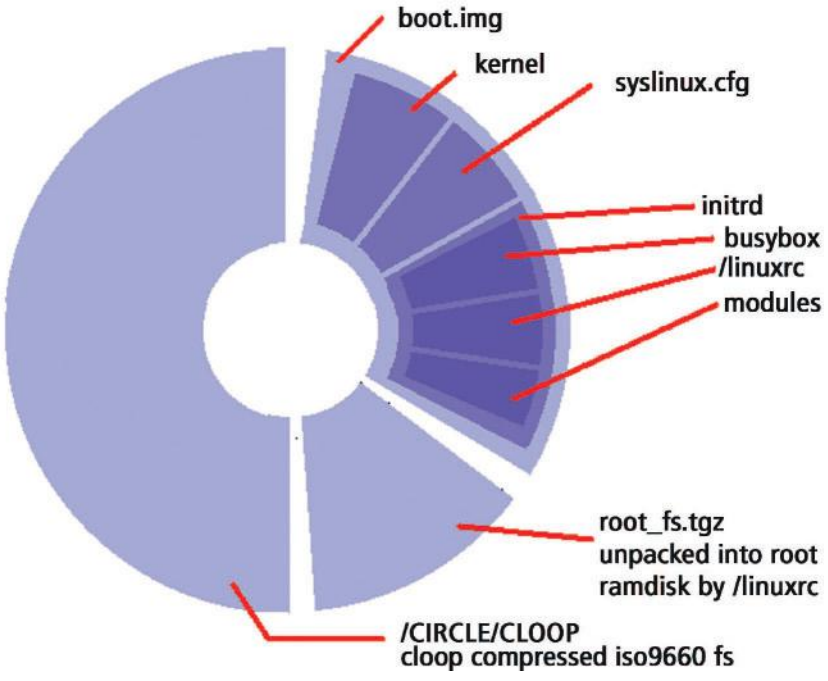


Figure 1. Wheels within wheels: the finished CD nests filesystem images inside filesystem images inside filesystem images.

We create the initrd using `genext2fs`, avoiding the need for a loopback mount. This generates an ext2 filesystem from a directory tree, which we gzip and copy into the boot floppy image (Figure 1).

Booting

The standard for booting from CD-ROM is known as El Torito and was originally produced by the Phoenix BIOS writers. El Torito allows the creation of one or more disk images on the CD-ROM. At boot time, the BIOS locates these and creates an emulated disk from which it then boots. Images may be of floppies (1.44MB or 2.88MB) or of hard disks. There's also a no-emulation mode, in which the BIOS loads sectors from the specified file and executes them without setting up an emulated disk.

There's a catch, of course: El Torito is implemented by BIOS writers. Linux users with laptops or other interesting hardware already know that BIOSes are not always the least-buggy code on the planet. It's been suggested that some manufacturers happily ignore the actual specification as long as whatever they concoct manages to load the current version of Windows. So, painful though the space restriction is, to ensure maximum portability, we follow Knoppix's lead and stick to a single 1.44MB floppy image.

boot.img

What do we put in this 1.44MB? We could boot a raw Linux kernel, or we could use a normal Linux bootloader such as LILO or Grub. H Peter Anvin's SYSLINUX tool beats both of these options for ease of use, though. SYSLINUX creates boot disks that use an MS-DOS filesystem, so we can create the floppy disk image using the userland mtools. The disk needs the kernel vmlinuz file, syslinux.cfg, any ancillary help files and the initrd image. When done, we run SYSLINUX on it.

All that remains now is to create our filesystems and burn them, much as we did earlier. The inner filesystem is in $(SCRATCH)/CLOOP. We create an outer filesystem containing this, boot.img and root_fs.tgz. We then write that to CD (a CD-RW or two would be useful) and reboot with it. And, with any luck, it works.

Finishing Up

As a longtime Linux user who hasn't done a normal install in years, it's impressive to see how much work has been done recently on hardware detection and autoconfiguration. As time goes by, I'm sure it'll get even better.

Where does this project go next? The automount support needs work; we might try something like Volumatic instead. Other than that, it depends on the product based on it. But all the scripts are free software, and I'm looking forward to feedback.

Resources for this article: www.linuxjournal.com/article/8060.

Daniel Barlow is an independent consultant in Oxford, UK, where he hacks Linux and Common Lisp compilers. In his spare time, he likes to play the electric guitar badly, which is fortunate as it's the only way he knows how to play it. Comments are welcome to dan@metacircles.com.

Building Impress and PowerPoint Slides with LaTeX and Perl

Forced to use proprietary file formats? Let open source ease the burden. BY PAUL BARRY

Let's begin with a story. Here's what happened: my second book, coauthored with Dr Michael Moorhouse, finally was finished. I had spent an extra six months on it, which meant it now was at least six months late. I had spent every spare minute typesetting, proofreading, writing, manually converting Michael's Microsoft Word files to LaTeX, reading and then re-reading. Then, I'd proofread it all again. When it was done and dusted, I was jaded. Soon after, I received the final proof of the cover. And there it was—printed right on the back cover—a promise to provide Microsoft PowerPoint slides on the Web site for use with the text. It was too late to change the cover, which meant I was committed to providing the slides one way or another. I had forgotten that we had decided to do this at the start of the project, more than 18 months prior.

The PowerPoint "Standard"

Eighteen months ago, PowerPoint was the de facto standard slide production technology within the academic community. Today, PDF is popular too. As with many in the Linux community, I already had made the move to OpenOffice.org, leaving PowerPoint behind. With 20 chapters in the book, I estimated it would take at least 20 days' effort to produce the slides manually. The thought of doing this work with PowerPoint was not something I relished. I could work within OpenOffice.org Impress, of course, and then export to PowerPoint when finished, but this idea didn't sit well with me, either. The basic problem was I knew all the content already was in the LaTeX files and having to reproduce it using a slide production application left me feeling even more drained than I already was. If only I could find a way to extract the content programmatically from my LaTeX files and populate PowerPoint slides with it—that would improve things considerably.

Working with Presentation File Formats

Searching Google resulted in frustration. Perhaps not surprisingly, details of the PowerPoint file format were hard to come by. I did find a file in Microsoft Windows Help format that described the XML standard for Microsoft Office documents, to which PowerPoint documents can be exported. Unfortunately, it was a large, complicated piece of writing. Having decided I wasn't going to get anywhere on Google, I surfed over to Comprehensive Perl Archive Network (CPAN). Perl, my programming language of choice, has been hooked up to all types of file formats and other computing forms. If anyone had played with Perl and PowerPoint, details of the work would be available on CPAN. Unfortunately, this search also drew a blank.

Then it occurred to me: if I could work with the open and widely published OpenOffice.org Impress document format, I then could export my Impress slides to PowerPoint as a last step. A quick perusal of the OpenOffice.org Web site uncovered the official XML description of the OpenOffice.org file formats. Weighing in at more than 600 pages, the standard is bigger than my book!

The XML document is well written, but it's pretty heavy going. I surfed back to CPAN to see if any other programmers had taken the time to work with OpenOffice.org formats and were gracious enough to upload their work to CPAN. This time I wasn't disappointed. Jean-Marie Gouarne of Genicorp recently had released the OpenOffice::OODoc module, a Perl interface to the OpenOffice.org formats. Given an existing document, OpenOffice::OODoc can manipulate the content, adding to, deleting from and updating the disk file as need be.

The Slide-Producing Strategy

I started with a simple filter, written in Perl, that takes a LaTeX file as input and produces the slide content as output in a customized textual form. By producing a text file, I ensured that any text editor could be used to edit the output from the filter, fine-tuning the textual content as necessary. Once happy with the textual content, another filter, also written in Perl, uses the textual content to create an Impress presentation. The Impress presentation then can be opened in Impress and exported to PowerPoint and/or PDF format.

Slide Design

I made a conscious effort to keep my presentations as simple as possible and decided to have only three slide types. The title_slide would contain the title of the chapter at the start of the presentation file. Within the presentation, the title_slide would do double duty as a placeholder for any graphic images associated with the chapter, with one title_slide created per graphic image. The bullet_slide would contain section titles as

its slide heading and subsection titles as bullet items. Finally, the sourcecode_slide would provide a mono-spaced, verbatim slide used for program listings.

I used Impress to create a three-slide presentation manually, which I called blank.sxi. Each of the created slides corresponded to each of the three slide types described in the last paragraph. I planned to clone this presentation every time I programmatically created a presentation for each of my chapters. By cloning, I'd ensure that all of the presentations conformed to a standardized look and feel.

The Filter for Extracting Textual Content

The getcontent script is the type of script that Perl programmers typically create, use and then throw away. (See the on-line Resources for downloading the files referred to in this article.) It loops on standard input, reading one line at a time, and attempts to pattern-match on content of interest. If a match occurs, appropriate output is produced. As an example of what getcontent does, here's the code for dealing with the chapter title from the LaTeX file:

```
if ( /\\chapter\{(.*)\}/ )
{
    print "CHAPTERTITLE: $1\n";
    next;
}
```

A simple regular expression attempts to match on the LaTeX chapter macro; if a match is found, the chapter title is extracted and output is generated. The call to next short-circuits the loop, allowing the next line to be read in from standard input when a match is found. In this way, the following LaTeX snippet:

```
\chapter{Working with Regular Expressions}
```

is transformed into this textual content:

```
CHAPTERTITLE: Working with Regular Expressions
```

That is, the LaTeX markup is removed and replaced with a much simpler markup. The section and subsection LaTeX macros were treated in a similar way. Here's the code:

```
if ( /\\section\{(.*)\}/ )
{
    print "BULLETTITLE: $1\n";
    next;
}

if ( /\\subsection\{(.*)\}/ )
{
    print "BULLETCONTENT: $1\n";
    next;
}
```

Working with source code listings is only slightly more complex, due to the requirement to spot when a chunk of verbatim text has been entered and exited. Here's the code that handles entry into a LaTeX verbatim block:

```
if ( /\\begin\{verbatim\}/ )
{
    print "STARTCODE\n";
    $in_verbatim = TRUE;
    next;
}
```

And, here's the code used to handle the exit from a verbatim block:

```
if ( $in_verbatim )
{
    if ( /\\end\{verbatim\}/ )
    {
        print "STOPCODE\n";
        $in_verbatim = FALSE;
    }
    else
    {
        print;
    }
    next;
}
```

A simple boolean, the $in_verbatim scalar, helps to determine whether the script currently is working within a verbatim block. Similar code extracts the maxims that appear throughout the book's chapters, and a few if blocks handle the graphics, their captions and other content of interest. For example, consider the following chunk of LaTeX markup:

```
\chapter{The Basics}

\textit{Getting started with Perl.}

\section{Let's Get Started!}

There is no substitute for practical experience when
first
learning how to program. So, here is the first Perl pro-
gram
\index{welcome@\texttt{welcome}, and the first program,
called
\texttt{welcome}:

\begin{verbatim}
    print "Welcome to the World of Perl!\n";
\end{verbatim}

\noindent When executed by \texttt{perl}
\footnote{We will learn how to do this is in
just a moment.}, this small program displays
the following, perhaps rather not unexpected,
message on screen:

\begin{verbatim}
    Welcome to the World of Perl!
\end{verbatim}
```

The getcontent script transforms the above LaTeX into this textual content:

```
CHAPTERTITLE: The Basics
CHAPTERCONTENT: Getting started with Perl.
BULLETTITLE: Let's Get Started!
STARTCODE
    print "Welcome to the World of Perl!\n";
STOPCODE
STARTCODE
    Welcome to the World of Perl!
STOPCODE
```

Notice how all of the LaTeX markup is gone, replaced by a simpler markup language that will be used to produce slides programmatically. Assuming the LaTeX chunk was in a file called chapter3.tex, the getcontent script is executed as follows, piping the result of the transformations into an appropriately named file:

```
perl getcontent chapter3.tex > chapter3.input
```

The chapter3.input file now contains the textual content, and it can be fine-tuned with any text editor prior to producing the slides.

The Impress Presentation Creation Filter

Producing the slides within an Impress document was complicated by a number of factors. For starters, the OpenOffice::OODoc module cannot be used to create a new OpenOffice.org file; it can manipulate existing files only. Additionally, the module was created with a view to working primarily with OpenOffice.org Writer files—word processor documents—not Impress presentations. By way of example, here's a short program, called appendpara, that adds some text to an already existing Writer document:

```
#! /usr/bin/perl -w

use strict;

use OpenOffice::OODoc;

my $document = ooDocument( file => 'blank.sxw' );

$document->appendParagraph
(
    text     => 'Some new text',
    style    => 'Text body'
);

$document->save;
```

This small program uses the OpenOffice::OODoc module and creates a document object from the existing Writer file. The program then invokes the appendParagraph method to add some text before invoking the save method to commit the changed document to disk.

In addition to the appendParagraph method, the OpenOffice::OODoc module provides the insertElement method, which allows a new page of a specified type to be added to a document. The page can be a clone of an existing page or it can be actual, raw XML.

After reading as far as page 6 of the 600+ page OpenOffice.org XML file format document, I discovered that Impress used the //draw:page XML type to represent a slide within a presentation. Unfortunately, the OpenOffice::OODoc module could not work directly with objects of this type, so I had to come up with some other mechanism to manipulate the data. Specifically, I wanted to take the blank template slides contained in the blank.sxi document and clone each slide as I needed it, populating the slide's content with the textual content produced by the getcontent script. To do so, I needed to learn more about the Impress XML format.

I had two choices: continue to read the 600+ page standard document or take a look at an actual file to see if I could learn enough to get the job done. I chose the latter. Recalling from a previous *Linux Journal* article that OpenOffice.org compacts its multipart file using the popular ZIP algorithm, I created a temporary directory and unzipped the blank.sxi file:

```
mkdir unzipped
cd unzipped
unzip ../blank.sxi
```

This produced a bunch of files and directories:

```
content.xml
META-INF
meta.xml
mimetype
settings.xml
styles.xml
```

Of most interest is the content.xml file, which contains the actual content that makes up the document. Viewing this on-screen or within an editor produced a mass of hard-to-decipher XML. In order to keep the parts as small as possible, no attention had been paid to formatting the XML, in any of the parts of the zipped container, in any meaningful way. Typically, the XML is dumped/stored as a non-indented, non-whitespace text stream. To try to make sense of it, I needed to be able to print the XML in a legible manner. In what I can describe only as a moment of temporary inspiration, I dropped into a command-line and typed `xml` followed by two tabs. A listing of pre-installed tools that start with the letters xml appeared on screen:

```
xml2-config     xml-config      xmllint
xmlto           xml2man         xml-i18n-toolize
xmlproc_parse   xmlwf           xml2pot
xmlif           xmlproc_val     xmlcatalog
xmlizer         xmltex
```

The xmllint tool immediately caught my eye. Reading its man page uncovered the --format option, which—yes, you guessed it—pretty-prints XML provided to the tool. Therefore, typing `xmllint --format content.xml` resulted in output I could pipe to less and actually read without losing my sanity. Here's an abridged snippet of the pretty-printed content.xml showing the XML for the title_slide from the blank.sxi Impress document:

```
<draw:page draw:name="page1" draw:style- ...
  <draw:text-box presentation:style-name= ...
    <text:p text:style-name="P1">
      <text:span text:style-name="T1">
        ChapterTitleSlide
      </text:span>
    </text:p>
  </draw:text-box>
  <draw:text-box presentation:style-name= ...
    <text:p text:style-name="P3">
      <text:span text:style-name="T2">
        ChapterTitleSlideText
      </text:span>
    </text:p>
  </draw:text-box>
  <presentation:notes>
    <draw:page-thumbnail draw:style-name= ...
    <draw:text-box presentation:style-name ...
  </presentation:notes>
</draw:page>
```

Notice the ChapterTitleSlide and ChapterTitleSlideText content, which I had typed into blank.sxi when creating it with Impress. If I could use the insertElement method to add raw XML based on this extract, with the empty content replaced with my textual content, I'd be home free.

By way of example, consider what happens once the title of the presentation and its subtitle are processed by produce_slides. The insertElement method is invoked as follows, creating a new slide:

```
$presentation->insertElement( '//draw:page',
  $last_slide++,
    title_slide( $title_title, $title_content ),
      position => 'after' );
```

The title_slide subroutine returns raw XML, which is inserted into the document.

Given an input file conforming to the textual content produced by getcontent, the produce_slides script clones the blank.sxi Impress file and populates any number of slides, programmatically producing a presentation. The script is not unlike getcontent in structure, its only warts being the verbatim inclusion of the required XML for each of the three slide types contained within blank.sxi. To create a presentation, invoke produce_slides as follows:

```
perl produce_slides 3 chapter3.input
```

This results in a new Impress document called chapter3.sxi appearing on disk.

With the Impress files created, I needed to replace my graphic image placeholders with the actual image. The getcontent script extracted the image filename, however, not the actual image. Importing the images into Impress should have been straightforward, except that the originals I had were of pretty poor quality compared to those that made it into the book. The final images had been improved greatly during the publisher's final typesetting phase. And, of course, I didn't have the final image files.

Then I remembered that the publisher had sent final proof PDFs with all the high-quality graphic images in place. I used xpdf to view the proofs at 200% and then fired up The GIMP to screen-capture the xpdf display window. I then cut out the graphic image and saved it as a JPEG. It took a little while, but when finished I had a beautiful set of book-quality images to import into my Impress presentations. With this task complete, I exported the Impress document to PowerPoint format and the job was done. My initial estimate of 20 days of effort was reduced to about 20 hours of real work.

And now, of course, if I need to produce some slides quickly, I can create my textual content manually in vi, run it through the produce_slides script and I'm done.

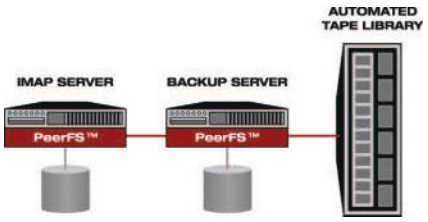
Final Words

What started off as a seemingly impossible task—programmatically producing PowerPoint presentations—turned out to be quite possible, thanks to open source. All the tools I needed shipped out of the box with my stock Red Hat 9 distribution: vi, unzip, Perl, xmllint, xpdf, The GIMP and the OpenOffice.org suite.

Resources for this article: www.linuxjournal.com/article/8055.

Paul Barry (paul.barry@itcarlow.ie) lectures at the Institute of Technology, Carlow, in Ireland. Information on the courses he teaches, in addition to the books and articles he has written, can be found on his Web site, glasnost.itcarlow.ie/~barryp.

PeerFS Version 3.0



Radiant Data Corporation has released PeerFS version 3.0, peer-to-peer continuous data availability technology for Linux-based enterprise applications. PeerFS enables simultaneous transactions on multiple servers in multiple locations with separate but identical data stores. New features of PeerFS version 3.0 include support for more distributions, including Trustix and Debian, support for the 2.6 kernel and support for SuSE Standard Server 9.0 and SuSE Enterprise Server 9.0; a lost node policy that detects when one or more nodes in the configuration group is no longer reachable; and support for consistency groups with more than two nodes. In addition, PeerFS diskless clients receive new functionality with the addition of load balancing and host affinity options to the mount command.

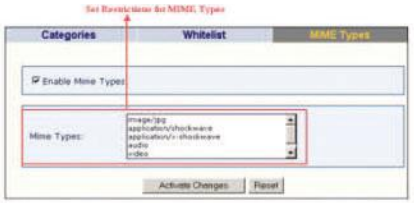
CONTACT Radiant Data Corporation, 6309 Monarch Park Place, Niwot, Colorado 80503, 866-652-0870, www.radiantdata.com.

1-Box for Linux 1.0

1-Box for Linux 1.0 is standalone software that can be added on to Linux distributions in order to turn a single PC into a network of up to ten workstations. With the addition of extra dual-head video cards to the main PC, each workstation needs only a standard monitor, a USB keyboard and a mouse. Users simultaneously can browse the Internet, send e-mail and independently run any installed software they desire. 1-Box offers support for Novell, Mandrake, Fedora Core and Red Hat distributions, with support coming soon for Sun Java Desktop.

CONTACT Userful, 2nd Floor, 928 6th Avenue SW, Calgary, AB T2P 0V5, Canada, 866-873-7385, www.userful.com.

WebScan for Linux



WebScan for Linux combines antivirus and content security features in order to protect the network on the gateway or proxy server level. WebScan was designed to allow organizations to control the type of Web traffic content that can flow through the gateway and to protect the network from viruses that gain access through proxy servers. WebScan can scan Web pages for content policy violations, viruses, worms, Trojans and other malware. It also allows blacklisting of MIME file types, such as audio and video, so that Internet bandwidth is used effectively. Also, HTTP file uploads can be blocked to prevent theft or leakage of sensitive data. Unauthorized access to certain Web sites also can be prevented based on ratings by organizations such as RASCi, Safe Surf and ICRA. For administration, WebScan offers an extensive reporting system for policy violations and a Web-based GUI front end for easy configuration and administration.

CONTACT MicroWorld Technologies, Inc., 33045 Hamilton Court East, Suite 105, Farmington Hills, Michigan 48334, 877-398-4787, www.mwti.net.

PostgreSQL 8.0

The PostgreSQL Global Development group has released version 8.0 of PostgreSQL, an object-relational database management system. Key new features for version 8.0 include savepoints, an SQL-standard feature that allows specific parts of a database transaction to be rolled back without aborting the entire operation. Also new for PostgreSQL 8.0 is point-in-time recovery, a feature that allows full data restoration from the automatic and continuously archived transaction logs, which is an alternative to hourly or daily backups. Version 8.0 also offers tablespaces, which allow the placement of large tables and indexes on their own individual disks or arrays, improving query performance. Finally, PostgreSQL offers improved disk and memory usage through the use of the Adaptive Replacement Cache algorithm, the new background writer and the new vacuum delay feature.

CONTACT The PostgreSQL Project, 415-752-2500, www.postgresql.org.

IBM OpenPower 710

IBM announced the release of the eServer OpenPower 710, a POWER5 processor-based server running Linux. The OpenPower 710 is a one- or two-way rack-mount system that uses IBM's 64-bit Power architecture and offers optional mainframe-inspired virtualization and micro-partitioning capabilities unique to POWER5 systems. The OpenPower 710 is available with 1.65GHz POWER5 microprocessors and a maximum memory of 32GB. It supports Novell SUSE LINUX Enterprise Server 9 and Red Hat Enterprise Linux AS 3. The 710 also comes with 1GB of memory, a 73GB 10KRPM disk drive, DVD-ROM and a three-year, next-business-day warranty. Four standard hot-swappable Ultra320 SCSI drive bays are available for more than 570GB of internal storage. The system has three PCI-X slots, dual 10/100/1000 Mbps Ethernet ports, hot-plug power supplies with optional redundancy and redundant hot-plug cooling.

CONTACT IBM Corporation, 1133 Westchester Avenue, White Plains, New York 10604, www-1.ibm.com/servers/eserver/openpower.

Please send information about releases of Linux-related products to Heather Mead at newproducts@ssc.com or New Products c/o *Linux Journal*, PO Box 55549, Seattle, WA 98155-0549. Submissions are edited for length and content.

Register Today At WWW.FOSE.COM • FOSE is FREE* for Government and Military personnel!

FOSE

Where Government Meets Technology

April 5-7, 2005 • Washington DC Convention Center

FOSE provides more resources and solutions through expanded educational programs, 500+ exhibits, informative keynotes speeches, and networking opportunities.

- ***Visit the NEW Linux Pavilion, providing government agencies with a look at forward-thinking companies who provide Linux solutions and product demonstrations, specifically for the government sector***
- ***Explore new job opportunities at The Washington Post Defense Tech Career Fair at FOSE***
- ***Find 8(a) and Small Business that will fulfill your service needs from the many new businesses like: Access Innovations, Capital Asset Recovery, and Komplete Systems Integrators on display in the Small Business Pavilion***

E-Town and Homeland Security Center Presented By

accenture
High performance. Delivered.

Media Sponsors:

GCN Government Computer News

WASHINGTON TECHNOLOGY

FOSE Presented By:

PostNewsweek Tech Media

** Registration is FREE for government & military professionals and $50 for non-government, and only open to trade professionals. No one under 18 is admitted, regardless of affiliation or circumstances.*

Tweaking inodes and Block Sizes

I want to ask a couple of questions. 1) I was wondering if there was a serious performance impact to formatting a Linux partition with the following commands:

```
mkfs.ext2 -i 1024 -b 1024 /dev/hda1
mkfs.ext3 -i -1024 -b 1024 /dev/hda2
```

I know that using the second command would enable the Journal filesystem, but would having so many inodes slow down the system? I'm trying to use this on a firewall system with Squid, INN and qmail services.

2) I have a matching pair of 486DX 66MHz systems and a 486SLC2 50MHz system, each with 32MB of RAM. Is there any way I could use Linux Rat Hat 9 on them? Or should I install Red Hat 6.2 and use up2date on them?

--
Lee Spivey, tuskyhe@yahoo.com

1) The effect of the size and number of inodes on disk access speed depends on the types of files they are used to reference. The commands given above indeed would yield greater utilization of the hard drive's capacity, and this seems like a good thing. This is especially true on larger hard drives, which multiply the effect of this value.

In practice, however, Web pages and messages have grown beyond 1KB files. Limiting a filesystem's block size to this value forces Linux to traverse a much larger tree of inodes to find the relevant entries and then remember which they are. The more inodes there are in one file, the longer this takes. Given the cost per megabyte of hard drives today, and the likelihood that the savings would amount to less than 100MB of space, 4–8KB might be a more reasonable value.

--
Chad Robinson, chad@lucubration.com

1) As Chad pointed out, the block size you choose will affect the performance. If the files you access the most often are over 1KB in size, you will have to access multiple inodes to retrieve these files and, thus, incur a performance hit. It's not so much a question of having a lot of inodes, but rather one of how many inodes will need to be accessed in order to retrieve the most commonly used files. That is, the issue is the average inode-to-file-size ratio—the inverse of the -i parameter in your mkfs command. Take this into consideration when laying out your filesystem and decide whether you want to optimize for speed or for total storage capacity. And, take into account what you predict to be the average size of what will be the most commonly accessed files. Also, make sure you don't limit yourself to too few inodes. It's likely that you will end up with significantly more files in the long run than you originally thought—depending on what you plan to do with the machine, of course—so make sure to not be too stingy. As for the performance issues between ext2 and ext3, an additional amount of overhead is associated with a journaling filesystem, but the performance hit generally is thought to be minimal, especially when weighed against the benefit of having a journal.

--
Timothy Hamlin, thamlin@nmt.edu

2) Neither Red Hat 9 nor Red Hat 6.2 is still supported, which means no more security updates. The successor, Fedora, requires a Pentium or better. You'll need to install a distribution such as Gentoo or Debian that has both pre-Pentium CPU support and current security fixes.

No matter what you install, this class of machine will be too slow for a modern desktop. You can use them for Web servers, print servers, firewalls or machines to learn on, though.

--
Don Marti, dmarti@ssc.com

Old Red Hat

I am having a problem with a Red Hat 7.2 installation on a 133MHz PC that I'm using as a Smoothwall proxy. I successfully installed the software, but when the computer rebooted and I tried to log in, I got a message similar to `error in service mode`. It's hard to tell because it flashes on the screen very quickly and then brings me back to a login screen. I checked the filesystem and made sure that bash was installed and that the environment path was set correctly. There still is something wrong though, because it's not logging me in. Can you suggest what the problem might be or, even better, point me toward a solution to this issue? I really would appreciate it.

--
Jeff, jloyd1@comcast.net

When the system is booted up and is showing the login screen, press and hold the Ctrl-Alt keys and press the F1 function key. This gives you the command line. You should be able to log in there as the root user with the root password. You can navigate to console 1 through 6 by using the Alt-F1 to Alt-F6 key combinations; F7 is graphical display. As you navigate from console 1 to 6, you may see more details about the error message and/or the events leading to it. Once you log in, look at /var/log/messages and other log files in the /var/log directory. This should get you started.

--
Usman S. Ansari, usmannsari@yahoo.com

Are you running with a graphical login? If so, try disabling it by editing /etc/inittab and changing to runlevel 3 instead of 5. Change the line:

```
x:5:respawn:/etc/X11/prefdm -nodaemon
```

to:

```
x:3:respawn:/etc/X11/prefdm -nodaemon
```

or do it temporarily through your bootloader. If you aren't running xdm, try examining your log files and searching for errors. Specifically, look at /var/log/messages and /var/log/secure, and if using X, look in the X logs as well.

--
Timothy Hamlin, thamlin@nmt.edu

Which Distribution?

This may be a silly question, but I'm considering putting Linux on my 80GB HD as a second OS. I'm looking to use it mainly for media, word processing, movies and music, as I've heard Linux is resource efficient. I'll be keeping Windows on mainly for gaming. I also have an Athlon 64 3500+ and want to make use of it with a 64-bit build that

works well. Can you direct me to a distro that would allow me to use my 64-bit processor to its best ability and that also would allow for easy media playback, Net surfing and so on? I looked at MandrakeLinux, but I've been hearing a lot of bad things about its AMD64 build. Thanks for your time, and I look forward to hearing your response.

--

Derek Allen, sock_ferret@hotmail.com

If I may shamelessly plug Gentoo (www.gentoo.org), this distribution allows you to get the most out of almost any hardware platform, because you have the option of natively compiling packages for your platform as you install them. This feature also commonly is listed as Gentoo's downside, because this process can be time consuming. However, the Gentoo team has worked hard to provide binary builds for a variety of platforms, including 64-bit, so this is less of an issue today.

Gentoo's installation process can be daunting, and although the developers are working on a formal installer, you may or may not like what you see when you start to load it. If you need an alternative, Red Hat and Novell/SuSE are good places to start. Both provide native builds and clear, intuitive installers. For a free option, you can't go wrong with Debian, whose developers call their AMD64 port "the most complete port after i386"—clearly an in-demand platform. All of the distributions mentioned here provide package managers that allow you to keep your system up to date and easily install new applications, such as the media players and, more important, the codecs you are after.

--

Chad Robinson, chad@lucubration.com

Finding the Home Page

I am running Red Hat 9.0, kernel 2.4.20-8, and I am using the supplied Apache server. When I log on to the server, I see a Test Page. I have my home page files in /var/local/www/html, as instructed. I am told to swap the test page for my home page, which is what I want to do. Have you any idea what file I should edit to make this happen? I have printed out the 15 pages of the httpd.conf file and scanned them for more than a few days, to no avail.

--

George Robertson, grobertson29@earthlink.net

I believe in Red Hat 9's default Apache installation, the test page is located in /var/www/html/index.html. So if you want to replace it, back up that file and replace it with yours.

--

Timothy Hamlin, thamlin@nmt.edu

Look for the DocumentRoot line in your Apache configuration file. That's the directory where your home page lives. Now look for the DirectoryIndex line. That's a list of possible names for the file. Before you put too much work into the system, though, you'd be better off to upgrade to a distribution that has current security updates. Red Hat 9 security fixes ended on April 30, 2004.

Is this Red Hat Museum Week or something?

--

Don Marti, dmarti@ssc.com

Remote Administration

I have been administering Windows servers through a VPN connection for a long time. Is there a similar way to administer Linux systems? I realize I can VPN to a Linux system,

but I mean is there a preferred method to access Linux systems remotely and do administration work? Could you recommend any books on the subject?

--

Ric Jones, rictjones@wideopenwest.com

The classic tool for administering Linux systems remotely is OpenSSH (www.openssh.com). It comes pre-installed on all the common distributions and gives you an encrypted way to run commands and transfer files without setting up a VPN. If you do want a VPN, Mick Bauer has an overview at www.linuxjournal.com/article/7881.

--

Don Marti, dmarti@ssc.com

Intranet DNS

I am trying to configure a bind server for my intranet using a residential cable modem router as the DHCP server. I am interested in having an intranet name to private IP address resolution and have any Internet DNS request forwarded to my ISP's DNS servers. I have been successful with getting the server to respond to an address record request (`ls -t`), but it won't return individual hostname IP addresses.

I have the root zone configured to point back to the bind server on the same PC. I also set up the domain zone ort.cloud containing the bind server host PC, router IP and hostnames of the individual network PC's IP to name mapping and canonical name to IP address mapping. Another zone takes care of the name to IP address and canonical name to IP address mapping. I'm not sure whether this redundancy is necessary or not, but it's kind of working for the time being.

--

Jeff, jloyd1@comcast.net

Probably the best source for information on setting up a DNS is the DNS-HOWTO, www.tldp.org/HOWTO/DNS-HOWTO.html. The author of that HOWTO, Nicolai Langfeldt, also has written a book entitled DNS and Bind *that claims to offer more details and examples than the HOWTO. I have a setup similar to the one you are looking to achieve: an internal DNS that serves the local private domain requests and connects to an outside server for external translations. If I recall correctly—it's been a while since I set it up—I found numerous simple examples and configs for accomplishing what I needed by Googling for "caching only nameserver".*

--

Timothy Hamlin, thamlin@nmt.edu

Nonstandard Driver Breaks on New Kernel

For some time I hesitated to forward my problem to you, but I have no idea how to solve it. My distribution is Slackware 10.0, my kernel 2.6.9, the compiler 3.3.4, and I am booting from CD with isolinux. The problem is the modem chip 536EP from Intel is not supported under Linux. The Intel-provided source code, Intel-536ep-4.69-5.4.src.rpm, is okay and my modem works. When I use a new kernel, I have to compile it separately. During the booting process I always get `Intel536: module license 'Proprietary' taints kernel`, but the modem works. I use KPPP under KDE 3.2. When kernel 2.6.10 came, I patched my kernel, compiled it with the same .config file and compiled the 536ep code again, but the modem doesn't work. There's no initialization, no waiting for the OK after ATZ and no dial tone. Of course, the old kernel 2.6.9 still is available and works with my modem. I would appreciate any help, comments or further assistance from you regarding this issue.

--

Werner Gerstmann, WGerstmann@web.de

You are relying on an out-of-the-main-kernel-tree driver to work properly on future kernel releases. That is almost guaranteed to not work over time, as kernel apis change and morph due to bug-fixes, security issues and feature changes. Please see www.kroah.com/log/linux/stable_api_nonsense.html for details about why the Linux kernel does not have a stable internal kernel api. I recommend contacting the author of the driver and asking him for help, as he is the one that knows the code the best.

--

Greg Kroah-Hartman, greg@kroah.com

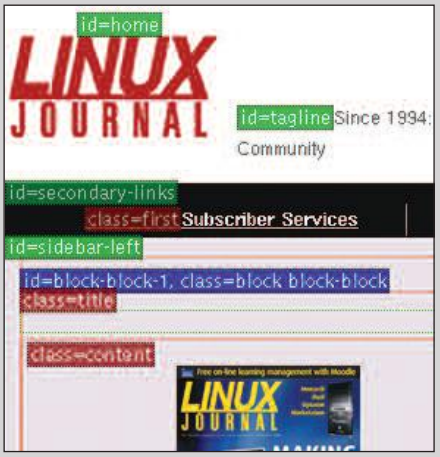
Many on-line help resources are available on the *Linux Journal* Web pages. Sunsite mirror sites, FAQs and HOWTOs can all be found at **www.linuxjournal.com**.

Answers published in Best of Technical Support are provided by a team of Linux experts. If you would like to submit a question for consideration for use in this column, please fill out the web form at **www.linuxjournal.com/lj-issues/techsup.html** or send e-mail with the subject line "BTS" to bts@ssc.com.

Please be sure to include your distribution, kernel version, any details that seem relevant and a full description of the problem.

Web Developer Extension

www.chrispederick.com/work/firefox/webdeveloper



Mozilla Firefox supports easy-to-install extensions, and one of the most useful is Chris Pederick's Web Developer Extension, which brings together many Webmasters' ideas for viewing and testing a site's look and functionality. For example, you can display all classes and IDs, as shown here, to make it easy to work on your stylesheet without viewing source on the HTML. You also can clear out cookies and HTTP authentication for your site to start a new session easily or run the W3C validator on the current page. You even can sanity-check tables with a temporary border without changing the HTML or the CSS.

—DON MARTI

Be a part of the largest gathering of MySQL users on the planet.

Discover an energizing array of new products, technologies, skills, and experts—all under one roof

Become a part of the MySQL community of developers and innovators who are defining the future of MySQL

Gain real-world knowledge, improve productivity, and learn best practices

Drill Down into the new MySQL 5.0 features with the MySQL development team

See how leading organizations have successfully implemented MySQL to grow their business and cut costs

Build relationships with fellow MySQL users, architects, innovators, industry leaders, and open source experts—including MySQL founders and key engineers

Register today!

Call 800-998-9938

http://mysqluc.com

Third Annual

MySQL®

Users Conference 2005

April 18-21, 2005

The Westin Santa Clara, Santa Clara, CA

"MySQL is at the heart of revolutionary changes in the computer industry.

–Tim O'Reilly, O'Reilly Media, Inc.

MySQL Everywhere

Web, Enterprise,
Data Warehouse, Embedded

Co-Presented By O'Reilly Media, Inc. and MySQL AB

LifeKeeper

REVIEWED BY SEAN TIERNEY

PRODUCT INFORMATION

Manufacturer:
Steeleye Technology, Inc.

URL:
www.steeleye.com/products/linux

Price:
Core Application $2,000 US per server; Application Recovery Kits $500 US per server

THE GOOD

- Easy implementation.
- Documentation.
- Supported applications.

THE BAD

- Data-storage options.
- Communication.

LifeKeeper for Linux is a high-availability clustering software package from Steeleye Technology, Inc. Steeleye acquired LifeKeeper when NCR spun off the technology, originally developed by AT&T Bell Labs. Steeleye ported LifeKeeper to Linux as well as to other operating systems. Version 4.4.3 supports failover for communications resources, databases, filesystems and mail, print and Web servers.

Steeleye refers to the type of high availability provided by LifeKeeper as fault resilience, the ability to recover from a failure automatically. This is differentiated from the idea of fault tolerance, where the system continues to operate after a failure occurs.

LifeKeeper is supported on various Linux distributions, including Red Hat, SuSE, UnitedLinux and Miracle Linux. The minimum system requirements for LifeKeeper are a supported Linux distribution running on an Intel-based server, 64MB of RAM and approximately 10MB of local disk space. Data protection is achieved by using either shared storage with SCSI or Fibre Channel or non-shared storage using LifeKeeper Data Replication.

The LifeKeeper software contains of a set of core applications and is extended by application-specific recovery kits (ARKs). The installation support and core applications package installed the software base. This included binaries and configuration files for the graphical and command-line interfaces, recovery support for the operating system, filesystems, SCSI subsystem, processor, memory, IP address and raw I/O. It also included an on-line help system and man pages. Application recovery kits are available for Apache Web server, data replication, IBM DB2, Informix, Logical Volume Manager, MySQL, NAS, NFS, Oracle, PostgreSQL, print services, SAMBA, SAP and Sendmail.

The software is licensed per server and per recovery kit. A cluster of two servers requires two licenses for the core application and two additional licenses for each of the application recovery kits. For instance, to protect a pair of LAMP Web application servers, licenses are required for the core application, plus Apache and MySQL application recovery kits. Although licensing costs can mount up quickly, it does allow you to pay for only what you need.

I began my review of LifeKeeper for Linux by reading the product documentation, taking the on-line tutorial and attending a Web-based seminar. This is a well-documented product. The CD-ROMs I received from Steeleye contained a planning and installation manual, a configuration guide and manuals for each of the application recovery kits. The documentation was available on the Web as well as in PDF format. The on-line tutorial was fairly basic and covered the same information as the manuals.

The seminar consisted of a marketing presentation and a live demonstration of LifeKeeper. I felt that the presentation and demonstration would be useful to anyone starting to look into the product. If you're looking to introduce LifeKeeper into your business, it may be useful to have managers and coworkers attend the seminar. The live question-and-answer session was the best part. I encourage anyone interested in the product to review the tutorial and on-line documentation and compile a list of questions to submit during the seminar.

Some flexibility exists in the cluster configuration, so it is a good idea to spend some time considering what hardware, applications and services you want to protect. As a minimum, you should consider server hardware, storage options, communications path, failover model, protected applications and services. Steeleye is positioning LifeKeeper as a commodity product. As such, it should support most reasonable server configurations. Nevertheless, they have certified some hardware and provide guidelines for verifying LifeKeeper with uncertified hardware. Certified hardware vendors include Dell, HP and IBM. In fact, you can include the LifeKeeper software when purchasing systems from them.

Multiple storage options are available to choose from. Shared storage consists of a SCSI or Fibre Channel array that is connected to both systems in the cluster. Data is located on the shared array. LifeKeeper's locking mechanism prevents the standby system from accessing the partition while the active system is in service. The data-replication option enables data stored on the local disks of one system to be mirrored to another system. The network-attached storage option facilitates the use of volumes mounted from an NFS server or NAS device. For instances in which the

data is static, such as Web servers, there is an option to not share or replicate the data store.

A central concept of LifeKeeper, as with most high-availability solutions, is the system heartbeat. One server sends a signal to the other to determine system and application health. Heartbeat communication path options include serial port and LAN. It is a good idea to use multiple paths, such as serial and LAN or multiple LAN connections. The failover models include active/active, active/standby and N+1. In active/active configuration, each server in the cluster is providing its own set of applications and services. If one fails, the other takes over. Users may experience some degradation of services, because the remaining system is serving both sets of applications and services, although it does allow for maximum resource utilization.

Active/standby provides the best continuity of service after a failure. However, it requires a redundant system and the associated cost. In N+1 configuration, one standby system provides failover protection for multiple active systems. This configuration provides reasonable utilization of resources while minimizing cost. If multiple failures should occur, users still may experience some increase in response time. Alternately, other active servers could be configured to take over. As previously mentioned, LifeKeeper offers failover protection for a variety of system components, services and applications. More information and documentation is available for each of the application recovery kits on the Steeleye Web site.

The first test scenario was a pair of servers running Linux, Apache, MySQL and PHP, serving up several Web applications. The hardware configuration I used was a cluster of two servers with dual network cards. I connected one NIC (eth0) on each server to the LAN; the second NICs (eth1) were connected to each other using a crossover cable. I connected the serial ports (ttyS0) with a null modem cable. I installed and tested the operating system, applications and supporting software before installing LifeKeeper. This is the recommended procedure, although the software could be installed after LifeKeeper.

During my first pass at installing LifeKeeper, I was running a custom kernel. Consequently, the Data Replication and NFS Recovery Kits were not installed. However, the installation guide provides instructions for patching your kernel and modules as needed. Later, I rebuilt the system and used a default kernel. No glitches occurred while running the installation support setup, installing the core applications and recovery kits. I used the LifeKeeper GUI to set up the communication paths for the heartbeat and to protect the Web application. Command-line procedures are available as well. The manual has step-by-step instructions for each phase of the setup and configuration, but the process is fairly intuitive. I tried several other configurations, including shared storage and legacy systems.

Once the software was installed and configured and I had tested all of the protected applications to ensure they were working properly, I ran several failover tests. I used the GUI to failover manually from one server to another and back again. This is the procedure that would be used to take a protected system out of service for maintenance. The other failures I induced included killing and shutting down protected services, shutting down and removing cables from the network interfaces and heartbeat communication paths and shutting down and pulling the power cord from a protected system. Manually taking a system out of service produced the quickest change over. Failover due to one of the faults I induced, however, was not as prompt. Failover from the active to standby system was quick but not immediate. A system administrator who might be watching the systems closely or a user who happened to be accessing the application when a fault occurred would notice a momentary pause in service. Depending on the type of application or service provided, this may not be a problem. Overall, I found the performance for failover and restoration of services to be adequate and consistent across all of my tests.

Having experimented with high-availability, open-source solutions and having used other commercial packages, I found LifeKeeper for Linux version 4.4.3 to be a good product. It is well documented and the software is comparatively easy to install and configure. Application recovery kits are available for most situations. Additionally, a generic recovery kit and a software development kit are available for those few cases not covered. The technical support is knowledgeable and helpful, and the cost is reasonable. Anyone in the market for a high-availability solution definitely should consider this product.

Sean Tierney is a graduate student at the University of Washington and a systems programmer working with UNIX and LANs. When not obsessed with a new computer project, he enjoys spending time with his wife, son and dogs on their dandelion ranch south of Seattle. He welcomes your comments sent to reviews@prnkstr.com.

Performers Go Web

With UpStage, the next theater is only a mouse click away. BY PATRICIA JUNG

Writers, musicians, painters, filmmakers and artists of every kind are using the Web as a platform. Only one traditional art form does not have a strong presence in cyberspace yet—theater. But, as soon as one is willing to adapt to the medium, a new art form evolves, cyberformance.

The term cyberformance was coined by New Zealand performance artist Helen Varley Jamieson to describe "performance that uses the Internet to bring remote performers together, in real time, in a live theatrical event". She has been working for several years with the cyberformance troupe Avatar Body Collision, using free Internet chat applications to create performances in cyberspace. To provide her, her coperformers and their audience with a Web-based stage, she initiated an open-source project called UpStage, written by Douglas Bagnall (see the on-line Resources). The first release, launched in January 2004, was funded by the New Zealand Ministry of Research, Science and Technology and Creative New Zealand, and funds now are being sought to continue its development.

Of course, the software isn't restricted to on-line performances. UpStage also makes an interesting tool for on-line teaching, as well as product and other types of presentations. It even serves as a collaboration tool for virtual workgroups. UpStage's strength is its user-friendly and highly accessible interface: players and audience alike need to have nothing more than a standard browser and Internet connection to participate. Newbies can learn the basics and find themselves happily text-rapping and avatar-hopping in no time.

Your Theater Needs Careful Planning

The server software itself is written in Python and comes with its own Web server, giving artists the opportunity to set up a stage easily, wherever their laptop is located on-line. Apart from the Web server, which requires the Python Twisted framework, the software makes extensive use of other open-source programs commonly installed on Linux systems, such as the text-to-speech-system Festival, the netpbm tools and gif2png. See the Problems with GIFs sidebar to this article for more details.

Often not shipped with Linux distributions are swfttools and the MP3 encoder lame. The timeout program from The Coroner's Toolkit, which is used during speech synthesis, also generally is not included. But it usually can be omitted if one isn't afraid to touch the source code.

The stage is a Flash client, and here is where the swfttools enter the picture. They convert the PNGs and JPEGs used both for stage decoration and as avatars into Flash format. Hence, performers and audience alike need the Macromedia Flash plugin for their Web browsers. KHTML- and Mozilla-based browsers work fine, but at present, Opera isn't suitable.

Unfortunately, at the time of this writing, the current version of UpStage does not honor PATH settings. Therefore, it is wise to check whether all the above-mentioned programs are situated in one of the directories that are hard-compiled into /bin/sh:

```
$ strings /bin/sh | grep -E "(bin|sbin)"
[...]
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:
/sbin:/bin
```

If not, appropriate links should be set. Otherwise, error hunting can become tricky, as UpStage isn't good at providing meaningful error messages in every situation. Things become even more complicated when using the sound tools. Despite UpStage using graphics tools in /usr/local/bin, it doesn't necessarily find lame there. So for users who aren't up to hacking the source, creating a link named /usr/bin/lame seems unavoidable.

Setting Up the Theater

Now it is time to start the server. Unpack the source archive, Upstage-2004-09-28.tar.gz, and enter the newly created Upstage directory. Here, you find the shell script go.sh that tries to kill an old twisted-server mentioned in the file Upstage/twistd.pid and starts a new one. So, don't worry about the relevant error message when you run `./go.sh` as a nonprivileged user for the first time. It's only then that Upstage creates

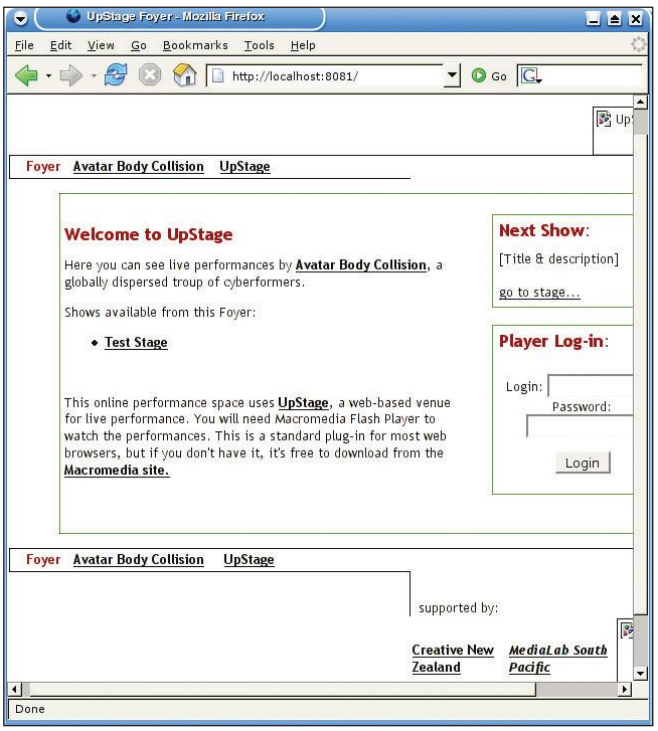


Figure 1. The default entrance hall clearly shows the origin of the software.

the pid-file.

For security reasons, it is not advisable to run UpStage as root. That's why the server uses an unprivileged port above 1024. The port on which your UpStage server runs can be configured. If you dislike the default port 8081, change the line:

```
WEB_PORT  =  8081
```

in Upstage/upstage/config.py, and re-run `./go.sh`.

Because the September 2004 version of UpStage is missing the directory that the server uses to store temporary MP3 files, you can save yourself a lot of trouble if you create it by hand:

```
mkdir html/speech
```

Now, point your local Web browser to the following: http://localhost:8081/, and you should end up at the entrance to your theater (Figure 1). To customize it according to your needs, change its HTML code in Upstage/html/index.html and the corresponding stylesheet, Upstage/html/style/main.css. It's a good idea to keep the relative link `"<a href="stages"></a>"` to the stages—your audience will be grateful—and the login for the artists.

The theater also has a back door for its personnel. The URLs http://localhost:8081/admin and http://localhost:8081/login.html lead you directly to a login dialog that can be changed in Upstage/html/login.html.

Hiring Personnel

The name of UpStage's default theater director is z, and z has no password. You probably want to change this, so log in and enter the theater's director. Using the Add a new player link, go to http://localhost:8081/admin/new/player and add the name and password of the new director. To make him or her the big boss who can hire and fire, make sure you tick the permission to Add or Remove Players (Figure 2).

This new player is written to the user configuration file, Upstage/config/players.xml, like this:

```
<player password="551a9c1c68844936b0d182080fe7dcc0"
 name="lj" rights="act,admin,su">
</player>
```

The password attribute doesn't contain the actual password, which is upstage for this example, but its md5sum. If you want to add users using your favorite text editor, you can generate the password like this:

```
$ echo -n "upstage" | md5sum
551a9c1c68844936b0d182080fe7dcc0  -
```

The name attribute contains the user name of the player, and you can grant up to three rights. The big boss needs the su right. Everyone who is supposed to create and edit things that can be seen and used on stage needs the admin permission, and all players need the right to act.

Unfortunately, the Web front end is quite buggy when it comes to deleting and editing users. It doesn't show you the correct rights, it doesn't allow you to change them (not even

Figure 2. LJ becomes a big boss.

PROBLEMS WITH GIFS

Even if you have installed gif2png properly, the September 2004 version of UpStage can't convert GIF pictures for use as avatars, props or backgrounds. Until a new version is available, you can fix this bug yourself by uncommenting line 38 in Upstage/img2swf.py and deleting `"giftopnm" flag "--background "#fff""` in line 63. The relevant lines then should read as follows:

```
[...]
35 def do_gif(tfn, swf):
[...]
38 #     os.path.remove(png)
[...]
57 def thumbnailer(filetype, tfn, thumb, log):
[...]
63            'image/gif'       :      'giftopnm %s |
pnmscale -height=10 | pnmtojpeg > %s'
```

with superuser power) and it doesn't let you delete users. If you click the check box before the relevant user entry in http://localhost:8081/admin/edit/player/ and press the Remove Players as a superuser button, UpStage removes the relevant player until the end of the session but doesn't delete him or her from players.xml. After restarting the server, all the players are alive and kicking again. Douglas Bagnall promised to fix this bug soon.

Fixing Up Roles and Props

These problems with users and permissions don't appear with the inventory of your theater. Using the workshop http://localhost:8081/admin/ URL, you can add and edit stages, avatars (an avatar complies with a character in your performance in one disguise), backdrops or stage designs and props. The latter can be carried by your avatar, and they always appear in the upper-left portion of the avatar, such as the blue bubbles attached to the bomb in Figure 6.

When creating new avatars, props and backdrops, you have some choices: two-dimensional pictures, Flash animations and video streams. Be careful with moving pictures, however; they require bandwidth and are real performance killers.

Video streams must be available locally and should be stored in Upstage/html/media/. For Linux, the UpStage user manual recommends webcamd as the software to use to upload a video stream by way of FTP. Unfortunately, webcamd's original project site seems to be closed (see Resources), but it still is available both as a binary and as a source archive from Debian servers.

Differing from real-world theater, an avatar, backdrop or prop can be assigned to multiple stages simultaneously. This is done in the Manage an existing stage section (Figure 3, http://localhost:8081/admin/edit/stage/<stagename>/).

The configuration data for the stages are stored in XML format in Upstage/config/stages.xml and Upstage/config/stages/<stage-id>/config.xml. The first file lists all available stages; each of the latter holds information about the inventory assigned to the appropriate stage.

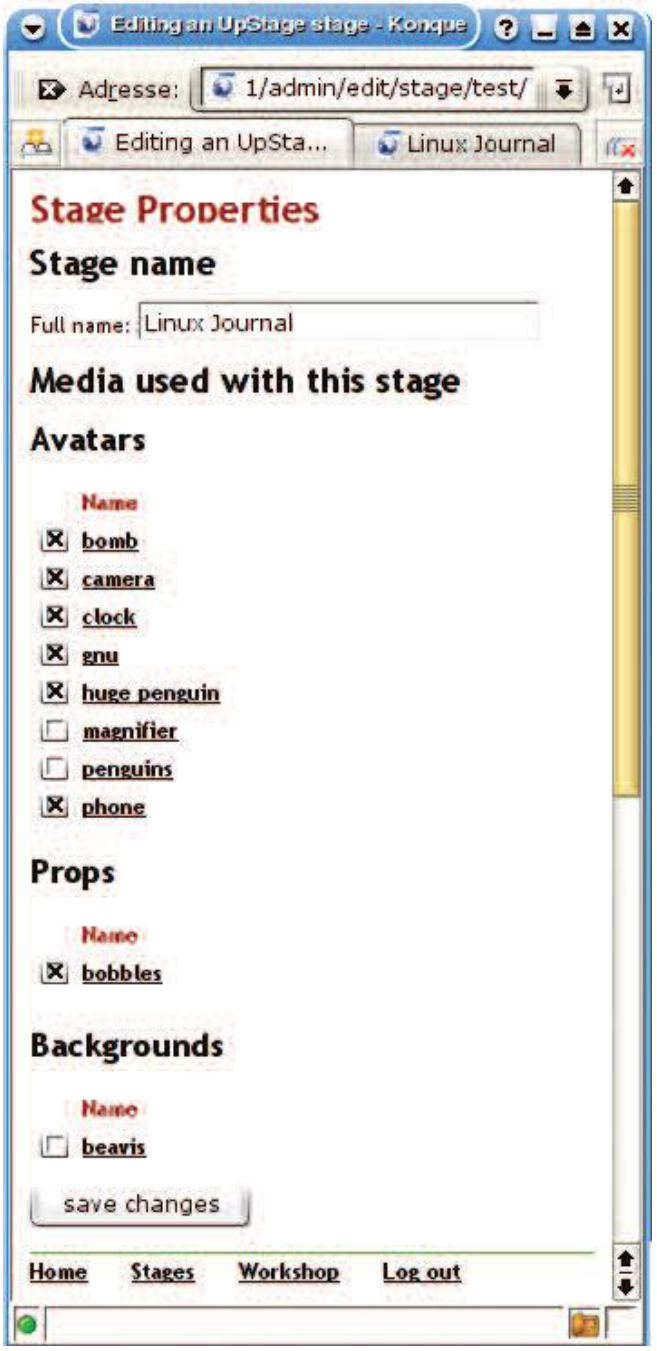


Figure 3. Although the inventory names are clickable, these links don't lead to the edit dialog for the relevant item but instead point to the Flash file.

Needless to say, the three types of inventory have their own text configuration files, namely Upstage/config/props.xml, avatars.xml and backdrops.xml. They all follow the structure shown in Listing 1.

Although the name of the root element does not actually matter, UpStage uses avatars, props and swamp, respectively, when generating the files. What matters is the name of the sub-elements: avatar, prop and backdrop. Each sub-element has four mandatory attributes plus one optional attribute, as described in Table 1.

Choose the http://localhost:8081/admin/edit/avatar/ link

Listing 1. The Structure of avatars.xml

```
configuration file

<avatars>
<avatar url="/media/Pbp9_q8I.swf" voice="ked"
name="huge penguin" file="Pbp9_q8I.swf"
thumbnail="/media/thumb/Pbp9_q8I.jpg">
</avatar>
<avatar url="/media/clock.swf" name="clock"
file="clock.swf" thumbnail="/media/thumb/clock.jpg">
</avatar>
</avatars>
```

Table 1. Attributes to Stage Inventory and Avatar Elements

Attribute	Value
url	Path to the relevant Flash file, starting with the media catalog below Upstage/html. UpStage generates random filenames. If you edit entries by hand, it is fine to use filenames suitable for humans.
name	The name of the item. It appears on stage, so choose carefully. To change it during performance, use the `/nick <name>` command, typed into the text input field below the chat window.
file	The filename of the relevant Flash file repeated, without the path. Thumbnail Path to the thumbnail in JPEG format, relative to the Upstage/html directory. UpStage stores them in Upstage/html/media/thumb. These thumbnails appear on stage to help players select items.
voice	This attribute affects avatars only and even here it is optional. It defines the voice used in text-to-speech synthesis. The voice names are defined in Upstage/upstage/config.py.

from the workshop and click the name of the relevant item to edit an existing avatar. The appropriate dialog (Figure 4) leaves you with two options, to change the item's name and voice.

Unfortunately, this dialog is of little help when it comes to estimating the size of the picture on stage. The UpStage client renders backdrops to fit the size of the browser window, while props and avatars appear about three times their original dimensions. The user manual (see Resources) contains a section with recommendations for sizes and formats for creating graphics.

Making Noise

When it comes to voice definitions, one no longer has to deal with XML—now it's Python. The file Upstage/upstage/config.py contains a section, actually a dictionary object, called VOICES that defines the commands used in text-to-speech synthesis (Listing 2). Having said this, UpStage speech generation does not depend on Festival exclusively. This is especially important for non-English speakers, because the Festival distribution as is limits itself to English.

If you want to add new voices, simply start a new line inside the curly braces following the VOICES keyword. Type the name of the new voice in single quote marks and add:

```
: ("| ", _fest),
```

Make sure you start the line with as many whitespaces as needed to place your opening single quote directly below the beginning of the other voice definitions. Python is picky about indentations, and incorrect indentations mean that UpStage stops working.

Following the pipe character (|), enter whatever command (pipeline) you like, provided it reads text from stdin and provides 16kHz raw PCM output on stdout. To test it, issue the

following command:

```
echo "Say something in the relevant language" |
<command> | timeout 15 lame -S -x -m s -r -s 16
--resample 22.05 --preset phone - /tmp/test.mp3
```

If an MP3 player playing the resulting /tmp/test.mp3 file says what it is meant to say, insert your command into config.py. Because UpStage is particular about paths, make sure you're using absolute paths in this file.

The original config.py file contains more text-to-speech commands than probably will work with your installation. Because all of them appear in the voice drop-down menu when adding or editing an avatar, it is wise to comment them out using the # sign. Notice that with the original voice definitions, you have to comment out two or three lines per item. If you miss one, you receive an error message such as this:

```
Failed to load application: invalid syntax
(config.py, line 92)
```

when restarting the server using `./go.sh` in order to activate the changes.

If all of your avatars lose their voices after this, you probably commented out the default voice definition as well. Bad idea! It's perfectly fine to redefine the command behind the default entry, but you must not leave UpStage without having one.

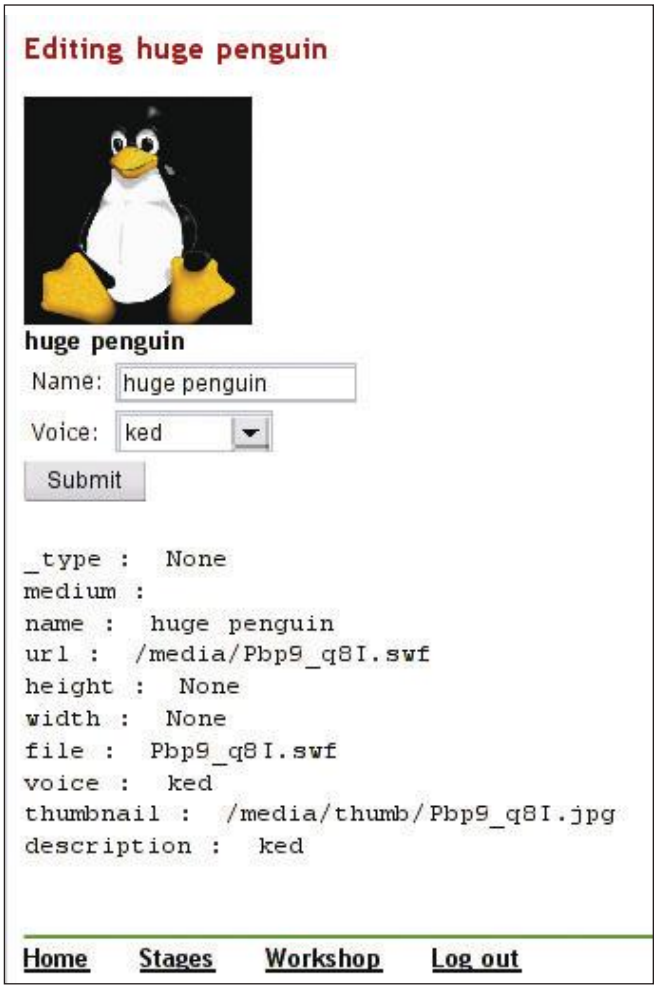


Figure 4. The Edit dialog of this avatar doesn't tell you this penguin is so big that it takes up almost the entire screen.

Listing 2. Voice Definitions in config.py

```
VOICES = {
          'kal': ("| timeout 15 text2wave -eval
'(voice_kal_diphone)' -otype raw",
                  _fest),
         [...]
         }
```

Rehearsal Time

When your stage is prepared, it is time to start rehearsing. This means all players need to log in and enter the relevant stage using the Stages link, http://localhost:8081/stages/, in the workshop. Once in, they at first find a big empty space, the stage, surrounded by the chat window to the right where all uttered text can be read. An image gallery is located beneath the chat window. Clicking one of the backdrop icons in the left part of the image gallery changes the stage design. The right part holds the props (Figure 5).

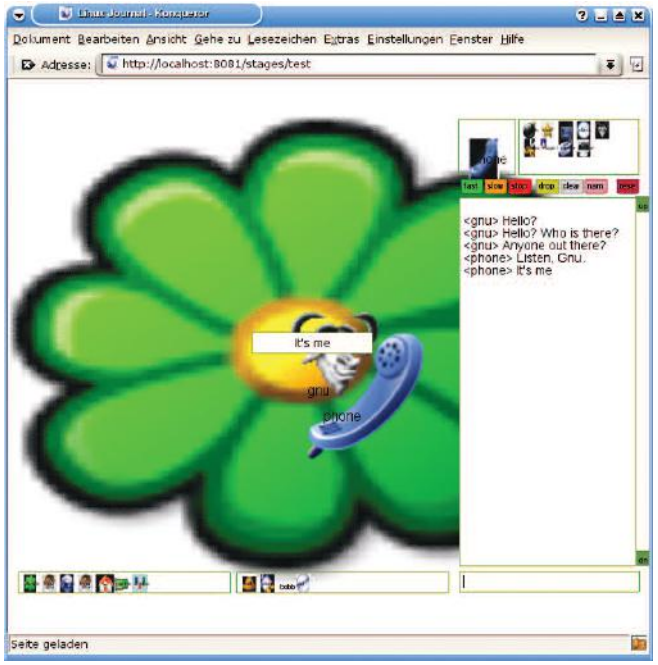


Figure 5. When choosing backdrops, one needs to consider that the outer-right portion will be obscured by the chat window.

Above the chat window users see a button bar that mainly serves to control avatars. The characters themselves can be found in the wardrobe above the buttons on the right-hand side. Here users find thumbnails of all avatars activated for this stage. If you click one of them, it appears in the mirror to the left of the wardrobe. Hence, a glimpse in the mirror always

shows you which role you are playing.

But, your character can't be seen on stage at once. If you type some text in the input field below the chat window, your avatar acts as a voice-over. When you first click on the stage window, the avatar appears there and its utterances can be read as balloons (Figure 5). Whether UpStage shows the avatar's name as text on stage can be triggered by using the pink name button.

When you click elsewhere on stage, your avatar moves slowly there. If you want it to jump there at once, click the green fast button first; the orange slow button brings you back into slow-motion mode. To bring the character to a full stop use the red stop button.

To equip your avatar with a prop, click the appropriate thumbnail in the right part of the image gallery below the stage window. It then follows your avatar in all its movements.

When you click another thumbnail in the wardrobe, your old character remains on stage but can be overtaken by your coplayers. When the avatar you currently hold needs to leave the stage, use the yellow drop button. At the moment, this also is the only way to get rid of a prop. Even though it is possible to change props by clicking another prop icon—although this is not done entirely without side effects—this current UpStage version has no "get rid of prop" button yet.

The gray clear button empties the stage except for the avatars your coplayers are holding. The entire operation, however, has a side effect: before your coplayers can move their characters again, they have to reselect them in the wardrobe.

Sometimes it might seem as though things haven't disappeared from the stage. In most cases, a browser reload helps, but then you need to grab your avatar again.

When for some reason you need to start from scratch, you can use the red reset button. This should not be done during a performance or when others are on the same stage, as it dramatically throws everyone off and requires a browser reload. Some players even may need to log in again. Moving the reset button to a less-tempting location is on the priority fix list.

If not logged in, one sees the stage and the chat window only (Figure 6). This however, doesn't mean the audience has no voice. Everything non-actors type in can be seen by everyone in the chat window, which makes UpStage a brilliant choice for on-line teaching and presentations. You can choose to respond or ignore the audience comments. The only differences are the audience text appears in gray font, without an avatar name attached, and it isn't spoken aloud. Hence the applause in UpStage is silent.

You can try it out even without installing UpStage. Every month Avatar Body Collision offers an open session for those interested in sampling and learning more about performing interactively with UpStage. Watch out for the next date (see Resources). Additional help is available through the user manual and the mailing list.

Resources for this article: www.linuxjournal.com/article/8056.

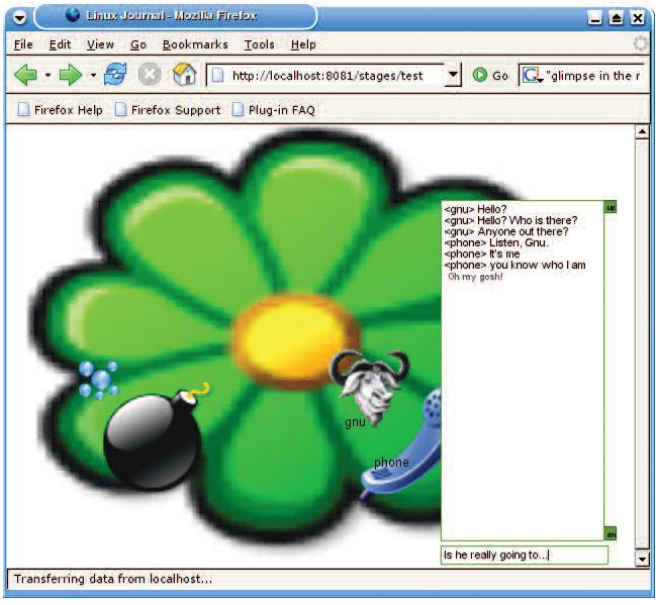


Figure 6. To applaud or to hoot, the audience can type into the chat.

Patricia Jung (trish@answergirl.de) works as an editor and sysadmin for Open Source Press (www.opensourcepress.de). As such, she is happy to have the privilege of dealing with Linux and UNIX exclusively.

My Favorite bash Tips and Tricks

Save a lot of typing with these handy bash features you won't find in an old-fashioned UNIX shell.

BY PRENTICE BISBAL

Bash, or the Bourne again shell, is the default shell in most Linux distributions. The popularity of the bash shell amongst Linux and UNIX users is no accident. It has many features to enhance user-friendliness and productivity. Unfortunately, you can't take advantage of those features unless you know they exist.

When I first started using Linux, the only bash feature I took advantage of was going back through the command history using the up arrow. I soon learned additional features by watching others and asking questions. In this article, I'd like to share some bash tricks I've learned over the years.

This article isn't meant to cover all of the features of the bash shell; that would require a book, and plenty of books are available that cover this topic, including *Learning the bash Shell* from O'Reilly and Associates. Instead, this article is a summary of the bash tricks I use most often and would be lost without.

Brace Expansion

My favorite bash trick definitely is brace expansion. Brace expansion takes a list of strings separated by commas and expands those strings into separate arguments for you. The list is enclosed by braces, the symbols { and }, and there should be no spaces around the commas. For example:

```
$ echo {one,two,red,blue}
one two red blue
```

Using brace expansion as illustrated in this simple example doesn't offer too much to the user. In fact, the above example requires typing two more characters than simply typing:

```
echo one two red blue
```

which produces the same result. However, brace expansion becomes quite useful when the brace-enclosed list occurs immediately before, after or inside another string:

```
$ echo {one,two,red,blue}fish
onefish twofish redfish bluefish

$ echo fish{one,two,red,blue}
fishone fishtwo fishred fishblue

$ echo fi{one,two,red,blue}sh
fionesh fitwosh firedsh fibluesh
```

Notice that there are no spaces inside the brackets or between the brackets and the adjoining strings. If you include spaces, it breaks things:

```
$ echo {one, two, red, blue }fish
{one, two, red, blue }fish

$ echo "{one,two,red,blue} fish"
{one,two,red,blue} fish
```

However, you can use spaces if they're enclosed in quotes outside the braces or within an item in the comma-separated list:

```
$ echo {"one ","two ","red ","blue "}fish
one fish two fish red fish blue fish

$ echo {one,two,red,blue}" fish"
one fish two fish red fish blue fish
```

You also can nest braces, but you must use some caution here too:

```
$ echo {{1,2,3},1,2,3}
1 2 3 1 2 3

$ echo {{1,2,3}1,2,3}
11 21 31 2 3
```

Now, after all these examples, you might be thinking to yourself, "Gee, those are great parlor tricks, but why should I care about brace expansion?"

Brace expansion becomes useful when you need to make a backup of a file. This is why it's my favorite shell trick. I use it almost every day when I need to make a backup of a config file before changing it. For example, if I'm making a change to my Apache configuration, I can do the following and save some typing:

```
$ cp /etc/httpd/conf/httpd.conf{,.bak}
```

Notice that there is no character between the opening brace and the first comma. It's perfectly acceptable to do this and is useful when adding characters to an existing filename or when one argument is a substring of the other. Then, if I need to see what changes I made later in the day, I use the diff command and reverse the order of the strings inside the braces:

```
$ diff /etc/httpd/conf/httpd.conf{.bak,}
1050a1051
> # I added this comment earlier
```

Command Substitution

Another bash trick I like to use is command substitution. To use command substitution, enclose any command that gener-

ates output to standard output inside parentheses and precede the opening parenthesis with a dollar sign, `$(command)`. Command substitution is useful when assigning a value to a variable. This is typical in shell scripts, where a common operation is to assign the date or time to a variable. It also is handy for using the output of one command as an argument to another command. If you want to assign the date to a variable, you can do this:

```
$ date +%d-%b-%Y
12-Mar-2004

$ today=$(date +%d-%b-%Y)

$ echo $today
12-Mar-2004
```

I often use command substitution to get information about several RPM packages at once. If I want a listing of all the files from all the RPM packages that have httpd in the name, I simply execute the following:

```
$ rpm -ql $(rpm -qa | grep httpd)
```

The inner command, `rpm -qa | grep httpd`, lists all the packages that have httpd in the name. The outer command, `rpm -ql`, lists all the files in each package.

Now, those of you who have experience with the Bourne shell might point out that you could perform command substitution by surrounding a command with back quotes, also called back-ticks. Using Bourne-style command substitution, the date assignment from above becomes:

```
today2=`date +%d-%b-%Y`

$ echo $today2
12-Mar-2004
```

There are two important advantages to using the newer bash-style syntax for command substitution. First, it can be nested more easily. Because the opening and closing symbols are different, the inner symbols don't need to be escaped with back slashes. Second, it is easier to read, especially when nested.

Even on Linux, where bash is standard, you still encounter shell scripts that use the older, Bourne-style syntax. This is done to provide portability to various flavors of UNIX that do not always have bash available but do have the Bourne shell. bash is backward-compatible with the Bourne shell, so it can understand the older syntax.

Redirecting Standard Error

Have you ever looked for a file using the `find` command, only to learn the file you were looking for is lost in a sea of `permission denied` error messages that quickly fill your terminal window?

If you are the administrator of the system, you can become root and execute `find` again as root. Because root can read any file, you don't get that error anymore. Unfortunately, not everyone has root access on the system being used. Besides, it's bad practice to be root unless it's absolutely necessary. So what can you do?

One thing you can do is redirect your output to a file. Basic output redirection should be nothing new to anyone who has spent a reasonable amount of time using any UNIX or Linux shell, so I won't go into detail regarding the basics of output redirection. To save the useful output from the find command, you can redirect the output to a file:

```
$ find /  -name foo > output.txt
```

You still see the error messages on the screen but not the path of the file you're looking for. Instead, that is placed in the file output.txt. When the find command completes, you can `cat` the file output.txt to get the location(s) of the file(s) you want.

That's an acceptable solution, but there's a better way. Instead of redirecting the standard output to a file, you can redirect the error messages to a file. This can be done by placing a 2 directly in front of the redirection angle bracket. If you are not interested in the error messages, you simply can send them to /dev/null:

```
$ find /  -name foo 2> /dev/null
```

This shows you the location of file foo, if it exists, without those pesky `permission denied` error messages. I almost always invoke the find command in this way.

The number 2 represents the standard error output stream. Standard error is where most commands send their error messages. Normal (non-error) output is sent to standard output, which can be represented by the number 1. Because most redirected output is the standard output, output redirection works only on the standard output stream by default. This makes the following two commands equivalent:

```
$ find / -name foo > output.txt
$ find / -name foo 1> output.txt
```

Sometimes you might want to save both the error messages and the standard output to file. This often is done with cron jobs, when you want to save all the output to a log file. This also can be done by directing both output streams to the same file:

```
$ find / -name foo > output.txt 2> output.txt
```

This works, but again, there's a better way to do it. You can tie the standard error stream to the standard output stream using an ampersand. Once you do this, the error messages goes to wherever you redirect the standard output:

```
$ find / -name foo > output.txt 2>&1
```

One caveat about doing this is that the tying operation goes at the end of the command generating the output. This is important if piping the output to another command. This line works as expected:

```
find -name test.sh 2>&1 | tee /tmp/output2.txt
```

but this line doesn't:

```
find -name test.sh | tee /tmp/output2.txt 2>&1
```

and neither does this one:

```
find -name test.sh 2>&1 > /tmp/output.txt
```

I started this discussion on output redirection using the find command as an example, and all the examples used the find command. This discussion isn't limited to the output of find, however. Many other commands can generate enough error messages to obscure the one or two lines of output you need.

Output redirection isn't limited to bash, either. All UNIX/Linux shells support output redirection using the same syntax.

Searching the Command History

One of the greatest features of the bash shell is command history, which makes it easy to navigate through past commands by navigating up and down through your history with the up and down arrow keys. This is fine if the command you want to repeat is within the last 10–20 commands you executed, but it becomes tedious when the command is 75–100 commands back in your history.

To speed things up, you can search interactively through your command history by pressing Ctrl-R. After doing this, your prompt changes to:

```
(reverse-i-search)`':
```

Start typing a few letters of the command you're looking for, and bash shows you the most recent command that contains the string you've typed so far. What you type is shown between the ` and ' in the prompt. In the example below, I typed in `htt`:

```
(reverse-i-search)`htt': rpm -ql $(rpm -qa | grep httpd)
```

This shows that the most recent command I typed containing the string htt is:

```
rpm -ql $(rpm -qa | grep httpd)
```

To execute that command again, I can press Enter. If I want to edit it, I can press the left or right arrow key. This places the command on the command line at a normal prompt, and I now can edit it as if I just typed it in. This can be a real time saver for commands with a lot of arguments that are far back in the command history.

Using for Loops from the Command Line

One last tip I'd like to offer is using loops from the command line. The command line is not the place to write complicated scripts that include multiple loops or branching. For small loops, though, it can be a great time saver. Unfortunately, I don't see many people taking advantage of this. Instead, I frequently see people use the up arrow key to go back in the command history and modify the previous command for each iteration.

If you are not familiar with creating for loops or other types of loops, many good books on shell scripting discuss this topic. A discussion on for loops in general is an article in itself.

You can write loops interactively in two ways. The first way, and the method I prefer, is to separate each line with a semicolon. A simple loop to make a backup copy of all the files in a directory would look like this:

```
$ for file in * ; do cp $file $file.bak; done
```

Another way to write loops is to press Enter after each line instead of inserting a semicolon. bash recognizes that you are creating a loop from the use of the for keyword, and it prompts you for the next line with a secondary prompt. It knows you are done when you enter the keyword done, signifying that your loop is complete:

```
$ for file in *
> do cp $file $file.bak
> done
```

And Now for Something Completely Different

When I originally conceived this article, I was going to name it "Stupid bash Tricks", and show off some unusual, esoteric bash commands I've learned. The tone of the article has changed since then, but there is one stupid bash trick I'd like to share.

About five years ago, a Linux system I was responsible for ran out of memory. Even simple commands, such as ls, failed with an `insufficient memory` error. The obvious solution to this problem was simply to reboot. One of the other system administrators wanted to look at a file that may have held clues to the problem, but he couldn't remember the exact name of the file. We could switch to different directories, because the cd command is part of bash, but we couldn't get a list of the files, because even ls would fail. To get around this problem, the other system administrator created a simple loop to show us the files in the directory:

```
$ for file in *; do echo $file; done
```

This worked when ls wouldn't, because echo is a part of the bash shell, so it already is loaded into memory. It's an interesting solution to an unusual problem. Now, can anyone suggest a way to display the contents of a file using only bash built-ins?

Conclusion

The bash shell has many great features to make life easier for its users. I hope this summary of bash tricks I like to use has shown you some new ways to take advantage of the power bash has to offer.

Prentice Bisbal started using Linux in January 1997 with Red Hat Linux 4.0 on a 486. He has been maintaining Linux systems professionally since 1998. He is a system administrator for a pharmaceutical company in central New Jersey.

Making Connections. Building Your Business.

From the technological advances you'll find in conference sessions, and the partnerships forged on the show floor, to the peering agreements made during social events, ISPCON connects you with the people and technology that will build your business.

As the Internet service providers' number one industry meeting place, ISPCON is the only show totally focused on ISPs – the pioneers who are building and deploying the global Internet.

Register online and use **Customer Code LNXS05** to take advantage of our huge registration discount!

THE #1 CONFERENCE IN THE INDUSTRY

An unmatched six-track conference that delves into the nitty-gritty of overclocking your business to exploit opportunities with knowledge and vision.

THE #1 PRODUCT SHOWCASE

The ISPCON exhibit hall is the only place to see everything at once, touch what's next and buy what's best.

THE #1 GATHERING OF SMART MAVERICKS

Only the brightest, risk-tolerant people can prosper while challenging huge monopolies. Having lunch with a few hundred minds like yours is invaluable.

ISPCON

The Definitive Event for Wired & Wireless ISPs

May 24-26, 2005
Baltimore Convention Center
Baltimore, MD
www.ispcon.com

File Synchronization with Unison

Keeping directories in sync on multiple machines can be difficult. Running Unison is one way to make the task easier. BY ERIK INGE BOLSØ

Unison is a file-synchronization tool that runs on Linux, UNIX and Microsoft Windows. Those of you who've used IBM Lotus Notes or Intellisync Mobile Suite probably have an idea of what synchronization is good for, as compared to one-way mirroring options such as rsync. You might have mirrored a company document directory to your laptop, for example, and then modified a document or two. Other people might have modified other documents in the same directory by the time you get back. With rsync, you'd need to reconcile the differences between the two directories manually or risk overwriting someone's changes. Unison can sort out what has changed where, propagate the changed files and even merge different changes to the same file if you tell it how.

Think of Unison as two-way rsync with a bit of revision control mixed in. The most common use is keeping your local and remote home directory, or some data directory you often use in different contexts, in sync. It uses the rsync algorithm to keep network traffic down and should be tunneled through SSH over untrusted networks. No extra work is needed—simply specify ssh:// when adding a directory location. Quite a bit of extra disk space often is needed for Unison, though, because the synchronizer needs to keep track of what the files looked like on the last run.

Getting, Compiling and Installing Unison

Unison's home page is maintained at the University of Pennsylvania; the project leader, Benjamin C. Pierce, is a professor in the Department of Computer and Information Science. See the on-line Resources for the URL.

Unison isn't as widely deployed as rsync, so you might not be able to find a precompiled package for your distribution. But the binaries downloadable from the Unison home page should work for most people.

If you'd like to compile from source, you can. A few extra hoops must be jumped through, however, because Unison is programmed in OCaml, not the most common language. See Resources if there is no handy package for your distribution.

Compiling and installing Unison is simple; type `make UISTYLE=xxx`. The GTK user interface needs additional OCaml bindings for GTK, so I use the text interface in this article. Typing `make UISTYLE=text` or `make UISTYLE=gtk` should give you a Unison executable. Simply copy the executable to somewhere in the path on both machines you want to synchronize.

In this article, I'm using the current stable version of Unison, 2.9.1, unless otherwise noted. You need to use the latest betas if you're going to synchronize files larger than 2GB.

The developer versions tend to work well. They are what the developers run themselves on their own precious data. Sign up for the unison-hackers mailing list if you feel a bit adventurous. Jerome Vouillon, Benjamin C. Pierce and Trevor Jim tend to hang out there discussing improvements. Commit logs also float by, so you can track what is going on.

Configuring and Using Unison

Unison keeps its config and working files in a .unison directory in your home directory or wherever you want to put it. Set the UNISON environment variable to specify an alternate location.

The default configuration is stored in .unison/default.prf. Listing 1 shows a plain config file suitable for testing. Synchronizing two directories is now as simple as:

```
$ unison /nfsmount/dir1 /home/me/dir1
```

Listing 1. .unison/default.prf

```
# Unison preferences file
merge = diff3 -m CURRENT1 OLD CURRENT2 > NEW
backup = Name *
maxbackups = 10
log = true
logfile = /home/knan/.unison/unison.log
rshargs = -C
```

Unison then asks the user about any differences between the directories and offers reasonable defaults. It does take a bit of time to get used to Unison's way of thinking, however. And, Unison is no substitute for backups. Unison happily propagates back the deletion of all the files in one replica, for example, which can be a rude awakening for programmers used to CVS. For example:

```
rm dir1/* ; unison ssh://server/dir1 dir1
```

doesn't do what you expect from a:

```
rm dir1/*; cvs update dir1
```

Deleting a file is an action that is replicated on the other side upon synchronization. So, this example command removes all files in dir1 on both sides.

Once you feel comfortable, consider adding `auto = true` to the Unison profile. This skips questions about any non-conflicting changes but gives you a chance to back out at the end.

The Unison manual is recommended reading. It is clear and well written and explains what happens at most corner cases.

Keeping Home Directories in Sync

Once users become familiar with Unison, a common thought is to use it for keeping one's home directory in sync between machines, say, your laptop and desktop. This can be realized pretty easily. Listing 2 has a simple profile that does the job, but you probably want to extend it. Listing 2, for example, ignores MP3 files and Unison's own files and demonstrates the use of `include` for having common settings applied to all profiles.

Listing 2. .unison/home.prf

```
# Unison preferences file
root = /home/erik
root = ssh://remotehost/home/erik
# exactly two or none "root" lines
ignore = Name *.mp3
# ignore all .mp3 files anywhere
ignore = Path .unison
# ignore all files with .unison somewhere in their
full path
include default
# imports settings from default.prf
```

Test our new profile like this:

```
$ unison home -testserver
```

And invoke it like this:

```
$ unison home -batch
$ unison home
```

The -batch run takes care of the easy cases without asking, backing up and logging as needed, and the second run asks you about any tricky business—like merging.

The `root =` lines can be omitted if you want to specify the files to be synchronized on the command line instead. The lines are equivalent to this invocation:

```
$ unison home /home/erik ssh://remotehost/home/erik
```

Merging Conflicting Changes

In order to do a three-way merge, backups must be enabled. By default, with backups disabled, Unison keeps only a checksum and metadata, such as permissions, so it has no unmodified file to reference.

In version 2.9.1 of Unison, if you choose merge for a conflict and the merge is successful without manual intervention, the changes are propagated immediately, which doesn't give you a chance to back out. So, if you have the space, I suggest

leaving maxbackups at 5 or so, instead of the default 2, to leave yourself the chance of recovering from automatic mismerges. Contents of the backup directory after a merge look like this:

```
$ ls -1 .unison/backup/
shared.txt                merged version ("NEW")
shared.txt.1.unibck       changed remotely ("CURRENT2")
shared.txt.2.unibck       changed locally ("CURRENT1")
shared.txt.3.unibck       old version ("OLD")
```

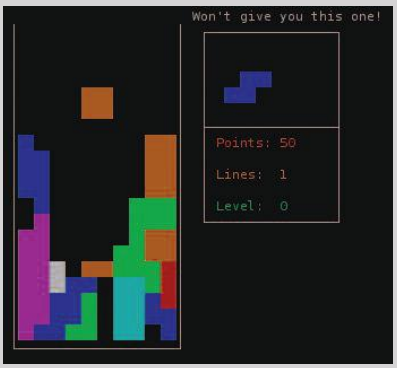
As of the newest beta, 2.10.3 at the time of this writing, Unison can invoke different merge programs for different files. You might want to use 3DM to merge XML files, for example, or a database merge tool for your Berkeley databases. This functionality still is new and subject to change. It has been noted by the project leader that the merge functionality was in need of a rewrite and didn't really work too well in 2.9.1 and 2.9.20. Thus, if you intend to do much merging, you will be better off tracking the bleeding edge.

Resources for this article: www.linuxjournal.com/article/8059.

Erik Inge Bolsø is a UNIX consultant and épée fencer who lives in Molde, Norway, and has been running Linux since 1996. Another of his hobbies can be found by doing a Google search for "balrog genealogy", and he can be reached at ljcomment@tvilsom.org.

Bastet

fph.altervista.org/prog/bastet.shtml



Everyone likes a nice game of *Tetris* every now and then, but what about an evil game of *Tetris*? Frederico Poloni's *Bastet*, short for Bastard Tetris, is a *Tetris* clone that picks the worst possible piece to fit into your stack. It's based on *petris* by Peter Seidler and has a lean, mean interface. No music and no fancy graphics, only the bastard AI and you.

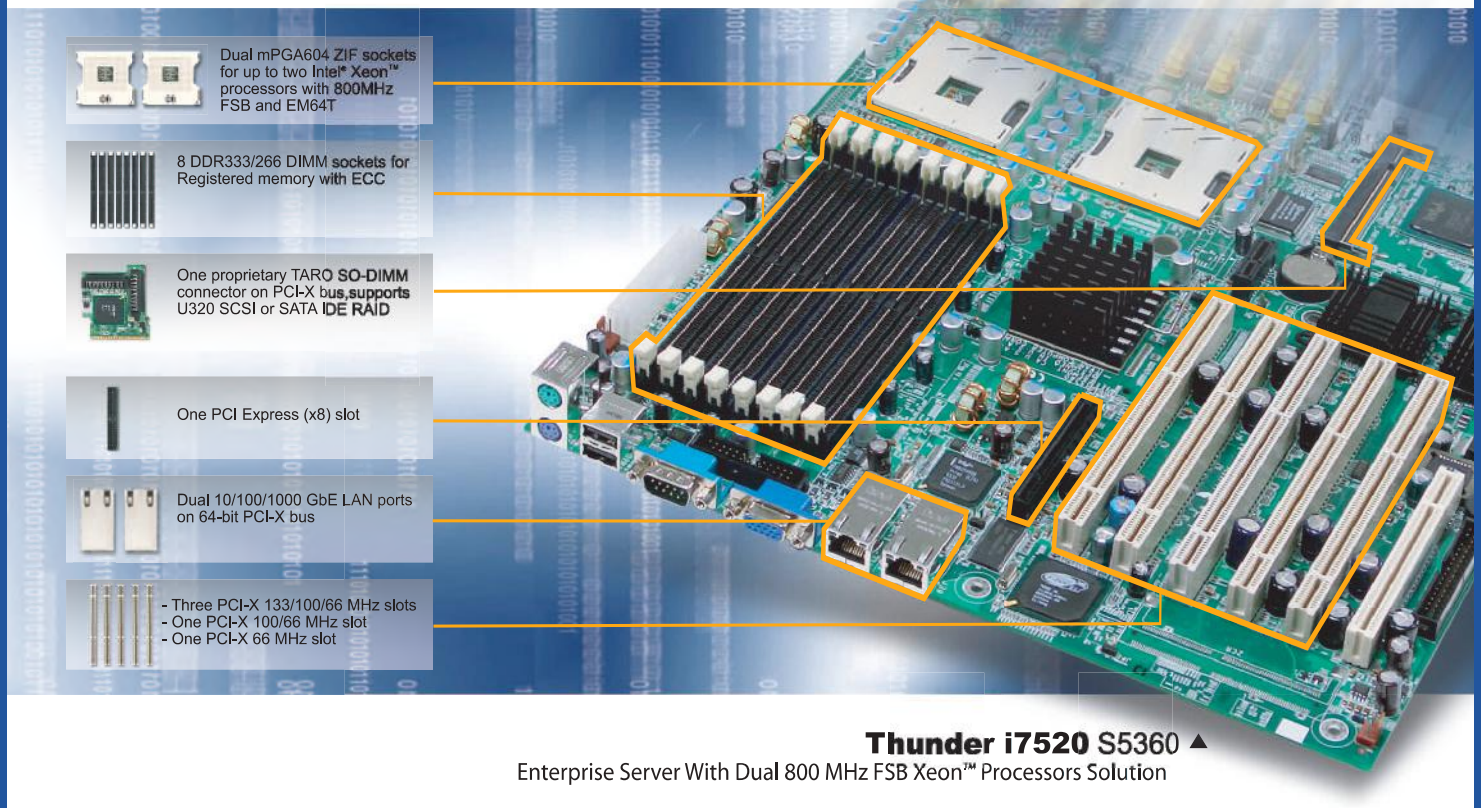
Whether you're working on your *Tetris* game and hoping to go pro, celebrating the 20th anniversary of the original *Tetris*, or simply looking for an implementation that will make you lose and get you back to work sooner, this is a game to try.

—DON MARTI

TYAN

Fast, Flexible, and Feature-Rich!

PCI Express and EM64T Servers Have Arrived



Thunder i7520 S5360 ▲
Enterprise Server With Dual 800 MHz FSB Xeon™ Processors Solution

intel inside XEON

Thunder i7520 S5360

- Supports two Intel® Xeon™ processors with 800 MHz FSB and EM64T
- 8 DIMMs for DDR266/333 memory
- Three PCI-X 133/100/66 MHz slots, one PCI-X 100/66 MHz slot, one PCI-X 66 MHz slot and one 33MHz PCI slot
- One PCI Express™ x8 slot
- One proprietary SO-DIMM connector on PCI-X bus, supports U320 SCSI or SATA
- Dual GbE LAN

Tiger i7320 S5350

- Supports two Intel® Xeon™ Processor with 800MHz FSB and EM64T
- 8 DIMMs for DDR266/333 memory
- Two PCI-X 64/66 MHz slots; three 32/33 PCI 2.3 slots
- One proprietary SO-DIMM connector on PCI-X bus, supports U320 SCSI or SATA
- Dual PCI Express GbE LAN

TYAN COMPUTER CORP.

Tyan Computer USA
3288 Laurelview Court
Fremont, CA 94538 USA
Tel: +1-510-651-8868 Fax: +1-510-651-7688
Pre-Sales Tel: +1-510-651-8868 x5120
Email: marketing@tyan.com

For more information about this and other Tyan products, please contact Tyan Pre-Sales at (510) 651-8868 x5120, or contact your local Tyan system integrator/reseller.

www.tyan.com

Using C for CGI Programming

You can speed up complex Web tasks while retaining the simplicity of CGI. With many useful libraries available, the jump from a scripting language to C isn't as big as you might think. BY CLAY DOWLING

Perl, Python and PHP are the holy trinity of CGI application programming. Stores have shelves full of books about these languages, they're covered well in the computer press and there's plenty on the Internet about them. A distinct lack of information exists, however, on using C to write CGI applications. In this article, I show how to use C for CGI programming and lay out some situations in which it provides significant advantages.

I use C in my applications for three reasons: speed, features and stability. Although conventional wisdom says otherwise, my own benchmarks have found that C and PHP are equivalent in speed when the processing to be done is simple. When there is any complexity to the processing, C wins hands-down.

In addition, C provides an excellent feature set. The language itself comes with a bare-bones set of features, but a staggering number of libraries are available for nearly any job for which a computer is used. Perl, of course, is no slouch in this area, and I don't contend that C offers more extensibility, but both can fill nearly any bill.

Furthermore, CGI programs written in C are stable. Because the program is compiled, it is not as susceptible to changes in the operating environment as PHP is. Also, because the language is stable, it does not experience the dramatic changes to which PHP users have been subjected over the past few years.

The Application

My application is a simple event listing suitable for a business to list upcoming events, say, the meeting schedule for a day or the events at a church. It provides an administrative interface intended to be password-protected and a public interface that lists all upcoming events (but only upcoming events). This application also provides for runtime configuration and interface independence.

I use a database, rather than write my own data store, and a configuration file contains the database connection information. A collection of files is used to provide interface/code separation.

The administrative interface allows events to be listed, edited, saved and deleted. Listing events is the default action if no other action is provided. Both new and existing events can be saved. The interface consists of a grid screen that displays the list of events and a detail screen that contains the full record of a single event.

Listing 1. MySQL Schema

```
CREATE TABLE event (
  event_no int(11) NOT NULL auto_increment,
  event_begin date NOT NULL default '0000-00-00',
  name varchar(80) NOT NULL default '',
  location varchar(80) NOT NULL default '',
  begin_hour varchar(10) default NULL,
  end_hour varchar(10) default NULL,
  event_end date NOT NULL default '0000-00-00',
  PRIMARY KEY  (event_no),
  KEY event_date (event_begin)
)
```

The database schema for this application consists of a single table, defined in Listing 1. This schema is MySQL-specific, but an equivalent schema can be created for any database engine.

The following functions are the minimum necessary to implement the functionality of the administrative interface: list_events(), show_event(), save_event() and delete_event(). I also am going to abstract the reading and writing of database data into their own group of functions. This keeps each function simpler, which makes debugging easier. The functions that I need for the data-storage interface are event_create(), event_destroy(), event_read(), event_write and event_delete. To make my life easier, I'm also going to add event_fetch_range(), so I can choose a range of events—something I need to do in at least two places.

Next, I need to abstract my records to C structures and abstract database result sets to linked lists. Abstraction lets me change database engines or data representation with relatively little expense, because only a little part of my code deals directly with the data store.

There isn't room here to print all of my source code. Complete source code and my Makefile can be downloaded from my Web site (see the on-line Resources).

Tools

The first hurdle to overcome when using C is acquiring the set of tools you need. At bare minimum, you need a CGI parser to break out the CGI information for you. Chances are good that you're also looking for some database connectivity. A little bit of logic/interface independence is good too, so you aren't rewriting code every time the site needs a makeover.

For CGI parsing, I recommend the cgic library from Thomas Boutell (see Resources). It's shockingly easy to use and provides access to all parts of the CGI interface. If you're

a C++ person, the cgicc libraries also are suitable (see Resources), although I found the Boutell library to be easier to use.

MySQL is pretty much the standard for UNIX Web development, so I stick with it for my sample application. Every significant database engine has a functional C interface library, though, so you can use whatever database you like.

I'm going to provide my own interface-independence routines, but you could use libxml and libxslt to do the same thing with a good deal more sophistication.

Runtime Configuration

At runtime, I need to be able to configure the database connection. Given a filename and an array of character strings for the configuration keys, my configuration function populates a corresponding array of configuration values, as shown in Listing 2. Now I can populate a string array with whatever keys I've chosen to use and get the results back in the value array.

Listing 2. Runtime Configuration Function

```
void config_read(char* filename, char** key,
                 char** value) {

  FILE* cfile;
  char tok[80];
  char line[2048];
  char* target;
  int i;
  int length;

  cfile = fopen(filename, "r");
  if (!cfile) {
    perror("config_read");
    return;
  }

  while(fgets(line, 2048, cfile)) {
    if ((target = strchr(line, '='))) {
      sscanf(line, "%80s", tok);
      for(i=0; key[i]; i++) {
        if (strcmp(key[i], tok) == 0) {
          target++;
          while(isspace(*target)) target++;
          length = strlen(target);
          value[i] = (char*)calloc(1, length + 1);
          strcpy(value[i], target);
          target = &value[i][length - 1];
          while(isspace(*target)) *target-- = 0;
        }
      }
    }
  }
  fclose(cfile);

}
```

User Interface

The user interface has two parts. As a programmer, I'm concerned primarily with the input forms and URL strings. Everybody else cares how the page around my form looks and takes the form itself for granted. The solution to keep both parties happy is to have the page exist separately from the form and my program.

Templating libraries abound in PHP and Perl, but there are no common HTML templating libraries in C. The easiest solution is to include only the barest minimum of the output in my C code and keep the rest in HTML files that are output at the appropriate time. A function that can do this is found in Listing 3.

Listing 3. HTML Template Function

```
void html_get(char* path, char* file) {

  struct stat sb;
  FILE* html;
  char* buffer;
  char fullpath[1024];

  /* File & path name exceed system limits */
  if (strlen(path) + strlen(file) > 1024) return;

  sprintf(fullpath, "%s/%s", path, file);
  if (stat(fullpath, &sb)) return;

  buffer = (char*)calloc(1, sb.st_size + 1);
  if (!buffer) return;
  html = fopen(fullpath, "r");
  fread((void*)buffer, 1, sb.st_size, html);
  fclose(html);
  puts(buffer);
  free(buffer);

}
```

Before generating output, I need to tell the Web server and the browser what I'm sending; cgiHeaderContentType() accomplishes this task. I want a content type of text/html, so I pass that as the argument. The general steps to follow for any page I want to display are:

- cgiHeaderContentType("text/html");
- html_get(path, pagetop.html);
- Generate the program content.
- html_get(path, pagebottom.html);

Form Processing

Now that I can generate a page and print a form, I need to be able to process that form. I need to read both numeric and text elements, so I use a couple of functions from the cgic library: cgiFormStringNoNewlines() and cgiFormInteger(). The cgic library implements the main function and requires that I imple-

Listing 4. save_event(), Parsing CGI Data

```
struct event* e;

e = event_create();
cgiFormInteger("eventno", &e->event_no, 0);
cgiFormStringNoNewlines("name", e->name, 80);
cgiFormStringNoNewlines("location",
                        e->location, 80);

/* Processing date fields */
cgiFormInteger("beginyear",
               &e->event_begin->year, 0);
cgiFormInteger("beginmonth",
               &e->event_begin->month, 0);
cgiFormInteger("beginday", &e->event_begin->day, 0);
cgiFormInteger("endyear", &e->event_end->year, 0);
cgiFormInteger("endmonth", &e->event_end->month, 0);
cgiFormInteger("endday", &e->event_end->day, 0);

/* Process begin & end times separately */
cgiFormStringNoNewlines("beginhour",
                        e->event_begin->hour, 10);
cgiFormStringNoNewlines("endhour",
                        e->event_end->hour, 10);

event_write(e);

cgiHeaderLocation(cgiScriptName);
```

ment int cgiMain(void). cgiMain() is where I put the bulk of my form processing.

To display a single record in my show_event function, I get the event_no (my primary key) from the CGI parameter eventno. cgiFormInteger() retrieves an integer value and sets a default value if no CGI parameter is provided.

I also need to get a whole raft of data from the form in save_event. Dates are thorny things to input because they consist of three pieces of data: year, month and date. I need both a begin and an end date, which gives me six fields to interpret. I also need to input the name of the event, begin and end times (which are strings because they might be events themselves, such as sunrise or sunset) and the location. Listing 4 shows how this works in code.

Listing 4 also demonstrates cgiHeaderLocation(), a function that redirects the user to a new page. After I've saved the submitted data, I want to show the event listing page. Instead of a literal string, I use one of the variables that libcgic provides, cgiScriptName. Using this variable instead of a literal one means the program name can be changed without breaking the program.

Finally, I need a way to handle the submit buttons. They're the most complex input, because I need to launch a function based on their values and select a default value, just in case. The cgic library has a function, cgiFormSelectSingle(), that emulates this behavior exactly. It requires the list of possible values to be in an array of strings. It populates an integer variable with the index of the parameter in the array or uses a default value if there are no matches.

Listing 5. Handling Submit Buttons

```
char* command[5] = {"List", "Show",
                    "Save", "Delete", 0};
void (*action)(void)[5] = {list_events,
  show_event, save_event, delete_event, 0};
int result;

cgiFormSelectSingle("do", command, 4, &result, 0);
action[result]();
```

See Resources for information on function pointers. If function pointers still baffle you, you can choose the function to run in a switch statement. I prefer the array of function pointers because it is more compact, but my older code still makes use of the switch statement.

Database System

MySQL from C is largely the same as PHP, if you're used to that interface. You have to use MySQL's string escape functions to escape problematic characters in your strings, such as quote characters or the back slash character, but otherwise it is basically the same. The show_event() function requires me to

fetch a single record from the primary key. All of the error checking bulks up the code, but it's really three basic statements. A call to mysql_query() executes the MySQL statement and generates a result set. A call to mysql_store_result() retrieves the result set from the server. Finally, a call to mysql_fetch_row() pulls a single MYSQL_ROW variable from the result set.

The MYSQL_ROW variable can be treated like an array of strings (char**). If any of the data is numeric and you want to treat it as numeric data, you need to convert it. For instance, in my application it is desirable to have the date as three separate numeric components. Because this data is structured as YYYY-MM-DD, I use sscanf() to get the components (Listing 6).

Listing 6. Retrieving Data from MySQL

```
MYSQL_RES* res;
MYSQL_ROW row;
int beginyear;
int beginmonth;
int beginday;

if (mysql_query(db, sql)) {
  print_error(mysql_error(db));
  return;
}
if((res = mysql_store_result(db)) == 0) {
  print_error(mysql_error(db));
  return;
}
if ((row = mysql_fetch_row(res)) == 0) {
  print_error("No event found by that number");
  return;
}

sscanf(row[0], "%d-%d-%d", &beginyear, &beginmonth,
       &beginday);
```

Writing data to the database is more interesting because of the need to escape the data. Listing 7 shows how it is done.

Listing 7. Using User-Supplied Data in MySQL

```
char name[11];
char escapedname[21];

cgiFormStringNoNewlines("name", name, 10);
mysql_real_escape_string(db, escapedname, name,
                         strlen(name));
```

escapedname holds the same string as name, with MySQL special characters escaped so I can insert them into an SQL statement without worry. It is essential that you escape all strings read from user input; otherwise, a devious person could take advantage of your lapse and do unpleasant things to your database.

Debugging CGI Programs

One distinct disadvantage of debugging C is that errors tend to cause a segmentation fault with no diagnostic message about the source of the error. Debuggers are fine for most other types of programs, but CGI programs present a special challenge because of the way they acquire input.

To help with this challenge, the cgic library includes a CGI program called capture. This program saves to a file any CGI input sent to it. You need to set this filename in capture's source code. When your CGI program needs debugging, add a call to cgiReadEnvironment(char*) to the top of your cgiMain() function. Be sure to set the filename parameter to match the filename set in capture. Then, send the problematic data to capture, making it either the action of the form or the script in your request. You now can use GDB or your favorite debugger to see what sort of trouble your code has generated.

You can take some steps to simplify later debugging and development. Although these apply to all programming, they pay off particularly well in CGI programming. Remember that a function should do one thing and one thing only, and test early and test often.

It's a good idea to test each function you write as soon as possible to make sure it performs as expected. And, it's not a bad idea to see how it responds to erroneous data as well. It's highly likely that at some point the function will be given bad data. Catching this behavior ahead of time can save unpleasant calls during your off hours.

Deployment

In most situations, your development machine and your deployment machine are not going to be the same. As much as possible, try to make your development system match the production system. For instance, my software tends to be developed on Linux or OpenBSD and nearly always is deployed on FreeBSD.

When you're preparing to build or install on the deployment machine, it is particularly important to be aware of differences in library versions. You can see which dynamic libraries your code uses with `ldd`. It's a good idea to check this information, because you often may be surprised by what additional dependencies your libraries bring.

If the library versions are close, usually reflected in the same major number, there probably isn't a big problem. It's not uncommon for deployment and development machines to have incompatible versions if you're deploying to an externally hosted Web site.

The solution I use is to compile my own local version of the library. Remove the shared version of the library, and link against this local version rather than the system version. It bulks up your binary, but it removes your dependency on libraries you don't control.

Once you have built your binary on the deployment system, run `ldd` again to make sure that all of the dynamic libraries have been found. Especially when you are linking against a local copy of a library, it's easy to forget to remove the dynamic version, which won't be found at runtime (or by `ldd`). Keep tweaking the build process; build and recheck until there are no unfound libraries.

If function pointers still baffle you, you can choose the function to run in a switch statement.

Speed: CGI vs. PHP

Conventional wisdom holds that a program using the CGI interface is slower than a program using a language provided by a server module, such as mod_php or mod_perl. Because I started writing Web applications with PHP, I use it here as my basis for comparison with a CGI program written in C. I make no assertions about the relative speed of C vs. Perl.

The comparison that I used was the external interface to the database (events.cgi and events.php), because both used the same method for providing interface separation. The internal interface was not tested, as calls to the external interface should dwarf calls to the internal.

Apache Benchmark was used to hit each version with 10,000 queries, as fast as the server could take it. The C version had a mean transaction time of 581ms, and the PHP version had a mean transaction time of 601ms. With times so close, I suspect that if the tests were repeated, some variation in time would be seen. This proved correct, although the C version was slightly faster than the PHP version more times than not.

My normal development uses a more complex interface separation library, libtemplate (see Resources). I have PHP and C versions of the library. When I compared versions of the event scheduler using libtemplate, I found that C had a much more favorable response time. The mean transaction time for the C version was 625ms, not much more than it was for the simpler version. The PHP version had a mean transaction time of 1,957ms. It also was notable that the load number while the PHP version was running generally was twice what was seen while the C version was running. No users were on the system, and no other significant applications were running when this test was done.

The fairly close times of the two C versions tell us that most of the execution time is spent loading the program. Once the program is loaded, the program executes quite quickly. PHP, on the other hand, executes relatively slowly. Of course, PHP doesn't escape the problem of having to be loaded into memory. It also must be compiled, a step that the C program has been through already.

Conclusions

With the right tools and a little experience, developing CGI applications with C is no more difficult than it is when using Perl or PHP. Now that I have the experience and the tools, C is my preferred language for CGI applications.

C excels when the application requires more advanced processing and long-term stability. It is not especially susceptible to failure when server changes are beyond your control, unlike PHP. Short of removing a shared library, such as libc or libmysqlclient, the C version of our application is hard to break. The speed of execution for C programs makes it a clear choice when the application requires more complex data processing.

Resources for this article: www.linuxjournal.com/article/8058.

Clay Dowling is the president of Lazarus Internet Development (www.lazarusid.com). In addition to programming, he enjoys brewing beer and wine. He can be reached by e-mail at clay@lazarusid.com.

Part III: AFS—A Secure Distributed Filesystem

Make your single sign-on infrastructure complete using a secure cross-platform distributed filesystem.

BY ALF WACHSMANN

The Andrew File System (AFS) is a secure distributed global filesystem that provides location independence, scalability and transparent migration capabilities for data. AFS works across a multitude of operating systems and is used at many large sites that have been in production for many years.

AFS provides unique features that are not available with other distributed filesystems, even though AFS is almost 20 years old. This age might make it less appealing to some, but with IBM making AFS available as open source in 2000, new interest sparked in its use and development. This article discusses the rich features AFS offers and invites readers to play with it.

Features and Benefits of AFS

AFS client software is available for Linux and for UNIX flavors from HP, Compaq, IBM, Sun and SGI. It also is available for Microsoft Windows and Apple's Mac OS X. This makes AFS the ideal filesystem for data sharing between platforms across local and wide area networks.

All AFS client machines have a local cache. A cache manager keeps track of users on a machine and handles the data requests coming from them. Data caching happens in chunks of files, which are copied from an AFS file server to local disk. The cache is shared between all users of a machine and persists over reboots. This local caching reduces network traffic and makes subsequent access to cached data much faster.

AFS is organized in a globally unique namespace. A global view of the AFS file space is shown in Figure 1. Pathnames leading to files are not only the same wherever the data is accessed, the pathnames do not contain any server information. In other words, the AFS user does not know on which file server the data is located. To make this work, AFS has a replicated data location database that a client has to contact in order to find data. This is unlike the Network File System (NFS), in which the client has the information about the file server hosting a particular part of the NFS filesystem.

The different independent AFS domains are called cells and correspond to Kerberos realms. A typical AFS pathname looks like this: /afs/cern.ch/user/a/alf/Projects/. This pathname contains the AFS cell name but not the file server name.

This location independence allows AFS administrators to move data from one AFS server to another without any visible changes to users. It also makes AFS easily scalable. If you run out of space or network capacity on your AFS file servers, simply add another one and migrate data to the new server. Clients do not notice this location change. AFS also scales well in terms of the number of clients per file server. On modern hardware, one AFS file server can serve up to about 1,000 clients without any problems.

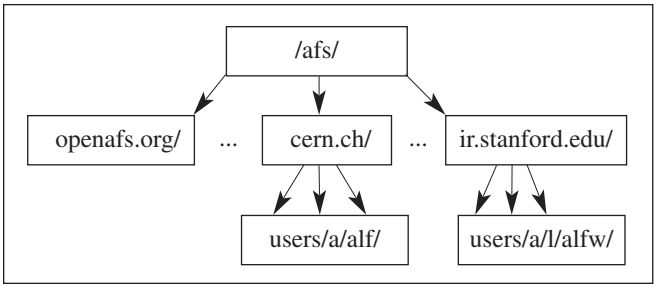


Figure 1. The AFS file space is the same anywhere and does not require clients to know which directory is on which server.

For users, the AFS file space looks like any other filesystem they have used. With the proper Kerberos credentials, they can access their AFS data from all over the world, facilitating the globally unique namespace. Here is an example: to be able to copy data from my home directory at CERN in Switzerland to my home directory at SLAC in California, I first need to authenticate myself against the two different AFS cells:

```
% kinit --afslog alfw@ir.stanford.edu
alfw@ir.stanford.edu's Password:
% kinit -c /tmp/krb5cc_5828_1 --afslog alf@cern.ch
alf@cern.ch's Password:
```

AFS comes with a command, `tokens`, to show AFS credentials:

```
% tokens Tokens held by the Cache Manager:

User's (AFS ID 388) tokens for afs@cern.ch [Expires Apr  2 10:30]
User's (AFS ID 10214) tokens for afs@ir.stanford.edu [Expires Apr  2 09:49]
--End of list--
```

Now that I am authenticated, I can access my two AFS home directories:

```
% cp /afs/cern.ch/user/a/alf/Projects/X/src/hello.c \
    /afs/ir.stanford.edu/users/a/l/alfw/Projects/Y/src/.
```

On an AFS file server, the AFS data is stored on special partitions, called /vicepXX, with XX ranging from a–zz, allowing for a total of 256 partitions per server. Each of these partitions can hold data containers called volumes. Volumes are the smallest entity that can be moved around, replicated or backed up. Volumes then contain the directories and files. Volumes need to be mounted inside the AFS file space to make them

visible. These mount points look exactly like directories.

AFS is particularly well suited to serve read-only data such as the /usr/local/ tree because AFS clients cache accessed data. To make this work even better and more robustly, AFS allows for read-only clones of data on different AFS file servers. If one server hosting such a clone goes down, the clients transparently fail-over to another server hosting another read-only copy of the same data. This replication technique also can be used to clone data across servers that are geographically far apart. Clients can be configured to prefer to use the close-by copy and use the more distant copy as a fallback. The openafs.org AFS cell, for example, is hosted on a server at Carnegie Mellon University in Pittsburgh, Pennsylvania, and on a server at the Royal Institute of Technology (KTH) in Stockholm, Sweden.

AFS provides a snapshot mechanism to provide backups. These snapshots are generated at a configurable time, say 2AM, and work on a per-volume basis. The snapshots then can be backed up to tape without interfering with user activities. They also can be provided to users by way of a simple mount point in their respective AFS home directories. This simple trick eliminates many user back-up/restore requests, because the files in last night's snapshot still are visible in this special subdirectory—the mount point to the backup volume—in users' home directories.

The AFS communication protocol was designed for wide area networking. It uses its own remote procedure call (RPC) implementation, called Rx, which works over UDP. The protocol retransmits only the single bad packet on a batch of packets, and it allows a higher number of unacknowledged packets as compared to what other protocols allow.

AFS administration can be done from any AFS client; there is no need to log on to an AFS server. This allows administrators to lock down the AFS server tightly, which is a big security plus. The location independence of AFS data also improves manageability. An AFS file server can be evacuated completely by moving all volumes to other AFS file servers. These moves are not visible to users. The empty file server then can undergo its maintenance, such as an OS upgrade or a hardware repair. Afterward, all volumes can be moved transparently back to the server.

Internally, AFS makes use of Kerberos to authenticate users. Out of the box this is Kerberos 4, but all major Kerberos 5 implementations are able to serve as a more secure substitute. AFS provides access control lists (ACLs) to restrict access to directories. Only Kerberos principals or groups of those can be put in ACLs. This is unlike NFS, in which only the UNIX user IDs are used for authorization. An additional authorization service, the protection service (PTS), is used to keep track of individual Kerberos principals and groups of principals.

AFS Components

To make all these features work, AFS comes in several distinct parts: the AFS client software that has to run on each computer that wants access to the AFS file space. The AFS server software is separated into four basic parts. It uses Kerberos for authentication, PTS for authorization, a volume location server for location independence and two servers for data serving (file server and volume server). All of these different processes are managed on each AFS server by the basic overseer (BOS) server. In addition to these necessary components, more service dæmons are available for AFS server maintenance and backup. How to install an AFS server is beyond the scope of this article.

Due to all of these different server components, the learning curve for AFS is steep at the beginning. However, the payoff is rewarding and many sites cannot go without it any longer. Once a cell is installed, the day-to-day maintenance cost for AFS is in the 25% full-time equivalent

(FTE) range, even for large installations.

For more information how AFS is used at various sites, including Morgan Stanley and Intel, have a look at the presentations given at the recent AFS Best Practices Workshop (see the on-line Resources).

AFS Client Installation

You do not need your own AFS servers to try AFS yourself. Simply installing the OpenAFS client software and starting the AFS client dæmon afsd with a special option allows users to access the publicly accessible AFS space of foreign AFS cells.

The most difficult part of installing an AFS client is obtaining the necessary kernel module. If you are using Red Hat or Fedora, you can download RPMs (see Resources). In addition to the kernel module, the AFS client needs a user-space dæmon (afsd) and the AFS command suite. These come in two additional RPMs.

Once you have these modules, the next step is to configure the AFS client for your needs. First, you need to define the cell your computer should be a member of. The AFS cell name is defined in the file /usr/vice/etc/ThisCell. If you do not have your own AFS servers, this name can be set to anything. Otherwise, it should be set to the name of the cell your AFS servers are serving. The next parameter to look at is the local AFS cache. Each AFS client should have a separate disk partition to contain the client software, but the cache can be put wherever you want. The location and size of the cache are defined in the file /usr/vice/etc/cacheinfo. The default location for the AFS cache is /usr/vice/cache, and a size of 100MB is plenty for a single user desktop or laptop computer. This is the setting as it comes with the openafs-client RPM. The cacheinfo file for this setting should look like this:

```
/afs:/usr/vice/cache:100000
```

Next, configure the parameters for afsd, the AFS client dæmon. They are defined in /etc/sysconfig/afs. Add the -dynroot parameter to the OPTIONS definition. This allows you to start the AFS client without your own AFS servers.

Another option to add is -fakestat. This parameter tells afsd to fake the stat(3) information of all entries in the /afs/ directory. Without this parameter, the AFS client would go out and contact each single AFS cell known to it. That currently is 133 cells, as seen if you do a long listing (`/bin/ls -l`) in the /afs/ directory.

Because AFS is using Kerberos for authentication, time needs to be synchronized on your machine(s). AFS used to have its own mechanism for synchronization, but it is outdated and should not be used anymore. To switch it off, the option -nosettime needs to be added to the OPTIONS definition in /etc/sysconfig/afs. If you don't have a time sync method, use Network Time Protocol (see Resources).

After all the changes have been made, the new OPTIONS definition in /etc/sysconfig/afs should look like this:

```
OPTIONS="$MEDIUM -dynroot -fakestat -nosettime"
```

The last step is to create the mount point for the AFS filesystem, which is accomplished by entering `% sudo mkdir /afs`. Now, you can start the AFS client with `% sudo /etc/init.d/afs start`. This part takes a few seconds, because afsd needs to populate the local cache directory before it can start. Because the cache is persistent over reboots, subsequent starts will be faster.

Explore AFS

Without your own AFS servers but with an AFS client configured as described above, you can familiarize yourself with some AFS commands and explore the global AFS space yourself. A quick test shows that you are not authenticated in any AFS cell:

```
% tokens
Tokens held by the Cache Manager:

--End of list--
```

No credentials are listed. See above for an example where credentials are present.

The first thing you should do is retrieve a long listing of the /afs/ directory. It shows all AFS cells known to your AFS client. Now, change into the directory /afs/openafs.org/software/openafs and do a directory listing. You should see this:

```
% ls -l
total 10
drwxrwxrwx  3 root  root    2048 Jan 7  2003  delta
drwxr-xr-x  8 100   wheel   2048 Jun 23 2001  v1.0
drwxr-xr-x  4 100   wheel   2048 Jul 19 2001  v1.1
drwxrwxr-x 17 100   101     2048 Oct 24 12:36 v1.2
drwxrwxr-x  4 100   101     2048 Nov 26 21:49 v1.3
```

Go deeper into one of these directories. For example:

```
% cd v1.2/1.2.10/binary/fedora-1.0
```

Have a look at the ACLs in this directory with:

```
% fs listacl .
Access list for . is
Normal rights:
  openafs:gatekeepers rlidwka
  system:administrators rlidwka
  system:anyuser rl
```

This shows that two groups have all seven possible privileges: read (r), lookup (l), insert (i), write (w), full file advisory lock (k) and ACL change right (a). The special group system:anyuser that comes with AFS has read (r) and lookup (l) rights, which allow access literally to anybody.

To list the members of a group, use the pts (protection server) command:

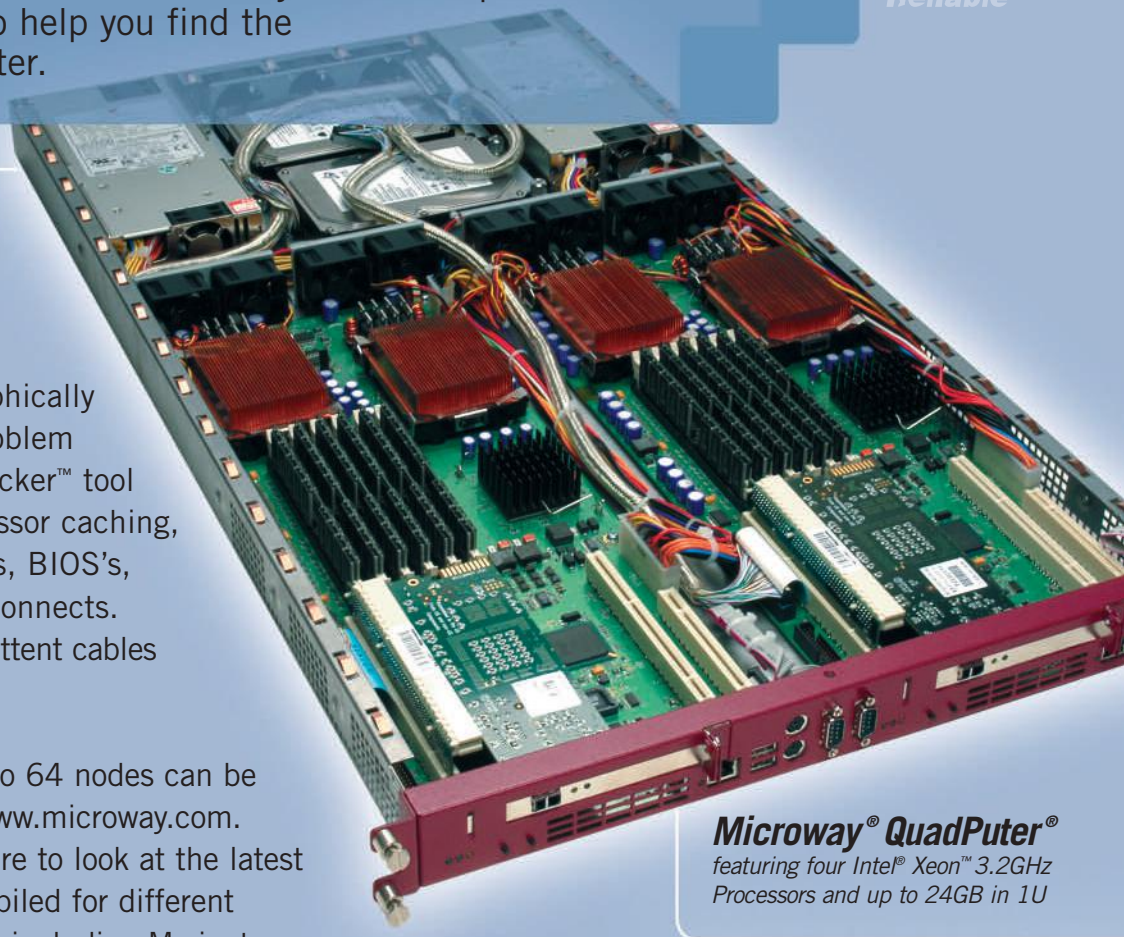
```
% pts member openafs:gatekeepers -cell openafs.org -noauth
Members of openafs:gatekeepers (id: -207) are:
  shadow
  rees
  zacheiss.admin
  jaltman
```

How good is your MPI?

The answer may surprise you.

Cool
High Density
Fast
Reliable

The performance of an MPI application is often limited by the speed of its slowest node! Microway has developed MPI diagnostic tools to help you find the weak link in your cluster.



MPI Link-Checker™ measures the bandwidth and latency between all nodes in a cluster then summarizes the data graphically so you can easily spot problem nodes. The MPI Link-Checker™ tool detects issues with processor caching, motherboards, PCI busses, BIOS's, riser cards and PCI interconnects. It can even detect intermittent cables and cross bar switches!

MPI Link-Checker™ for up to 64 nodes can be downloaded gratis from www.microway.com. When you are there, be sure to look at the latest benchmarks we have compiled for different interconnect technologies, including Myrinet, InfiniBand and Gigabit Ethernet and StarGen.

Microway® QuadPuter®
featuring four Intel® Xeon™ 3.2GHz Processors and up to 24GB in 1U

High Density Computing

Microway's XBlade™ powered QuadPuter® offers many unique features for building very dense clusters that will take a full load of DIMM modules, while interfacing to other nodes at speeds approaching a Gigabyte per second. XBlade employs a Microway-ported LinuxBIOS. It is a great building block for creating appliances and Microway has a long history of customizing hardware for COTS and OEM applications.

Please visit Microway.com and discover the very latest in parallel processing, blades and connectivity.

Microway Cluster features:

Nodewatch™
Remote monitoring and control of temperatures, fans and power, including the ability to remotely cycle and reset power via Microway's web-based GUI.

MCMS™
Microway Cluster Management Software provides OS performance data and performs common tasks like adding users, executing cluster-wide commands, and replicating hard disks.

Advanced CoolRack™ Chassis and Cabinets
Robust designs include Processor Targeted Cooling to enable 64 and 32 bit processor based clusters with power densities up to 25 KW per 44U cabinet!

Excellent Technical Support
Since 1982 Microway's Tech Team has been praised by clients for providing accurate and timely pre- and post-sale support.

Call us first at 508-746-7341 for quotations and benchmarking services.
Find technical information, testimonials, and online newsletter at www.microway.com

Microway
Technology you can count on™

The -noauth option is used because this command is run without any credentials for this cell.

Special administrative privileges are necessary to explore the authentication part of AFS, which is standard Kerberos, so I skip it here.

Now, find out where the current directory physically is located:

```
% fs whereis .

File . is on hosts andrew.e.kth.se VIRTUE.OPENAFS.ORG
```

This shows that two copies of this directory are available, one from andrew.e.kth.se and one from VIRTUE.OPENAFS.ORG.

The command:

```
% fs lsmount /afs/openafs.org/software/openafs
➥/v1.2/1.2.10/binary/fedora-1.0
/afs/openafs.org/software/openafs/v1.2/1.2.10/binary/fedo
ra-1.0
➥ is a mount point for volume #openafs.1210.f10
```

shows that this directory actually is a mount point for an AFS volume named openafs.1210.f10.

Another AFS command allows us to inspect volumes:

```
% vos examine openafs.1210.f10 -cell openafs.org -noauth
```

This command examines the read-write version of volume openafs.1210.f10 in AFS cell openafs.org. The output should look like this:

```
openafs.1210.f10       536871770 RW    25680 K On-line
    VIRTUE.OPENAFS.ORG /vicepb
    RWrite  536871770 ROnly  536871771 Backup      0
    MaxQuota           0 K
    Creation    Fri Nov 21 17:56:28 2003
    Last Update Fri Nov 21 18:05:30 2003
    0 accesses in the past day (i.e., vnode references)

    RWrite: 536871770     ROnly: 536871771
    number of sites -> 3
       server VIRTUE.OPENAFS.ORG partition /vicepb RW Site
       server VIRTUE.OPENAFS.ORG partition /vicepb RO Site
       server andrew.e.kth.se partition /vicepb RO Site
```

The output shows that this volume is hosted on server VIRTUE.OPENAFS.ORG in disk partition /vicepb. The next line shows the numeric volume IDs for the read-write and the read-only volumes. It also shows some statistics. The last three lines show where the one read-write (RW Site) and the two read-only (RO Site) copies of this volume are located.

To find out how many other AFS disk partitions are on the server VIRTUE.OPENAFS.ORG, use the command:

```
% vos listpart VIRTUE.OPENAFS.ORG -noauth
```

We learn that the partitions on the server are:

```
/vicepa      /vicepb      /vicepc
Total: 3
```

which show a total of three /vicep partitions. To see what volumes are located in partition /vicepa on this server, execute:

```
% vos listvol VIRTUE.OPENAFS.ORG /vicepa -noauth
```

This command takes a while and eventually returns a list of 275 volumes. The first few lines of output look like this:

```
Total number of volumes on server VIRTUE.OPENAFS.ORG partition /vicepa: 275
openafs.10.src                    536870975 RW      11407 K On-line
openafs.10.src.backup             536870977 BK      11407 K On-line
openafs.10.src.readonly           536870976 RO      11407 K On-line
openafs.101.src                   536870972 RW      11442 K On-line
openafs.101.src.backup            536870974 BK      11442 K On-line
openafs.101.src.readonly          536870973 RO      11442 K On-line
```

Another command, bos, communicates with a cell's basic overseer server and finds out the status of that cell's AFS server processes. Many more subcommands are available for the fs, pts, vos and bos commands. All of these AFS commands understand the help option (no dash in front of help) to show all available subcommands. Use `fs <subcommand> -help` (with the dash) to look at the syntax for a specific subcommand.

The Future of AFS

Several enhancement projects for AFS currently are underway. The most important project right now is to make AFS work with the 2.6 Linux kernels. These kernels no longer export their syscall table. Another project is to provide a disconnected mode that allows AFS clients to go off the network and continue to use AFS. Once they reconnect, the content of files in AFS space is re-synchronized.

Conclusion

Although all the different aspects of AFS can be overwhelming at first and the learning curve for setting up your own AFS cell is steep, the reward for using AFS in your infrastructure can be significant. Secure, platform-independent world-wide file sharing is a concept as attractive as serving your /usr/local/ area and all your UNIX home directories. And, all this comes with only minimal long-term administrative costs.

Resources for this article: www.linuxjournal.com/article/8079.

Alf Wachsmann, PhD, has been at the Stanford Linear Accelerator Center (SLAC) since 1999. He is responsible for all areas of automated Linux installation, including farm nodes, servers and desktops. His work focuses on AFS support, migration to Kerberos 5, a user registry project and user consultants.

Open Access for Science

When a university professor writes a journal article for no pay, and the university library can't afford the journal, something is wrong. Open access is bringing reform to scientific publishing.

BY CHRISTOPHER M. FRENZ

The Scientific community, especially in the area of bioinformatics, always has been a strong proponent of open-source technologies. Linux and related technologies, such as the Perl programming language, rapidly are becoming the de facto standards for conducting computational research in the biological sciences. In fact, openness and information sharing are some of the most fundamental tenets of the scientific world. Scientific progress is based on the ideal that information uncovered by one group should benefit the research and development efforts of other groups as well.

Information sharing is promulgated through the publication of scientific research in peer-reviewed journals. However, there is one kink in this system. Most scholarly journals do not pay authors, and many actually impose page charges on scientists who contribute. Journal editorial boards also are typically made up of scientists who serve without pay. Yet, despite the fact that scholarly journals have little or no costs for articles and editorial direction, many of them require the payment of prohibitively expensive subscription fees before researchers from an institution are able to access the research contained inside these journals.

The growth of the open-source revolution within the bioinformatics world is causing a reevaluation of this publishing model, however. Perhaps, in essence, these two campaigns are really striving for the same goal. The open-source software revolution seeks to promote freedom among software users, so they have the freedom to use the software in any way they see fit. Among these freedoms is the ability to examine the source code and study the inner workings of the software in order to learn how it operates. Users then are free to modify this source code and adapt it as they see fit. Users then can make these improvements available to the rest of the world.

Within the Open Source community, this is how software evolves. Someone has an idea and releases a program that makes this idea possible. Others then are able to take this functionality and apply it to new problem sets, perhaps even ones the original author never thought possible. As scientists, we currently are seeking the same kind of freedom for our research results.

The scientific world actually has made some strides in this direction, with several upstart publications, such as the Public Library of Science (PLOS), making all of the articles they publish open access. However, many established publications still insist that they must continue to charge high fees for subscriptions and for on-line access to archives in order to turn a profit.

The momentum of the open access movement is picking up, however, and has led to the development of a promising proposal by the National Institutes of Health (NIH), an organization that funds much of the biomedical research within the United States. According to this new proposal, publishers can keep exclusive subscription-based access to publications that result from NIH-funded research for a period of only six months. After that time, the papers must be made available to the public in an electronic format that has been archived in a scientific literature repository such as Pubmed. If this proposal passes, it will be a major stride toward achieving open access for research articles in the biological sciences. It is expected that many publishers will adapt their ways rather than risk losing large numbers of articles that help their journals sell in the first place.

As Linux and open-source enthusiasts, this issue holds more for us, however, than simply the freedom to access scientific discoveries. It represents a change toward openness in an environment that in many ways was trying to become more closed. We all have witnessed the development of copy restriction methods on audio files, videos, e-books and proprietary software. Industry groups such as the Recording Industry Association of America (RIAA) also seek to add additional restrictions to the way we can utilize our electronic media and software.

As Linux and open-source enthusiasts, this issue holds more for us, however, than simply the freedom to access scientific discoveries.

This current publication movement is a step in the other direction—a step toward promoting the same types of freedom and openness that we seek when we turn to an open-source solution. The open access issue is not only significant for the scientific advances it may help unleash, but also because it provides the Open Source community with an alternative means to enlighten society about the virtues of freedom and openness. This, in turn, may garner more support for the open-source cause. Thus, in the spirit of freedom and openness, we should rally behind this issue and demand the right to access openly the research that our tax and tuition dollars are supporting.

Christopher M. Frenz is a bioinformaticist with more than five years of experience using Linux. He also is the author of the book *Visual Basic and Visual Basic .NET for Scientists and Engineers* (Apress) and currently is writing a book about Perl programming.

routerOS: routerBOARD.com

User Management

- support more than 3000 PPPoE or Hotspot clients
  - full radius support for user parameters
    - tx/rx speed, address, filter rules
    - supports radius real time modification of parameters while users are online
- Peer to peer control (P2P)
  - burst time
  - per client P2P tx/rx rules
  - P2P pool
  - complete blocking of P2P

Wireless AP and Backbone

- Wireless monitoring
  - Frequency scanning with detailed report
  - Raw wireless packet sniffer
    - streaming option to Ethereal analyzer
    - option to save to a file format supported by Ethereal
  - Snooper packet inspection
    - analyzes all raw frames received for wireless parameters
    - monitor a single channel or all channels

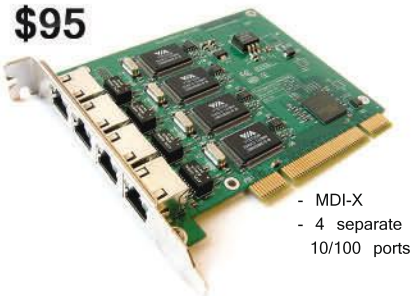
- Nstreme wireless polling protocol
  - no decrease in speed over long distances (as seen with the 802.11 ack packet bottleneck)
  - polling improves speed and eliminates contention for access to the wireless bandwidth
  - access point control over Nstreme clients tx data to optimize use of the wireless medium
  - radius support for the access control list including bandwidth settings for wireless clients

- Full 802.11a/b/g support

The above is a brief description of a few features, for more information and a fully featured 24 hour demo go to:

www.mikrotik.com

$95



- MDI-X
- 4 separate 10/100 ports

RouterBOARD 44

For the Router Builder !

- rackmount servers and routers
- up to 24 Ethernet ports in a PC
- no more straight/cross cable problems
- server quality VIA VT6105 chips

$195

RouterBOARD 230

No feature left behind !

Integrated router with various interfaces. Use as an AP on a tower with up to 500ft PoE. Includes IDE/CF, miniPCI, USB, PCMCIA, UART, PCI, GPIO, LCD controller, Linux SDK, and more.

$120 **$65**

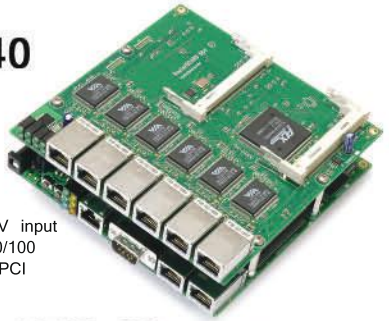
Eight ports Four ports

RouterBOARD 11/14/18

Multi radio tower !

MiniPCI to PCI adapters for multi radio system. Tested with sixteen radios in one Router/AP.

$240



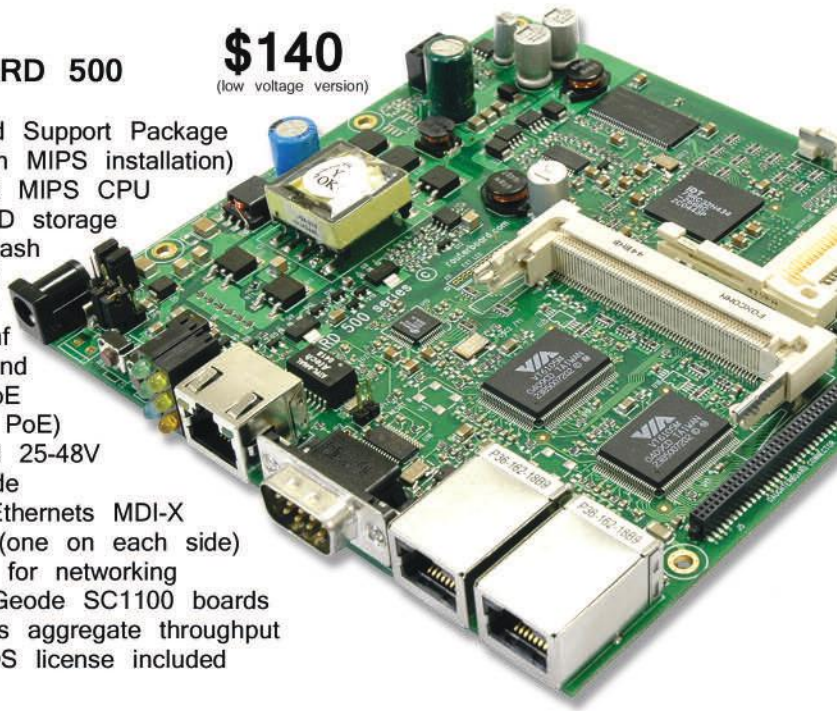
- PoE
- 10-56V input
- 9x 10/100
- 6x mPCI

RouterBOARD 500 & RouterBOARD 564

The Wireless Switchboard !

For a complete multi-radio tower system, the RouterBOARD 500 can carry a daughterboard (RouterBOARD 564) which adds six ethernets and four miniPCI.

RouterBOARD 500 **$140** (low voltage version)



- Linux Board Support Package (full Debian MIPS installation)
- 266-400Mhz MIPS CPU
- 64MB NAND storage
- Compact Flash
- 32MB DDR
- Low power
- PoE 802.3af standard and passive PoE (also 12V PoE)
- 10-24V and 25-48V power mode
- 3 10/100 Ethernets MDI-X
- 2 miniPCI (one on each side)
- 2-3x faster for networking than the Geode SC1100 boards
- 200-300MB/s aggregate throughput
- L3 RouterOS license included

contact sales@routerboard.com or go to **www.routerboard.com**

64-bit Enterprise Servers and HPC Clusters!

Opteron™, Xeon™ EM64T, and Itanium®2

Innovative Technology and Services at Competitive Prices

- **NEW! Microway Quadputer®-Navion™ enterprise-class server** incorporates 4 Opteron 850 processors in 4U chassis with hot-swap, redundant power supplies and hard drives. For other 64-bit solutions, go to www.microway.com.
- **NodeWatch™/MCMS™** provides remote control and monitoring of vital cluster parameters and failsafe shutdown. NodeWatch monitors temperatures, voltages, and chassis fans, runs off the master node, and is controlled by a secure web-based GUI. Available only on Microway HPC solutions.
- **Fully redundant, highly-available storage systems** based on fiber channel technology for multi-terabyte storage requirements. State-of-the-art storage directors for full access from any cluster node. Easily scales to address rapidly expanding storage requirements.

Expert Integration...

Microway offers innovative, competitively priced, custom designed clusters with NodeWatch™ management and monitoring tools. Our solutions incorporate the latest processors, proprietary cooling and storage solutions, plus high-speed Myrinet and InfiniBand interconnects for demanding applications.

Superior Service and Tech Support...

We understand that on-time delivery, out-of-the-box reliability and excellent ongoing technical support are critical to our users. Microway offers professional services from specialists with a wide range of expertise in HPC applications. On-site installations and training are also available.

Satisfied Customers...

AT&T, Cessna, GE, GSK, Johnson & Johnson, LANL, LLNL, MBL, Millennium Pharmaceuticals, NIH, Northrop Grumman, Raytheon, Sandia, Seagate, US Air Force, Army, Navy, NASA, NOAA and hundreds of leading universities are among our satisfied customers since 1982.

Microway CoolRak™ Cabinet with dual Opteron or Xeon 1U nodes, Myrinet connectivity and four 10" 535 CFM rear fans.

Quadputer®-Navion™ with four AMD Opteron 850s plus hot-swap, redundant power supplies and hard drives.

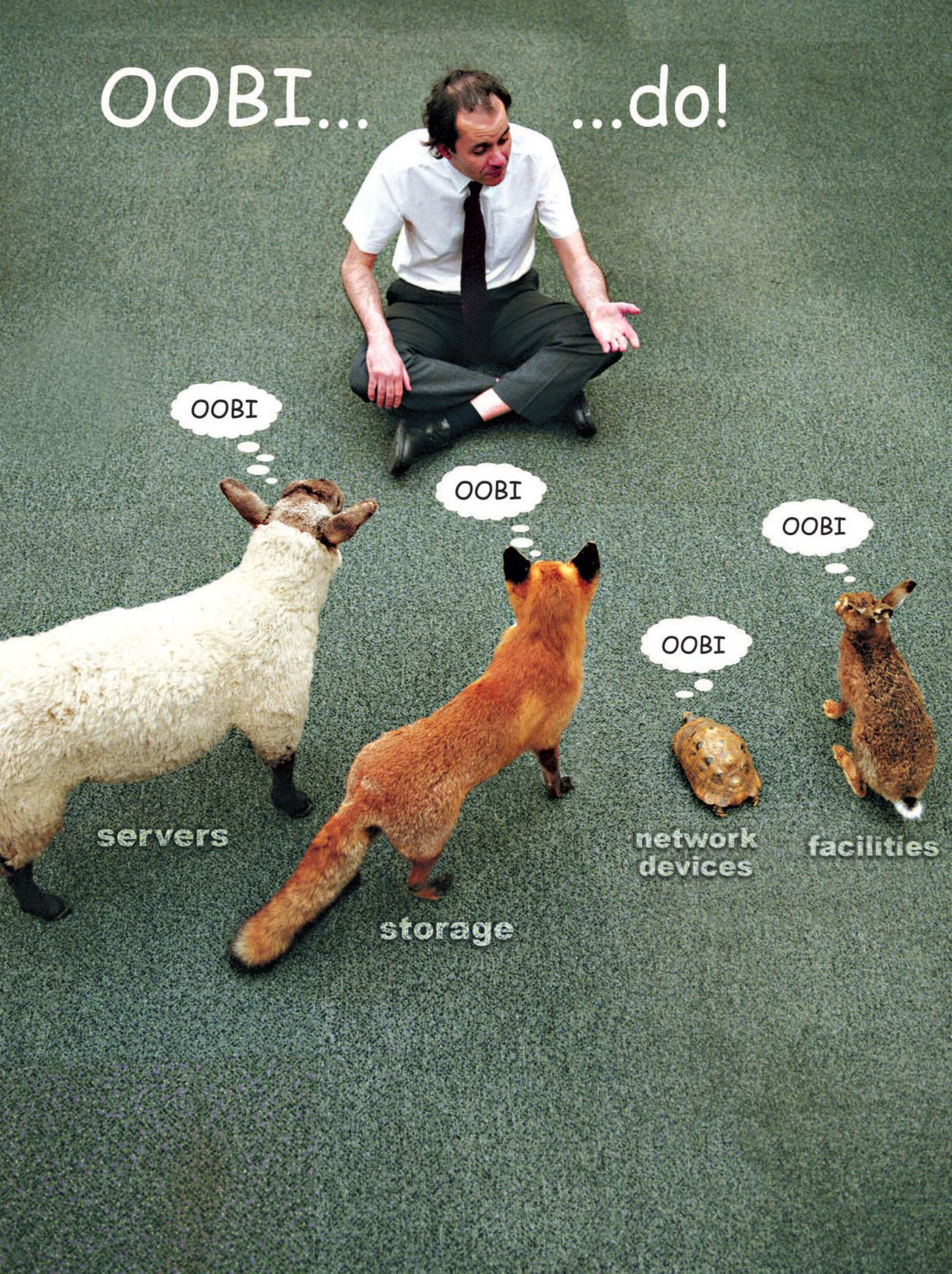
"The Brain Imaging Research Center (a joint center of Carnegie Mellon University and University of Pittsburgh) decided to purchase our Linux cluster from Microway because of the proven performance of their clusters at Carnegie Mellon in the processing of high-volume brain imaging data. Microway was flexible and helpful at all stages, starting from the initial custom configuration and ending with timely delivery and full installation."

— Marcel Just, Co-Director, Brain Imaging Research Center

*Call us first at **508-746-7341** for quotations and benchmarking services. Find technical information, testimonials, and online newsletter at **www.microway.com**.*

Microway®
Technology you can count on℠

AMD 64 Opteron™ · intel inside ITANIUM 2 · intel inside XEON


OOBI... ...do!
OOBI
OOBI
OOBI
OOBI
servers
storage
network devices
facilities