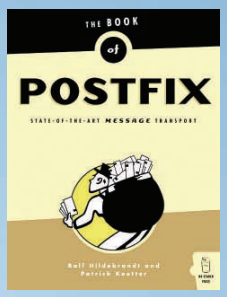
AN SSC PUBLICATION

Take the money and exec(): bill reader driver

LINUX JOURNAL™

Since 1994: The Original Magazine of the Linux Community
OCTOBER 2005

Fight spam and win with THE BOOK OF POSTFIX



PLUG IT IN AND IT WORKS

Project Utopia makes desktop hardware play nice

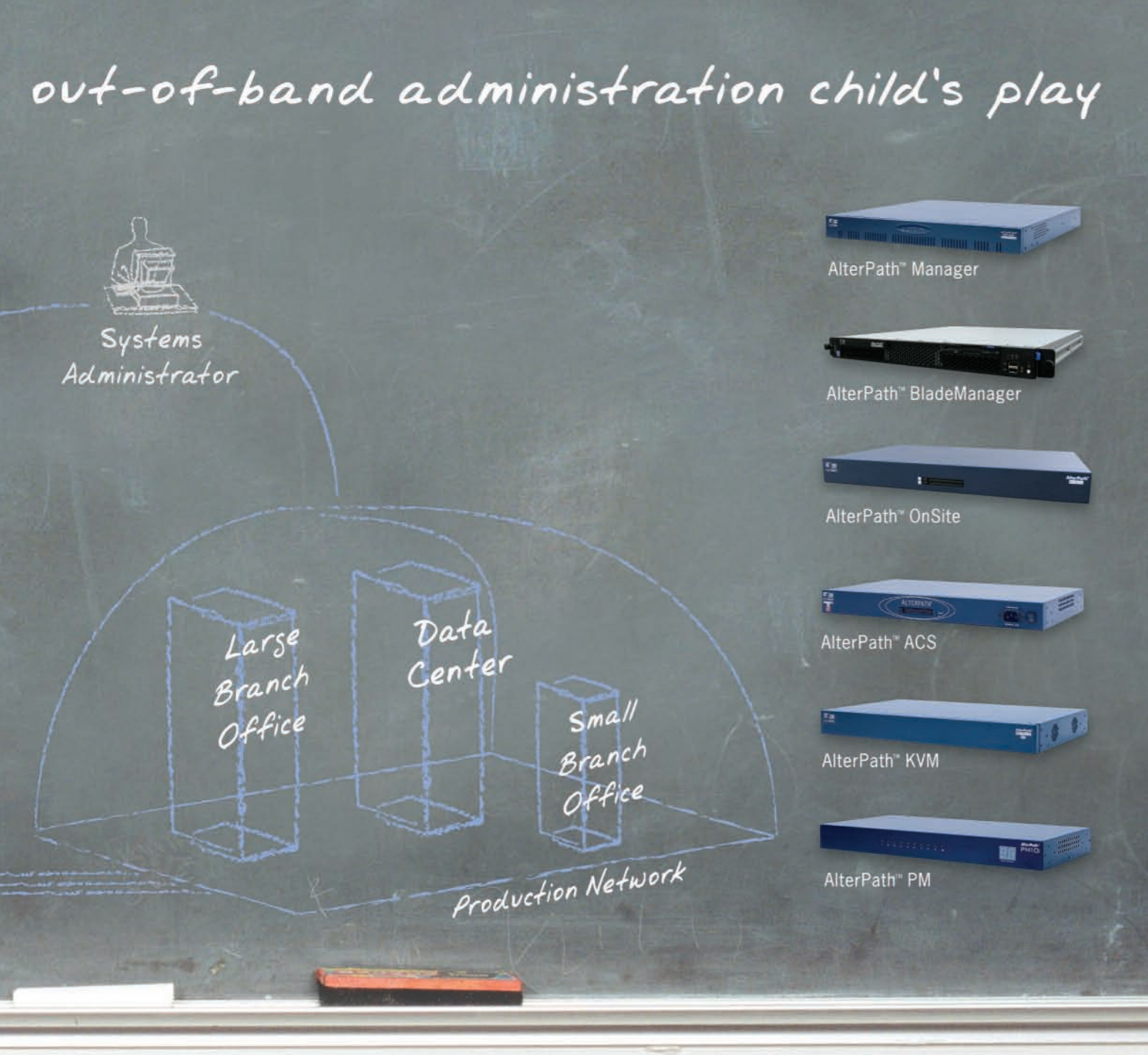
WHERE'S MY FILE?

Comparing file search tools

+ Managing your TODOs, from Evolution to Emacs
An in-kernel FTP client
Decrypting shc-encrypted shell scripts
An 8-speaker, 3-D sound system
Fixing other people's Web sites with GreaseMonkey

USA $5.00 CAN $6.50
www.linuxjournal.com
10
0 71486 03102 4

LINUX JOURNAL · PERSONAL DESKTOP · GreaseMonkey · Project Utopia · Ruby on Rails · 3-D Audio · shc · OCTOBER 2005 ISSUE 138



The Next-Generation IT Infrastructure

Cyclades AlterPath™ System is the industry's most comprehensive Out-of-Band Infrastructure (OOBI) system. The AlterPath System allows remote data center administration, eliminating the need for most time-consuming, remedial site visits. When fully deployed in your data center, Cyclades AlterPath System lowers the risks associated with outages, improves productivity and operational efficiency, and cuts costs.

Each component of the AlterPath System is designed to seamlessly integrate into the enterprise, able to scale in any direction. Whether you need serial console management of networking equipment, KVM for access to Windows® servers, branch management, IPMI or HP iLO for service processor management or advanced power management, the AlterPath System delivers. Cyclades brings it all together, making OOBI administration seem like child's play.

Over 85% of Fortune 100 choose Cyclades.

www.cyclades.com/lja

1.888.cyclades • sales@cyclades.com

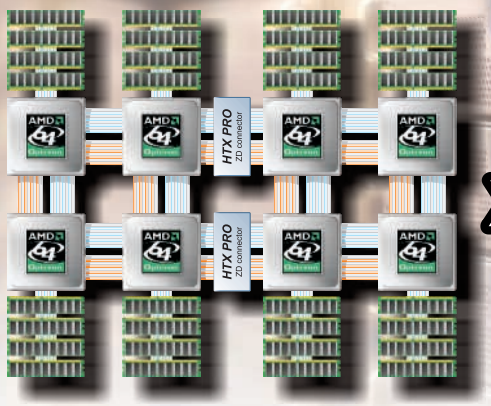
cyclades®

©2005 Cyclades Corporation. All rights reserved. All other trademarks and product images are property of their respective owners. Product information subject to change without notice.

8 Opteron sockets
IWILL H8501 Barebone System

AMD 64 Opteron™

High Performance Computing
Complex Executing
DataCenter
SuperCluster



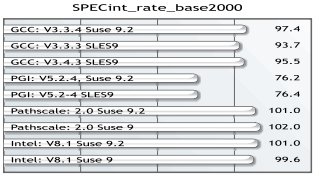
X2-core = 16-Way

IWILL H8501 ▶

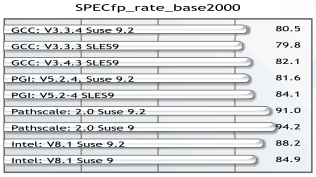
- 8x AMD Opteron Processor 940 sockets
- Supports 800 series Opteron CPUs with dual core tech.
- Up to 128GB DDR Registered ECC memory
- Support 4 Ranks memory module
- 1350W Redundant PSU 3+1
- Support IPMI server management
- Industry 19" rack-mountable 5U chassis
- 4 x Gigabit Ethernet ports, and 4 PCI-X slots
- Up to 10 hot-swap HDDs with option HDD canister
- Modularization design, I/O may vary

8-Way AMD Opteron Server Benchmark Rating
SPEC CPU2000: COMPILERS COMPARED

SPECint_rate_base2000



SPECfp_rate_base2000



SERVER & WORKSTATION
Opteron Dual Core Ready

Highest Performance
Accommodate 128GB RAM
8-WAY/ 16-CORE

IWILL Other Outstanding Motherboards & Small Form Factor:

▼ DK8-HTX

HTX TECHNOLOGY

HTX allows access into the AMD64 Direct Connect Architecture through a variety of HTX connectors and slots, integrating outside devices directly resident on the HyperTransport technology bus.

- 2 AMD Opteron Processor 940 sockets
- Supports 2xx Opteron CPUs with dual core tech.
- Up to 16GB DDR Registered ECC memory
- 1 x PCI-X 64bit 133/100/66MHz expansion slot
- 2 x PCI 64bit 100/66MHz expansion slots
- HTX-Pro support Pathscale InfiniPath HTX Adapter
- 2 x Gigabit Ethernet ports (Intel chipset)
- AGP 8MB on board & support IPMI

▼ DK8ES

- 2 AMD Opteron Processor 940 sockets
- Supports 2xx Opteron CPUs with dual core tech.
- Up to 16GB DDR Registered ECC memory
- 3 x PCI-X 64bit 133/100/66MHz expansion slots
- 2 x PCI-Express x16 expansion slots
  (one in PCI-Express x2 mode)
- 2 x Gigabit Ethernet ports (Broadcom chipset)
- AGP 8MB on board & support IPMI

▼ DK88

- 2 AMD Opteron Processor 940 sockets
- Supports 2xx Opteron CPUs with dual core tech.
- Up to **64**GB DDR Registered ECC memory
- 2 x PCI-Express x8 expansion slots
- 1 x PCI-X 64bit 133/100/66MHz expansion slot
- 1 x PCI-X 64bit 100/66MHz expansion slots
- 2 x Gigabit Ethernet ports (Broadcom chipset)
- AGP 8MB on board & support IPMI

▼ ZMAX-DP/ ZMAX-D2

Dual processors Small Form Factor

- 2 AMD Opteron Processor 940 sockets
- Supports 2xx Opteron CPUs with dual core tech.
- Up to 4GB DDR Registered ECC memory
- 1x AGP 8X, 1x PCI and 1x mini PCI slot (ZMAX-DP)
- 2x PCI-Express support SLI Tech.(ZMAX-D2)
- 3x 3.5" HDD bays, and 1x 5.25" CD-ROM bay
- 1x GbE, 1x IEEE1394, 8x USB 2.0 ports
- 300W Power supply

IWILL USA Corp.
9004 Research Drive
Irvine, CA 92618
Tel: +1 949 753-5488
Fax: +1 949 753-5499

Visit www.iwill.net for more information.
Or contact us: sales@iwillusa.com, oem@iwillusa.com

IWILL reserves the right to change specifications or other product information without notice. This publication could include technical inaccuracies or photographical errors. IWILL provides this publication as is without warranty of any kind, either express or implied, including the implied warranties of merchantability or fitness for a particular purpose. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions; therefore, this disclaimer may not apply to you.

Iwill®

LINUX JOURNAL
OCTOBER 2005 ISSUE 138

COVER STORY

66 PROJECT UTOPIA
Linux's long-standing tradition of isolating the user from the hardware might be great for security, but it can be a real pain when you just want to snag some photos from your camera or check for wireless access points you're allowed to use. But there is a plan. This month, Robert Love covers Project Utopia.

FEATURES

52 FIXING WEB SITES WITH GREASEMONKEY
This Web site is fine, but it could really use....Redesign other people's Web sites to your liking, on the fly.
NIGEL MCFARLANE

60 THE LINUX FOR KIDS EXPERIMENT
Can a Linux dad get his family moved to a secure, easy-administration box without giving up the fun and education?
PAUL BARRY

66 PROJECT UTOPIA
Traditionally, Linux has protected the hardware from the user for security. When apps need to understand the hardware, new modes of communication are arising.
ROBERT LOVE

INDEPTH

72 BUILDING A CALL CENTER WITH LTSP AND SOFT PHONES
You don't need to put a phone and a computer at every desk. Use a soft phone on an almost-thin client.
MICHAEL GEORGE

78 DIRT CHEAP 3-D SPATIAL AUDIO
Look out! Bogey at 10 o'clock high! Your next simulator project can have realistic sound above, below and on all sides of the user.
ERIC KLEIN, GREG S. SCHMIDT, ERIK B. TOMLIN AND DENNIS G. BROWN

88 TAMING THE TODO
Is your computer helping you get work done, or making more work for you? Try these software options to get your act together.
SACHA CHUA

COVER PHOTO:
BETHANY PASEMAN

EMBEDDED

42 DEVELOPMENT OF A USER-SPACE APPLICATION FOR AN HID DEVICE, USING LIBHD
We won't show you the money, but we'll show you the code for the device that shows you the money.
EOIN VERLING

TOOLBOX

14 AT THE FORGE
Ruby on Rails
REUVEN M. LERNER

22 KERNEL KORNER
Network Programming in the Kernel
PRADEEP PADALA AND RAVI PARIMI

34 COOKING WITH LINUX
Trekking through the Desktop Jungle
MARCEL GAGNÉ

38 PARANOID PENGUIN
Limitations of shc, a Shell Encryption Utility
NALNEESH GUAR

COLUMNS

47 LINUX FOR SUITS
The Only Silo
DOC SEARLS

96 EOF
The Universal Internet Time Source
ADRIAN VON BIDDER

REVIEW

71 THE BOOK OF POSTFIX
DON MARTI

DEPARTMENTS

4 FROM THE EDITOR
6 LETTERS
10 UPFRONT
70 NEW PRODUCTS
81 ADVERTISERS INDEX
95 MARKETPLACE

NetworkManager gets notifications of new network hardware and available access points, so all you have to do is find a hotspot (page 66).

NEXT MONTH

HACK ANYTHING

Internet radio doesn't have to tie you to the computer. Dan Rasmussen, Jon Morgan and Paul D. Norton have updated a classic radio design with the ability to tune in Internet streams.

Stuart Brorson covers the electronic design automation tools needed to work with schematics and crank out professional-looking board designs that you can build yourself or order from a PCB house.

If you think your favorite pinball machine is complicated now, try interfacing it to a Linux box. John Bork covers digital I/O techniques to help you connect to useful devices such as solenoids and switches.

rm -rf /opt/bs

A Linux desktop shouldn't be a kick in the teeth.

BY DON MARTI

To understand the IT industry, start with *On Bullshit* by Harry G. Frankfurt. Prof. Frankfurt poses, but doesn't answer, the question of why there is so much B.S. in our society. He compares his subject to shoddy construction, and that's an analogy we can work with, because in software we're working at the thrilling edge of language and craftsmanship. We have the tools for dealing with B.S. in computer languages. Try to B.S. a compiler and that's a bug. It's time to tackle the B.S. problem head-on and start reporting bugs in human communications too.

Consider this filler, I mean essay, to be a bug report on the big companies that are doing Linux for the desktop. "Let's 'position' Linux as a simplified desktop for 'transactional users'", they say. That's right—employees, if your company gives you Linux, that means Management thinks you're a human servlet. Decision-makers and content creators get a proprietary desktop OS.

Of course, offending the employees' pride might not show up on a TCO spreadsheet. But no executive would want to admit to running a division full of transactional, replaceable, outsourceable "human resources".

But what about Clayton Christensen, disruptive innovation and *The Innovator's Dilemma*? Doesn't the cheap, good-enough contender always grow the features and stability it needs to win? Yes, when it lets in the customers left pressing their noses against the Expensive Stuff Store window. In the 1980s Macintosh let you do layouts even if you couldn't afford phototypesetting. In the 1990s Linux let you put up a Web server without blowing the price of a Coupe de Ville on a UNIX box.

But selling less-capable products to customers who can get the good stuff doesn't fly. Seen an F-20 at an air show lately? It was a capable airplane, but it was positioned as an "export fighter" for air forces that weren't allowed to have, or couldn't afford, the F-16. Naturally, countries held out for the "real" fighter. Information freedom ideals can go only so far when vendors patronize Linux customers. "Aww, the little transaction worker filled out a Web form! Isn't that cute?"

Desktop Linux marketing is doing more harm than good, but work is under way to make Linux out-perform the other OSes. Robert Love's Project Utopia is bringing together the desktop interface and the necessary tweaking of hardware to make things work smoothly, not just securely (page 66).

Michael George has an example of how a thin-client environment almost works to solve a problem, but the project needed one key local app, the soft phone. See a hybrid approach to a VoIP station that works as a phone and a PC on page 72.

One of the projects where software excellence, not transaction-workerism, has triumphed, is Mozilla Firefox. Mozilla expert and author Nigel McFarlane died last month, leaving us with one last article (page 52). Let Firefox serve as an example for the standards the desktop is coming to meet because all B.S. aside, it has to.

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

LINUX JOURNAL

OCTOBER 2005
ISSUE 138

EDITOR IN CHIEF Don Marti, ljeditor@ssc.com

EXECUTIVE EDITOR Jill Franklin, jill@ssc.com

SENIOR EDITOR Doc Searls, doc@ssc.com

SENIOR EDITOR Heather Mead, heather@ssc.com

ART DIRECTOR Garrick Antikajian, garrick@ssc.com

TECHNICAL EDITOR Michael Baxter, mab@cruzio.com

SENIOR COLUMNIST Reuven Lerner, reuven@lerner.co.il

CHEF FRANÇAIS Marcel Gagné, mggagne@salmar.com

SECURITY EDITOR Mick Bauer, mick@visi.com

CONTRIBUTING EDITORS

David A. Bandel • Greg Kroah-Hartman • Ibrahim Haddad • Robert Love • Zack Brown • Dave Phillips • Marco Fioretti • Ludovic Marcotte • Paul Barry • Paul McKenney

PROOFREADER Geri Gale

VP OF SALES AND MARKETING Carlie Fairchild, carlie@ssc.com

MARKETING MANAGER Rebecca Cassity, rebecca@ssc.com

INTERNATIONAL MARKET ANALYST James Gray, jgray@ssc.com

REGIONAL ADVERTISING SALES

NORTHERN USA: Joseph Krack, +1 866-423-7722 (toll-free)

EASTERN USA: Martin Seto, +1 905-947-8846

SOUTHERN USA: Laura Whiteman, +1 206-782-7733 x119

INTERNATIONAL: Annie Tiemann, +1 866-965-6646 (toll-free)

ADVERTISING INQUIRIES ads@ssc.com

PUBLISHER Phil Hughes, phil@ssc.com

ACCOUNTANT Candy Beauchamp, acct@ssc.com

***LINUX JOURNAL* IS PUBLISHED BY, AND IS A REGISTERED TRADE NAME OF, SSC PUBLISHING, LTD.**

PO Box 55549, Seattle, WA 98155-0549 USA • linux@ssc.com

EDITORIAL ADVISORY BOARD

Daniel Frye, Director, IBM Linux Technology Center

Jon "maddog" Hall, President, Linux International

Lawrence Lessig, Professor of Law, Stanford University

Ransom Love, Director of Strategic Relationships, Family and Church History Department, Church of Jesus Christ of Latter-day Saints

Sam Ockman, CEO, Penguin Computing

Bruce Perens

Bdale Garbee, Linux CTO, HP

Danese Cooper, Open Source Diva, Intel Corporation

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.

LINUX is a registered trademark of Linus Torvalds.

The Power of Choice...

Command the game with your next I/O move.

Modularity. Scalability. Reliability. Cost-effectiveness. These represent the solid foundations that SBE delivers to OEMs for building innovative end solutions. Partnering with SBE for networking and communications I/O solutions allows you to take advantage of proven technology and field-tested products designed to optimize performance for your unique application needs.

SBE offers a full spectrum of interface cards, ranging from T1 and T3 to Gigabit Ethernet and IPsec/SSL acceleration. These boards are available in multiple form factors, including PCI, PMC, and PTMC. Customers have the choice of buying these boards individually or bundling any of the PMC/PTMC modules with our intelligent core processing platforms to create a flexible, cost-efficient blade solution ideal for serving demanding telecom applications. Full Linux support is available on every board.

- Channelized T3
- 24-port T1/E1
- LAN/Ethernet
- Storage
- IPsec/SSL Encryption
- Blade platforms
- I/O and beyond...

SBE®

Linux On Demand

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

Ultimate Linux Box Cooling?

I was impressed by the desire to make a quiet PC—more people should complain to their OEMs/System integrators about this—it is the only way it will be fixed ["Ultimate Linux Box", August 2005]. But I am not sure that removing 100% of the airflow is a positive thing for overall system performance or stability.

I have seen motherboards designed where the processor power supply components can exceed the design rating from the suppliers without airflow. Even if things don't go bad enough to cause system stability issues, it can damage the processor by allowing the CPU voltage to go out of specification. Intel is so concerned about this—they are telling motherboard manufacturers to add circuits to the motherboard to monitor the temperature of the processor power supply and modulate the clock of the processor if things get too hot. See Section 9.4 in download.intel.com/design/Pentium4/guides/30235604.pdf.

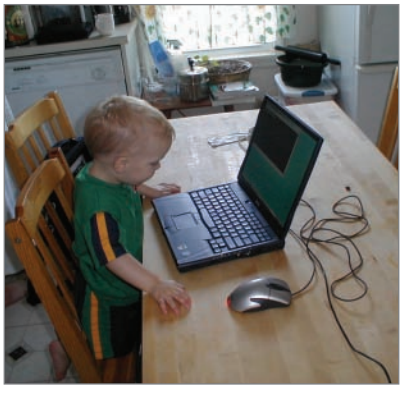
Thanks—and looking forward to more sub-minute kernel compiles.

--
Robin

Whenever you experiment with any alternate cooling method, always measure and log temperatures.—Ed.

Linux/BSD Confusion

My child is almost two in the picture. The laptop is running KDE on FreeBSD. He quite plainly CALLED it "Linux". My wife and I both looked at each other and at him, and he said it again. He has also said "Ethernet". We are afraid, very afraid. Thanks for a (decade of a) great publication!



--
kurtseel

More on the ULB

I'm used to reading the annual Ultimate PC article in *Maximum PC* magazine each year, and they give a great deal of detail and many more pictures than the Ultimate Linux Box article [August 2005]. However, the big difference between the two articles is that your Ultimate Linux Box does much more customization.

The big question is, if I want to replicate the Ultimate Linux Box, where would I get details on the custom-made power supply cooler modifications?

I went to the Resources page, but many of the things I would need to access require that I'm a *Linux Journal* subscriber. I have subscribed to the *Linux Journal* in the past, but for the last year, I simply run out and purchase *Linux Journal*.

--
Dean

Articles from that issue will be openly available soon. We'll look for more info on the power supply mods.—Ed.

Ten Years of Progress

The shock has worn off from seeing my name in *Linux Journal*, and I'm able to write again. [See "Ten Years Ago in *Linux Journal*", July 2005, page 14.] Has it been *ten* years since "Novice to Novice" appeared in *Linux Journal*? It must be. I stopped writing the column after my first child was born and—bless it!—if it's not his tenth birthday already.

And how much has Linux changed (and stayed the same) in ten years. Though I haven't quite made the switch away from Microsoft, I did recently install Fedora Core. In ten years, installation has vastly improved since when I used version 0.99 of Slackware. The Fedora installation was relatively fast and idiot-proof. Everything worked except the modem, and although I'm finally getting DSL installed, I wanted modem access as backup. Turns out I have a PCTel modem, which seems unsupported by the 2.6+ kernel and by the drivers currently out there in Webland.

(Hmmm, could be another "Novice to Novice" here.)

But what's blown me away about Linux are

the live CDs. Knoppix and the variations are fantastic not only for emergencies but also just to learn *nix, which is why I got involved with Linux originally.

Ah well! Thanks for remembering me after all these years. Yes, I still have the 0.99 Slackware CD with the grinning Bob. It just seems right to keep it.

--
Dean

Become Boring and Pigeonholed, Please

Hi. I've been a subscriber to *LJ* for several years, and I've never figured out your niche. Servers, or desktops? Sysadmins, application programmers, system developers or home users? Your intention seems to be: appeal to everyone. Unfortunately for me, you cover so many different topics that there is very little in each issue for me.

If you can't figure out your niche, I'll let my subscription expire in January.

--
jh

When different areas of information technology can stop learning from each other, we'll pick a "niche".—Ed.

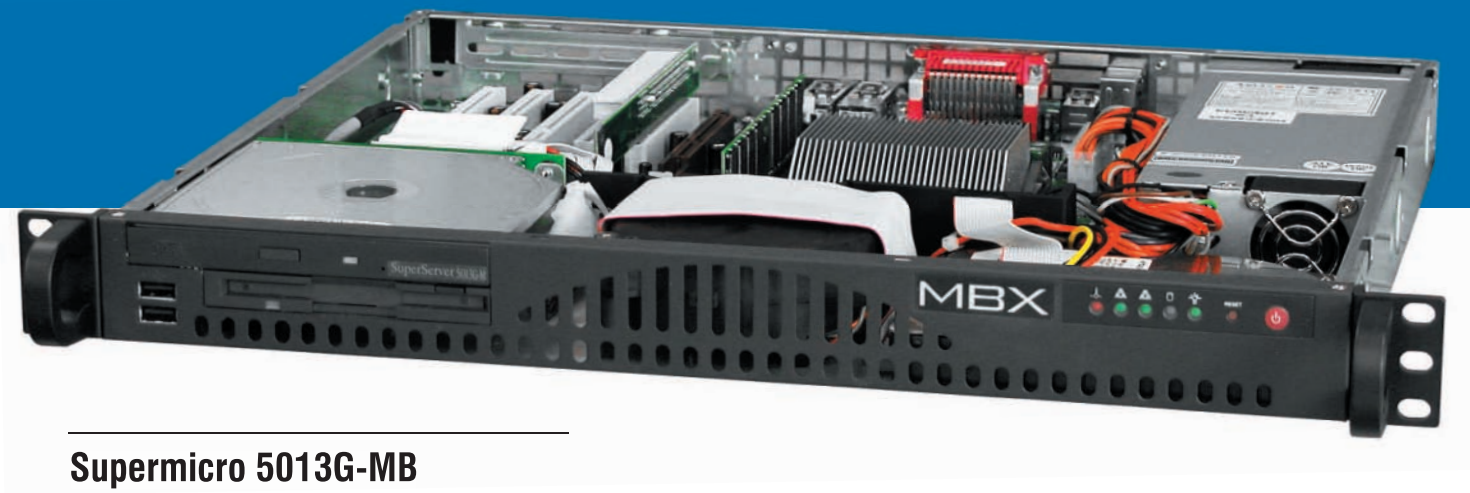
Ergonomic Comments on Ultimate Linux Box

The case on the cover for your Ultimate Linux Box is gorgeous [August 2005]. Beautiful. Amazing.

It also blows. It's a terrible design. Ghastly. Ideal for a computer show, but awful to use. There's no leg room on the box. For many folks, that would mean sitting obliquely or too far from the keyboard—both would cause strain.

Reduce Your Deployment and Support Costs

MBX is *the* leader for your server and appliance manufacturing needs



Supermicro 5013G-MB

- Intel® Pentium 4 Processor® at 3.0E GHz
- 1U Rackmount Chassis
- 512MB PC3200 DDR
- Maxtor 80GB Serial ATA Hard Drive
- Dual Onboard Gigabit NIC's
- Includes CDROM, Floppy and Video
- Lifetime toll free tech support
- 3 Year Warranty

$959 or lease for **$33**/mo.

Or Promote Your Brand

- Same Configuration as Above
- Custom Branded With Your Logo
- 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 **$38**/mo.

MBX is the leader in custom appliances. Many premier application developers have chosen MBX as their manufacturing 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™
MOTHERBOARDEXPRESS

www.mbx.com
1.800.688.2347

Intel, Intel Inside, Pentium and Xeon are trademarks and registered trademark of Intel Corporation or its subsidiaries in the United States and other countries. Lease calculated for 36 months, to approved business customers. Prices and specifications subject to change without notice. Setup fee may apply to certain branding options. Motherboard Express Company. 1101 Brown Street Wauconda, IL. 60084.

Tactile response acts as a brake reducing the amount of impact on the end of the fingers (while, agreed, increasing the finger travel), so the keyboard used may actually increase hand stress for some. Its placement isn't adjustable for height, and that can be catastrophic for arm stress.

On a much less important note, I'd point out that the case puts the cooling up very high indeed, into airspace often several degrees hotter than the rest of the room.

--
Paul Pomerleau

The coolant loops need to run well above the motherboard in order to get adequate convective cooling. For daily use, you can build a tall case without the monitor mounts or keyboard shelf.—Ed.

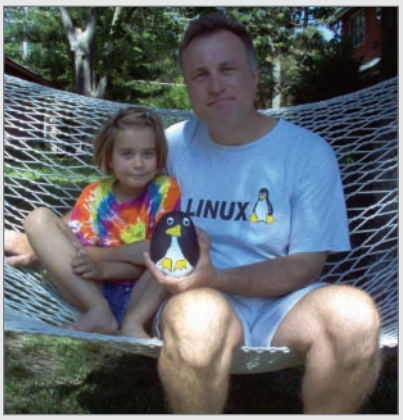
Another Happy Reader

Here's a photo of my son Merit (about 26-months old) sitting on his trusty fire truck checking out my July 2005 issue of *LJ*. When he was done, he went back into the office and picked up an Advanced C Programming book!

--
tim

Photo of the Month: a Linux Father's Day

I had a great Father's Day and hope all the other open-source dads did too. Of course, the best gift was Paige, my going-to-be-seven-in-September daughter—and you will see she painted Tux on a rock for me this Father's Day. She always has been a good drawer, and her favorite program is Tux Paint with *Tux Racer* a close second.



--
James M. Susanka

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

Pipe Tip

"Text Manipulation with sed" by Larry Richardson had some useful hints [July 2005]. For instance, I wasn't aware of the ! modification to the range field.

But writing to a file at the same time you are reading it is decidedly dangerous. You are depending on the pipe buffer between cat and sed to hold the entire contents of the file. You are also banking on the assumption that cat will be started and allowed to fill its buffer *before* the file is written and, therefore, truncated at zero length. A far better way to do this is in two steps:

```
sed -e 's/$/ mycomputer/' < \
/etc/exports > tempfile \
&& mv tempfile /etc/exports
```

The mv command is executed only if sed returns without problems. You don't want to be overwriting important files with the wrong data!

--
Allen Brown

Her First Computer T-Shirt

My daughter (2.2-years old) made the transition. She was introduced to Potato Guy and Tux Paint on my Linux box. On this day, she learned everything she needed to know about the mouse operations. A very proud moment for me, she is on the road to becoming a geek just like her daddy. Elizabeth Su WOHID Certified (Wireless Optical Human Interface Device), T-shirt awarded shortly after. You can't start them too young. This is only the beginning for them. I am just happy to be there.

--
Jesse Apple

LETTERS CONTINUED ON PAGE 94

The Straight Talk People℠
SINCE 1991

ABERDEEN

YOU CAN BUY THESE NOW OR WAIT TILL DELL FREEZES OVER

ABERDEEN STONEHAVEN A141

1U Dual Opteron™ 4 SATA/SCSI

High performance dual server for top-of-the-line processing power with ultra-dense storage capacity.

- Dual AMD Opteron™ Processors w/HyperTransport and 1MB Cache
- AMD 8000 Series Chipset w/64-bit Support
- Up to 16GB DDR-400 Reg. ECC Memory
- Up to 4 x 400GB (1.6TB) Hot-Swap SATA or 4 x 300GB (1.2TB) Hot-Swap SCSI Drives
- 400W AC Power Supply w/PFC
- **5-Year Limited Warranty**

Starting at **$1,895**

ABERDEEN STONEHAVEN A261

2U Dual Opteron™ 6 SATA/SCSI

The highest performing 2U server available for the money. "Staggering ... Powerhouse Performance ... Highest Webbench numbers we've seen to date" – *PC Magazine, December 27, 2004.*

- Dual AMD Opteron™ Processors w/HyperTransport and 1MB Cache
- AMD 8000 Series Chipset w/64-bit Support
- Up to 16GB DDR-400 Reg. ECC Memory
- Up to 6 x 400GB (2.4TB) Hot-Swap SATA or 6 x 300GB (1.8TB) Hot-Swap SCSI Drives
- 460W Hot-Swap Redundant Power Supply
- **5-Year Limited Warranty**

Starting at **$2,875**

ABERDEEN STONEHAVEN A381

3U Dual Opteron™ 8 SATA/SCSI

Gargantuan storage beast with a capacity of up to 3.2TB, room for a dual-height tape drive, at an incomparable cost/TB ratio.

- Dual AMD Opteron™ Processors w/HyperTransport and 1MB Cache
- AMD 8000 Series Chipset w/64-bit Support
- Up to 16GB DDR-400 Reg. ECC Memory
- Up to 8 x 400GB (3.2TB) Hot-Swap SATA or 8 x 300GB (2.4TB) Hot-Swap SCSI Drives
- 760W Hot-Swap Redundant Power Supply
- **5-Year Limited Warranty**

Starting at **$2,975**

ABERDEEN STONEHAVEN A124

1U Quad Opteron™ HPC

64-bit HPC environment workhorse server/cluster node. Superior cooling with plenty of power to handle any project.

- Quad AMD Opteron™ 800 Series Processors
- AMD 8000 Series Chipset w/64-bit Support
- Up to 32GB DDR-400 Reg. ECC Memory
- Up to 2 x 300GB (600GB) SCSI Hard Drives
- 500W Power Supply
- Ultra Cool with Superb Air Flow
- **5-Year Limited Warranty**

Quads Starting at **$8,265**

ABERDEEN STONEHAVEN A234

2U Quad Opteron™ 3 SATA/SCSI

Robust 64-bit server ideal for the HPC environment as a high performance server. Able to provide all the power and I/O for large databases and memory intensive projects.

- Quad AMD Opteron™ Processors w/HyperTransport and 1MB Cache
- AMD 8000 Series Chipset w/64-bit Support
- Up to 32GB DDR-400 Reg. ECC Memory
- Up to 3 x 400GB (1.2TB) Hot-Swap SATA or 3 x 300GB (900GB) Hot-Swap SCSI Drives
- 700W Power Supply
- Ultra Cool with Superb Air Flow
- **5-Year Limited Warranty**

Quads Starting at **$8,125**

ABERDEEN STONEHAVEN A484

4U Quad Opteron™ 8 SATA/SCSI

Best of both worlds, all-inclusive server with enterprise-class 64-bit HPC Quad power along with maximum storage capacity.

- Quad AMD Opteron™ Processors w/HyperTransport and 1MB Cache
- AMD 8000 Series Chipset w/64-bit Support
- Up to 32GB DDR-400 Reg. ECC Memory
- Up to 8 x 400GB (3.2TB) Hot-Swap SATA or 8 x 300GB (2.4TB) Hot-Swap SCSI Drives
- 950W 3+1 Hot Swap Redundant Power Supply
- Ultra Cool with Superb Air Flow
- **5-Year Limited Warranty**

Quads Starting at **$9,625**

Trademarks are of their respective owners. lj010

888-297-7409
www.aberdeeninc.com/linux

On the WEB

To go along with this month's theme of Personal Desktop, here are some articles from the *Linux Journal* Web site that will help you find your way through OpenOffice.org, try out some Linux audio software and rescue data from a hosed USB device:

- Do you want to move to OpenOffice.org but aren't sure what to expect? Are you trying to convince friends and/or family members to give OOo a try, but they want to know about the learning curve? If so, Bruce Byfield's article "OOo Off the Wall: What New Users Need to Know About OpenOffice.org" (www.linuxjournal.com/article/8443) is suggested reading. Bruce sheds some light on OOo's "interface shortcomings" and "the limits of its on-line help", as well as the "logic of its interface design and the importance of styles and templates in an efficient work flow".

- Audio for Linux has come a long way in the past couple of years, and Dave Phillips continues his tour of what's new for musicians and engineers, whether full-time or part-time. In recent months, he's introduced us to FreeWheeling, "a powerful loop-based performance tool" (www.linuxjournal.com/article/8445), as well as QSynth and QJackCtl, GUI front ends that "make Linux audio tasks easier and faster, letting you get straight to the music" (www.linuxjournal.com/article/8354).

- Finally, Collin Park shares his story of "How a Corrupted USB Drive Was Saved by GNU/Linux" (www.linuxjournal.com/article/8366), offering hope to those of us who have lost important data and will lose it again.

diff -u

What's New in Kernel Development

After a long and difficult life, **DevFS** is finally being removed from the Linux kernel. Created by **Richard Gooch**, DevFS has been around for years, and it represented a serious attempt to cure the runaway /dev directory. Developing DevFS was an uphill battle against many detractors, but Richard did succeed in creating a very useful tool. In the end, however, critics of DevFS won out, citing "unfixable races" and other problems, and Richard vanished from kernel development completely. **Greg Kroah-Hartman** and others then developed udev as a replacement for DevFS. Some lingering sense of the 2.6 kernel as a stable tree has made this decision slightly controversial even now, but almost certainly it's not enough to influence the outcome. Farewell DevFS—it was a valiant effort.

Recently, various folks have reported compilation problems when trying to compile the **2.4 kernel** with GCC version 4, and some developers have posted patches to address these issues; however, **Marcelo Tosatti** has stated that it is simply too late in the day for these sorts of patches to make it into the 2.4 tree. Unlike 2.6 development, the maintainers of 2.4, 2.2 and 2.0 have not decided to follow suit and abandon the idea that their trees must aim for stability. Marcelo has been trying to rein in 2.4 development ever since the first 2.6 kernel came out, but he has still allowed large IDE changes, new hardware support and other patches whose invasiveness would typically fly in the face of a push for stability. And with 2.6 development showing no sign of slowing down, Marcelo has been under constant pressure to incorporate new features into 2.4 to be available to folks who needed 2.4's stability. With the advent of the w.x.y.z tree, however, some of this pressure has undoubtedly flagged, and Marcelo has been able to tighten up the restrictions on what can and cannot get into 2.4 at this late date.

The **git** versioning system continues to grow and strengthen. **Andrew Morton**'s -mm tree will be available as a git repository, although Andrew himself has no plans to use any versioning tool for actual development. The **ALSA** Project has migrated development to git, as has **libata**. Marcelo Tosatti's 2.4 tree also will use git for ongoing development. **Linus Torvalds** is still very strongly involved with the project, and although mailing-list traffic has tapered off somewhat from its frantic early weeks, much of this is explained by the fact that folks now understand the basics of the tool, and the fundamental concepts no longer need to be explained to newcomers.

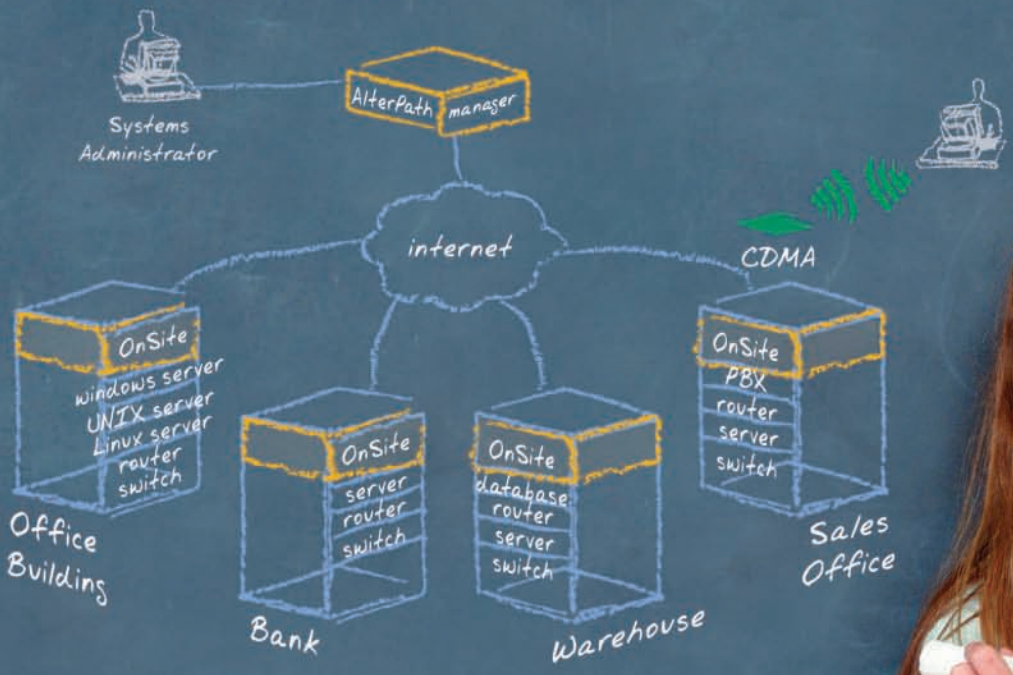
In the midst of all the version-control upheaval, it's hard to know for certain if the new w.x.y.z stable kernels are working out. But several kernel folks, including **Jeff Garzik** and **Alan Cox**, feel that this tree successfully provides a stable kernel to supplement the 2.6 tree's ongoing large-scale development. Greg Kroah-Hartman and **Chris Wright**, the primary maintainers of the w.x.y.z tree, do seem to be doing a rigorous job, not only collecting and applying patches, but adhering to Linus Torvalds' strict guidelines on what patches may be applied, and how and when they may be accepted. A number of aspects make this project less appealing than doing real development work, but Chris and Greg seem to be bearing up nicely, and the rest of us are the beneficiaries.

Martin J. Bligh has put together a set of automatic testing scripts that compile and boot all official kernel releases (including the w.x.y.z kernels) and several prominent branches like the -mm tree, within 15 minutes of their release. If a kernel boots successfully, Martin's scripts hit it with a variety of benchmarks. Compilation and boot results are recorded, benchmark results are graphed and everything is made available as a set of ongoing **kernel.org** Web pages. This is the sort of project that will not solve all bugs, but it will identify many trivial bugs, track performance problems across multiple kernel releases and may identify hard-to-find bugs that regular users would not normally see.

The relatively recent introduction of **Signed-Off-By** tags in kernel patch submissions has made a huge difference in providing a trail of authorship, so that if anything like the SCO lawsuit occurs again, it will be easy to prove who wrote any disputed source code. This was, in fact, Linus Torvalds' stated purpose in introducing the Signed-Off-By header. When first introduced, the idea was quite amorphous, with few details settled. Since then, various wrinkles have been introduced to improve its usefulness. One of the most recent of these is the addition of a From header as the first line of the body of patch e-mails. This header identifies the true author of a given patch. Before this wrinkle, the true author was assumed to be the person with the bottom-most Signed-Off-By header. This, however, became confusing and was not always adhered to. The From header is intended to leave no doubt as to the original authorship of a given patch.

—ZACK BROWN

Cyclades AlterPath™ OnSite makes branch office administration child's play



AlterPath™ OnSite

The Next-Generation IT Infrastructure

Cyclades AlterPath™ OnSite is the most comprehensive remote site and branch office administration appliance available. This small, inexpensive solution for controlling network equipment, servers and other IT infrastructure devices can

- Access, diagnose and restore remote IT devices quickly
- Download software to multiple devices automatically and simultaneously
- Configure user information, system settings and operating parameters
- Send alerts of intrusions, equipment failures and alarms

The AlterPath OnSite combines the functionality of both serial console and KVM over IP, allowing IT administrators to manage multiple servers and network devices through a single appliance. Cyclades brings it all together making remote site and branch office administration seem like child's play.

A+

For a ***FREE*** *copy of Enterprise Management Associates' white paper* **Increased IT Operations Effectiveness With Cyclades AlterPath™ System for the Out-of-Band Infrastructure,** please visit us at www.cyclades.com/ema

Over 85% of Fortune 100 choose Cyclades.

www.cyclades.com/ljb

1.888.cyclades · sales@cyclades.com

cyclades

©2005 Cyclades Corporation. All rights reserved. All other trademarks and product images are property of their respective owners. Product information subject to change without notice.

Nigel McFarlane

With the sudden death of Nigel McFarlane, the Web Development and Open Source Software communities, both in Australia and around the world, have lost one of their most well-known authors, consultants and pundits.

Although in many ways a very private person, Nigel had a professional and personal network that spanned the globe and included such on-line luminaries as the lead engineer for the open-source browser Firefox Ben Goodger, and countless others in the Open Source, Web Development and Linux communities. Since his passing, many community sites, in a number of languages, have expressed their sorrow, a testament to Nigel's influence.

A real Melbourne boy, describing the city proudly as "the World's most liveable", Nigel had science degrees from both the University of Melbourne and LaTrobe University. Even when speaking in Sydney, he was always keen to get home as soon as possible, where he would bushwalk and ramble, swim and surf.

Nigel forged a global reputation from his beloved Melbourne, in a way impossible until the 1990s. Many others have and will follow his lead, but as with much of what he did, here, too, Nigel was a pioneer. Since 1997, Nigel had become well known and respected in the Web Development and more recently Open Source Technology communities through the publication of several successful books on JavaScript, Mozilla and most recently, the increasingly popular free open-source browser Firefox.

Two earlier books on JavaScript, *Instant JavaScript* in 1997 and the co-authored *Professional JavaScript* in 2001, are still considered by many to be among the best books on the subject. More recently, the benchmark *Rapid Application Development with Mozilla*, and *Firefox Hacks* carved out a place in the increasingly important Open Source community.

Nigel's writing extended to the columns "Searching for Substance" for *InformIT*, and articles for such publications as *Linux Journal*, *DevX*, *Builder.com*, *CNet*, *The Age* and the *Sydney Morning Herald*. Nigel was an entertaining speaker as well as a writer. I particularly recall chairing a conference session that Nigel presented late last year. Often conference-goers are anxious to get early places in the meal queue, but although we had gone overtime for lunch, Nigel captivated the room. When offered the opportunity to break, the entire room turned it down, glued as they were to Nigel's presentation.

Generous with his time, energies and knowledge, Nigel contributed to mailing lists, newsgroups and forums, as well as speaking to audiences large and small at conferences and for user groups. His reach went far beyond

HLA Adventure

members.tripod.com/~panks/hlaadv.html

When *Zork* appeared on the scene in the late 1970s, computer enthusiasts from around the world were instantly hooked on the interactive fiction genre known fondly as the Text Adventure game.

HLA Adventure is the latest in a long line of public domain and free software text adventures being released by people all over the world. It combines elements from *MUD*s, *Advanced Dungeons & Dragons* and J.R.R. Tolkien's famous *The Lord of the Rings*.

Using verbs and nouns to communicate with the game world, the player moves about *HLA Adventure* with but a simple goal in mind: slay the menacing dragon at the end of a large expanse of caves. While solving this main quest,

```
HLA Adventure - Version 3.10 : hp: 8407 g: 242 exp: 281
http://members.tripod.com/~panks/hlaadv.html
vampire
dragon
>talk to dragon
The dragon bellows,'I am the last of my kind. And here you are, a
dirty human, come to slay me?! I breathe fire upon trees, rustling
the bad kami out of the poison forests and valleys of this forgotten
world. I keep my peace inside these caves, only to be disturbed by
the wretched humans which pollute this landscape! I was once happy
in my silence here, yet you come here to make me miserable! I won't
allow it, Knave! Prepare to feel my wrath!!!'
The dragon moves about angrily, stomping the floor!
>examine dragon
A large dragon with tough scales and powerful claws.
>inv
You are carrying:
 lantern
 rope
 key
 sword
 longsword
 broadsword (wielded)
 gauntlets (worn)
```

the player is also presented with nine other unique quests, which allow the player to find items and equip weapons, armor and a brightly lit lantern. Even a magical flute plays a role—useful in putting magical beasts to sleep.

Players will encounter hellhounds, werewolves, vampires, hobbits, ghosts, barbarians and demigorgons. Talk to creatures in the game with the `TALK TO` command. Once you have acquired the necessary armament and passed the requisite number of quests, you can then enter into the cave and slay the dragon for good.

Despite some bugs in the game, *HLA Adventure* is a solid, robust open-

Reclaim lost time

Levanta Intrepid ™

The world's first Linux management appliance

Plug the Levanta Intrepid ™ into your network and perform the most important Linux management tasks in a fraction of the time you spend now. And gain power and flexibility that you've never had before:

- **Fast & Portable:** Provision servers or workstations practically anywhere, anytime – in minutes. Swap them around, mix it up.
- **Flexible:** Supports commodity hardware, blades, virtual machines, and even mainframes.
- **Out of the Box:** Includes pre-defined templates for servers, workstations, & software stacks. Or create your own.
- **Total Control:** Track any file changes, by any means, at any time. And undo them at will.
- **Disaster Recovery:** Bring dead machines quickly back to life, even if they're unbootable.

Based upon technology that's already been proven in Fortune 500 enterprise data centers. Now available in a box, priced for smaller environments. **Just plug it in and go.**

© 2005 Levanta, Inc. All rights reserved. Levanta and the Levanta logo are registered marks of Levanta, Inc.

30-Day
Money-Back Guarantee

Order online by 10/30/05
Get $500 Off
Enter PROMO CODE: LJ1005

LEVANTA®
www.levanta.com
1.877.LEVANTA

Ruby on Rails

Explore a Web development framework that comes with its own Web server, magically keeps track of details for you and integrates new code without restarting. BY REUVEN M. LERNER

Ruby, an interpreted programming language that looks and feels like a cross between Smalltalk and Perl, has been around for about ten years. Ruby has been gaining in popularity over the last few years, partly because of the release of English-language books and documentation. In addition, programmers have become more interested in finding an alternative to Perl and Python for their general-purpose programming needs.

Ruby's popularity might have continued to grow slowly were it not for Ruby on Rails, a Web development framework that has become the focus of enormous attention. Everyone in the Web development world seems to be talking about Rails; magazine articles, blog postings, conference tracks and even some new books all are dedicated to Rails. Rails is supposed to be elegant, easy to use and easy to modify. Even developers with no previous Ruby experience are switching to Rails.

Does Rails live up to the hype surrounding it? To a large degree, I believe the answer is "yes"—it has a relatively shallow learning curve, it connects easily and quickly to relational databases and it makes the creation of many small- and medium-sized sites faster and easier than I would have expected. But, of course, no framework is perfect, particularly one that was released publicly only one year ago. It remains to be seen whether Rails can hold up against more-established technologies on several different fronts.

This month, we begin to look at several aspects of Ruby on Rails, so you can decide for yourself if my assessment is accurate. We begin by installing and configuring a basic Rails application. Over the next few installments of At the Forge, we will extend our application in several different ways, considering the ways in which Rails allows us to create and modify our applications.

Installing Rails

The first step in creating a Rails application is to install Ruby and then Rails itself. Most modern Linux distributions come with Ruby, although only the latest released version as of this writing (1.8.2) works with the most recent version of Rails (0.12.1). New versions of Rails have been coming out frequently, which means that one or both of these versions might have changed by the time you read this.

Assuming you have installed Ruby, you next need to install Gems. It provides access to the Ruby Gems library, which is something of a cross between SourceForge and Perl's CPAN (see the on-line Resources). Download and unpack the most recent .tar.gz file:

```
tar -zxvf rubygems-0.8.10.tar.gz
```

Enter the directory as the root user and type:

```
ruby setup.rb all
```

This installs the entire Gems package. Among other things, this installs the gem program in /usr/bin. You then can install Rails, which is distributed via Gems, with the following command:

```
gem install --remote rails
```

As with such systems as CPAN and Debian's apt, the gem program is smart enough to identify and download any dependencies it might encounter. By default, you need to answer "y" explicitly when asked if you are interested in installing any dependencies. Because Rails depends on a number of other packages, you should be sure to answer "y" when prompted.

When you are returned to the shell prompt, you can assume that Rails has been installed. However, this is not quite enough. If you are interested in working with a relational database, you also need to install a database interface library. Because I work with PostgreSQL, I installed the pure Ruby client, called postgres-pr:

```
gem install --remote postgres-pr
```

Somewhat confusingly, there also is a set of PostgreSQL client libraries (called postgresql) that can be used with Ruby. However, it seems as though most Rails developers are working with the postgres-pr library, at least for now.

Creating an Application

Once Rails is installed, we can create a simple "Hello, world" program. To do this, we use the `rails` command, which is installed in /usr/bin/ by default. Because our example application is a Weblog, we call the application blog. For reference, the name of the application doesn't have to be linked to the name of the URL under which it will appear. Type:

```
rails blog
```

Running this produces a fair amount of output, listing the files that have been created on our filesystem. When we give only a single name, blog, the application is created inside of a directory with that name. We can keep all of our applications inside of a single container directory, such as ~/Rails, with:

```
mkdir ~/Rails
rails ~/Rails/blog
```

If we look inside the newly created application directory, we see a number of directories and files. The script directory contains administrative programs, written in Ruby, of course. The public directory contains static HTML files, as well as images, stylesheets, JavaScript code and templates that you

EmperorLinux

...where Linux & laptops converge

You choose your laptop... from a wide selection of top tier laptops manufactured by IBM/Lenovo, Dell, Sharp, and Sony. They come in all sizes from two pound ultra-portables to eight pound desktop replacements; get exactly as much Linux laptop as you need. Need help deciding? Our experts will help you select a Linux laptop to meet your needs.

The Meteor: 3lb Linux

- Sharp Actius MM20/MP30
- 10.4" XGA screen
- 1.6 GHz Transmeta Efficeon
- 20-40 GB hard drive
- 512-1024 MB RAM
- CDRW/DVD (MP30)
- 802.11b/g wireless
- ACPI hibernate
- 1" thin
- Ask about the 3D Molecule

The SilverComet: 4 lb Linux

- Sony VAIO S380
- 13.3" WXGA+ screen
- X@1280x800
- 1.6-2.13 GHz Pentium-M
- 40-100 GB hard drive
- 256-1024 MB RAM
- CDRW/DVD or DVD-RW
- 802.11b/g wireless
- ACPI hibernate
- Ask about the 17" Gazelle

You choose your distribution... from among the most popular Linux distributions available. We'll install the distribution you select, then we'll install our custom, laptop-specific kernel and configure your distribution for full hardware support, including: X at the native resolution, wireless ethernet, power management, 3-D graphics, optical drives, and more.

The Toucan: 5 lb Linux

- IBM/Lenovo ThinkPad T series
- 14.1" SXGA+/15.0" UXGA
- X@1400x1050/X@1600x1200
- ATI FireGL graphics
- 1.6-2.13 GHz Pentium-M 7xx
- 40-80 GB hard drive
- 512-2048 MB RAM
- CDRW/DVD or DVD-RW
- APM suspend/hibernate
- Ask about the 3 lb Raven X41

The Rhino: 7 lb Linux

- Dell Latitude D810/M70
- 15.4" WUXGA screen
- X@1920x1200
- NVidia Quadro or ATI Radeon
- 1.73-2.26 GHz Pentium-M 7xx
- 30-100 GB hard drive (7200 rpm)
- 256-2048 MB RAM
- CDRW/DVD or DVD±RW
- 802.11a/b/g wireless, GigE
- Ask about the tiny Koala X1

Let EmperorLinux do the rest. 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/Lenovo, Dell, Sharp, and Sony. We offer a range of the latest Linux distributions, as well as Windows dual boot options. All systems come with one year of Linux technical support by both phone and email, and full manufacturers' warranties apply.

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

Model prices, specifications, and availability may vary. All trademarks are the property of their respective owners.

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

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, Pentium, and
*Important Information. All prices, specifications and promotional offers are subject to change without notice. Ciara cannot be responsi

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™

l Pentium III Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries
ble for typography errors, photographies errors, pricing errors. All pricing in US dollar. Shipping and applicable taxes are not included.

may use in your application.

The directory you are mostly likely to work with is app, which contains the application itself. The app directory contains subdirectories named models, views and controllers. This design reflects the fact that Rails uses the MVC (model/view/controller) style widely used in many modern desktop and Web applications.

In an MVC architecture, we divide our work into three parts—the controller, which acts like a switchboard, invoking the appropriate model and view; the model, which contains the data and some of the logic; and the view, which displays information to the user. If you have ever built a database-backed site with PHP and Smarty templates or with Zope and its Page Templates or even with Java and JavaServer Pages (JSPs), you already are familiar with at least some of these ideas. Rails simply makes them more explicit with its prenamed directory structure.

Although it can't do much, we now can start our empty Rails application with:

```
cd ~/Rails/blog
ruby script/server
```

This starts the WEBrick HTTP server on port 3000. To access this fairly empty Rails site, we point our browsers to an appropriate IP address or hostname. In my particular case, I started Rails on my test server, whose IP address is 192.168.2.3. I thus point my Web browser to http://192.168.2.3:3000/. And sure enough, there I see a "Welcome on board" message, indicating I have set up Rails correctly.

Customizing the Behavior

Now that we know how to see the default message, let's move toward a "Hello, world" program. In Rails, there are two basic ways to do this. We can create a controller that returns HTML to the user's browser, or we can create a view that does the same. Let's try it both ways, so that we can better understand the relationship between controllers and views.

If all we want to do is include a simple, static HTML document, we can do so in the public directory. In other words, the file blog/public/foo.html is available under WEBrick—started by executing `blog/script/server`—at the URL /foo.html.

Of course, we're interested in doing something a bit more interesting than serving static HTML documents. We can do that by creating a controller class and then defining a method within that class to produce a basic "Hello, world" message. Admittedly, this is a violation of the MVC separation that Rails tries to enforce, but as a simple indication of how things work, it seems like a good next step.

To generate a new controller class named MyBlog, we enter the blog directory and type:

```
ruby script/generate controller MyBlog
```

Each time we want to create a new component in our Rails application, we call upon `script/generate` to create a skeleton. We then can modify that skeleton to suit our specific needs. As always, Rails tells us what it is doing as it creates the files and directories:

```
exists  app/controllers/
exists  app/helpers/
create  app/views/my_blog
exists  test/functional/
create  app/controllers/my_blog_controller.rb
create  test/functional/my_blog_controller_test.rb
create  app/helpers/my_blog_helper.rb
```

Also notice how our controller class name, MyBlog, has been turned into various Ruby filenames, such as app/views/my_blog and app/helpers/my_blog_helper.rb. Create several more controller classes, and you should see that all of the names, like FooBar, are implemented in files with names like foo_bar. This is part of the Rails convention of keeping names consistent. This consistency makes it possible for Rails to take care of many items almost magically, especially—as we will see next month—when it comes to databases.

The controller that interests us is my_blog_controller.rb. If you open it up in an editor, you should see that it consists of two lines:

```
class MyBlogController < ApplicationController
end
```

In other words, this file defines MyBlogController, a class that inherits from the ApplicationController class. As it stands, the definition is empty, which means that we have neither overridden any methods from the parent class nor written any new methods of our own. Let's change that, using the built-in render_text method to produce some output:

```
class MyBlogController < ApplicationController

  def hello_world
render_text "Hello, world"
  end

end
```

After adding this method definition, we can see its results by going to http://192.168.2.3:3000/MyBlog/hello_world.

Notice how the URL has changed: static items in the public directory, such as our file foo.html, sit just beneath the root URL, /. By contrast, our method hello_world is accessed by name, under the controller class that we generated. Also notice that we did not need to restart Rails in order to create and test this definition. As soon as a method is created or changed, it immediately is noticed and integrated into the current Rails system.

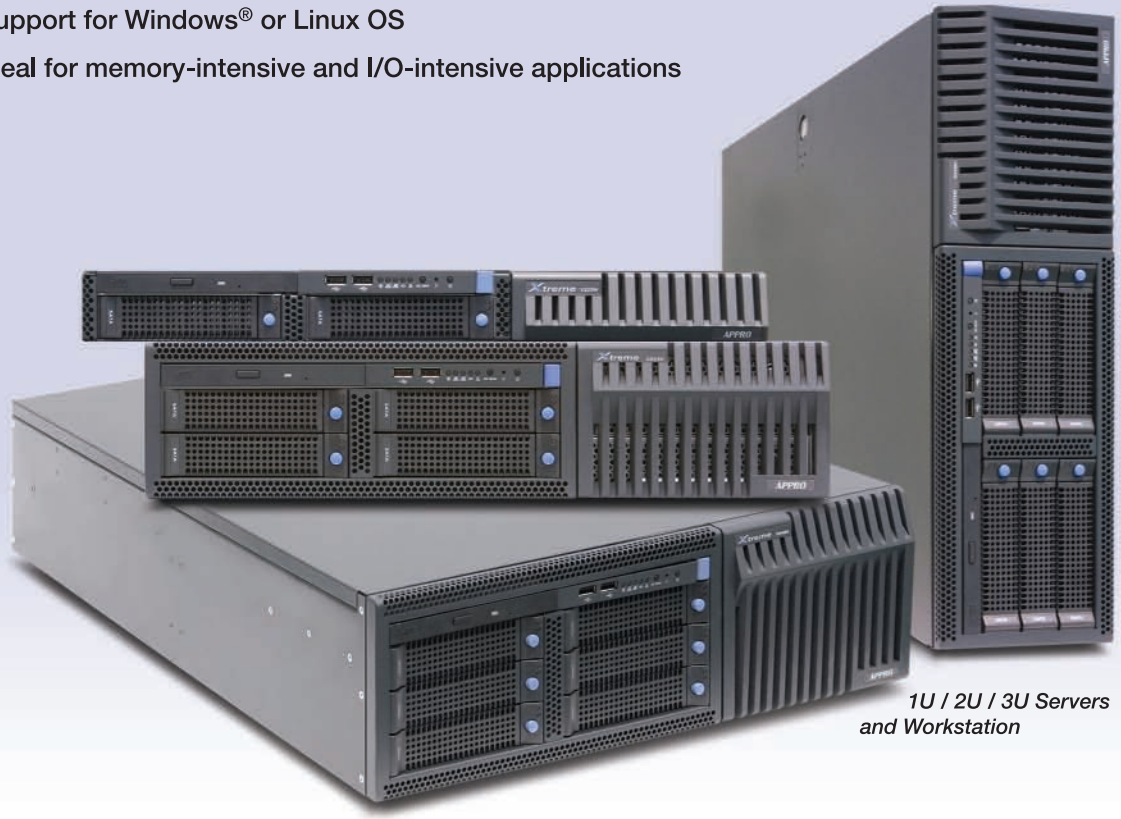
If we define an index method for our controller class, we can indicate what should be displayed by default:

```
class MyBlogController < ApplicationController

  def hello_world
render_text "Hello, world"
  end

  def index
render_text "I am the index!"
```

Let's be Xtreme

Introducing Appro *Xtreme* Servers & Workstations with 8 DIMM Sockets per CPU

- 2-way or 4-way, Single or Dual-Core AMD Opteron™ processors
- **Largest memory capacity - 8 DIMM Sockets per CPU, up to 128GB**
- PCI-Express technology to increase I/O bandwidth and reduce system latency
- Outstanding Remote Management – IPMI 2.0 compliant
- Cable-free design, ready to run, simple to install, service and maintain
- Support for Windows® or Linux OS
- Ideal for memory-intensive and I/O-intensive applications



1U / 2U / 3U Servers and Workstation

AMD Opteron™ Processors - AMD64 dual-core technology reduces memory latency and increases data throughput
- Dual-core processors with Direct Connect Architecture deliver the best performance per watt in the industry with little or no increase in power consumption or heat dissipation.

APPRO
HPC Cluster Solutions

Appro delivers high-performance computing solutions to help you maximize productivity for a solid ROI. On-site maintenance and installation services are also available.

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

AMD 64 Opteron™

```
  end

end
```

Of course, it's not that exciting to be able to produce static text. Therefore, let's modify our index method such that it uses Ruby's built-in Time object to show the current date and time:

```
def index
    render_text "The time is now " + Time.now.to_s + "\n"
end
```

And voilà! As soon as we save this modification to disk, the default URL (http://192.168.2.3:3000/MyBlog/, on my computer) displays the time and date at which the request was made, as opposed to a never-changing "Hello, world" message.

Let's conclude this introduction to Rails by separating the controller from its view once again. In other words, we want to have the controller handle the logic and the view handle the HTML output. Once again, Rails allows us to do this easily by taking advantage of its naming conventions. For example, let us modify our index method again, this time removing its entire body:

```
def index
end
```

This might seem strange at first glance. It tells Rails that the MyBlog controller class has an index method. But it doesn't generate any output. If you attempt to retrieve the same URL as before, Rails produces an error message indicating that it could not find an appropriate template.

Because the template is a view, we can define it inside of the blog/app/views directory of our application. And because we are defining the index view for the MyBlog class, we modify the index.rhtml file in the my_blog subdirectory of views. Notice how Rails turns ThisName into this_name when it comes to directories. Doing so saves users from having to think about capitalization in URLs, while staying consistent with traditional Ruby class naming conventions.

.rhtml files are a Ruby version of the same kind of template that you might have seen before. It acts similarly to ASP and JSP syntax, with <% %> blocks containing code and <%= %> blocks containing expressions that should be interpolated into the template. However, nothing stops us from creating an .rhtml template that actually is static:

```
<html>
<head>
<title>
    Hello, again!
</title>
</head>

<body>
<p>Hello, again!</p>
</body>

</html>
```

Consider what happens now if you attempt to load MyBlog in your browser. The controller class MyBlog is handed the request. Because no method was named explicitly, the index method is invoked. And because index doesn't produce any output, the my_blog/index.rhtml template is returned to the user.

Finally, let's take advantage of our template's dynamic properties to set a value in the controller and pass that along to the template. We modify our index method to read:

```
def index
  @now = Time.now.to_s
end
```

Notice how we have used an @ character at the beginning of the variable @now. I found this to be a little confusing at first, as @ normally is used as a prefix for instance variables in Ruby. But it becomes fairly natural and logical after a little time.

Finally, we modify our template such that it incorporates the string value contained in @now:

```
<html>
<head>
<title>
    Hello, world!
</title>
</head>

<body>
<p>Hello, world!</p>
<p>It is now <%= @now %>.</p>
</body>

</html>
```

Once again, you can retrieve the page even without restarting Ruby. You should see the date and time as kept on the server, updated each time you refresh the page.

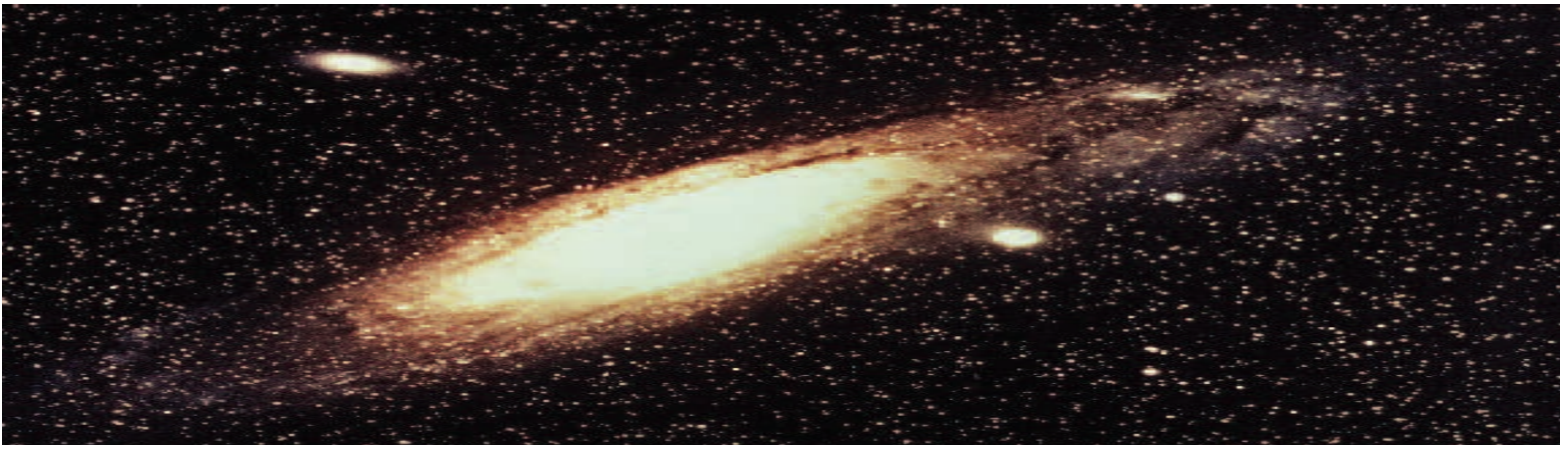
Conclusion

Ruby on Rails is, without a doubt, one of the most talked-about Web technologies to emerge in the past few years. This month, we saw how straightforward it is to create a new Rails application, to create a controller and a view and to integrate them using a combination of naming conventions and relatively standard template syntax. However, we did not discuss views, particularly those associated with a relational database. Next month, we will do exactly that, connecting Rails to the PostgreSQL database. I believe doing so will begin to show why people are so excited about Rails and why it might be a good tool for many Web developers to learn.

Resources for this article: www.linuxjournal.com/article/8457.

Reuven M. Lerner, a longtime 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.

MORE SPACE. LESS MONEY.



UNLIMITED AFFORDABLE NETWORK STORAGE

Everybody needs more space. And they need to spend less money. What if you can both have more space and spend less money?

What if you could put 7½ terabytes in only 3 rack units? What if that 7½ terabytes cost less than $10,000? Including the SATA disk drives. Imagine if you could glue it all together with a RAID appliance into one system. What if you could add as much storage as you wanted, one shelf at a time, and never have to `fork-lift' anything?

Coraid's new SATA EtherDrive Storage allows you to do just that. Using industry standard SATA disk drives, EtherDrive Storage connects disks directly to your Ethernet network. Each disk appears as a local drive to any Linux, FreeBSD or Solaris system using our open ATA-over-Ethernet (AoE) protocol. Since the disks just appear as local drives you already know how to use them.

The EtherDrive® SATA Storage Shelf is a 3U rack-mount network appliance that contains 15 SATA drive slots. Its triple redundant power supply protects you from your most likely failure. Its dual Gb Ethernet interfaces allow your data to go fast; 200MB per second. And at a very affordable price. List price for the EtherDrive Storage Shelf, without disks, is only $3,995.

Our companion product, the RAIDBlade RAID controller, allows a virtually unlimited number of Storage Shelves to be combined into a set of logical AoE storage devices.

Now you can have unlimited storage at a very affordable price. For complete information, visit our website at www.coraid.com, or call, toll-free, 1-877-548-7200. And we'll show how we've made network storage so affordable, you can have all the space you want.

www.coraid.com
info@coraid.com
1.706.548.7200

CORAID

Network Programming in the Kernel

Take a tour of the kernel's networking functionality by writing a network client that runs in kernel space.

BY PRADEEP PADALA AND RAVI PARIMI

All Linux distributions provide a wide range of network applications—from dæmons that provide a variety of services such as WWW, mail and SSH to client programs that access one or more of these services. These programs are written in user mode and use the system calls provided by the kernel to perform various operations like network read and write. Although this is the traditional method of writing programs, there is another interesting way to develop these applications by implementing them in the kernel. The TUX Web server is a good example of an application that runs inside the kernel and serves static content. In this article, we explain the basics of writing network applications within the kernel and their advantages and disadvantages. As an example, we explain the implementation of an in-kernel FTP client.

Advantages and Disadvantages of In-Kernel Implementations

Why would one want to implement applications within the kernel? Here are a few advantages:

- When a user-space program makes a system call, there is some overhead associated in the user-space/kernel-space transition. By programming all functionality in the kernel, we can make gains in performance.
- The data corresponding to any application that sends or receives packets is copied from user mode to kernel mode and vice versa. By implementing network applications within the kernel, it is possible to reduce such overhead and increase efficiency by not copying data to user mode.
- In specific research and high-performance computing environments, there is a need for achieving data transfers at great speeds. Kernel applications are useful in such situations.

On the other hand, in-kernel implementations have certain disadvantages:

- Security is a primary concern within the kernel, and a large class of user-mode applications are not suitable to be run directly in the kernel. Consequently, special care needs to be taken while designing in-kernel applications. For example, reading and writing to files within the kernel is usually a bad idea, but most applications require some kind of file I/O.
- Large applications cannot be implemented in the kernel due to memory constraints.

Network Programming Basics

Network programming is usually done with sockets. A socket serves as a communication end point between two processes. In this article, we describe network programming with TCP/IP sockets.

Server programs create sockets, bind to well-known ports, listen and accept connections from clients. Servers are usually designed to accept multiple connections from clients—they either fork a new process to serve each client request (concurrent servers) or completely serve one request before accepting more connections (iterative servers). Client programs, on the other hand, create sockets to connect to servers and exchange information.

FTP Client-Server Interaction

Let's take a quick look at how an FTP client and server are implemented in user mode. We discuss only active FTP in this article. The differences between active and passive FTP are not relevant to our discussion of network programming here.

Socket Programming Basics

Here is a brief explanation of the design of an FTP client and server. The server program creates a socket using the `socket()` system call. It then binds on a well-known port using `bind()` and waits for connections from clients using the `listen()` system call. The server then accepts incoming requests from clients using `accept()` and forks a new process (or thread) to serve each incoming client request.

The client program creates a control socket using `socket()` and next calls `connect()` to establish a connection with the server. It then creates a separate socket for data transfer using `socket()` and binds to an unprivileged port

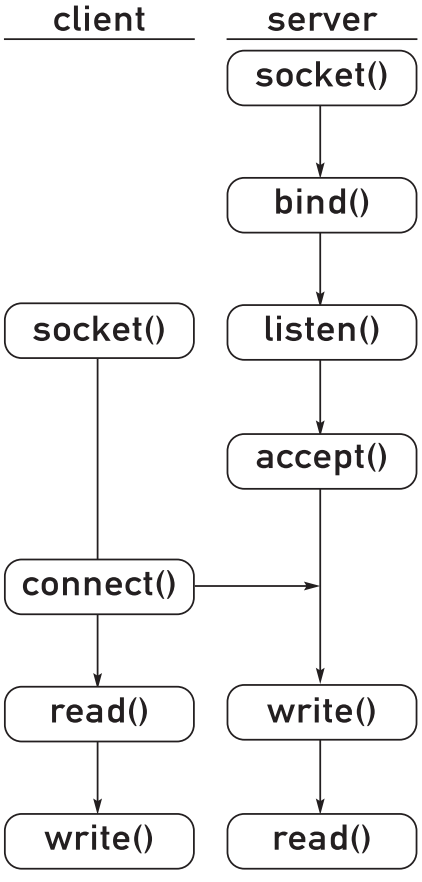
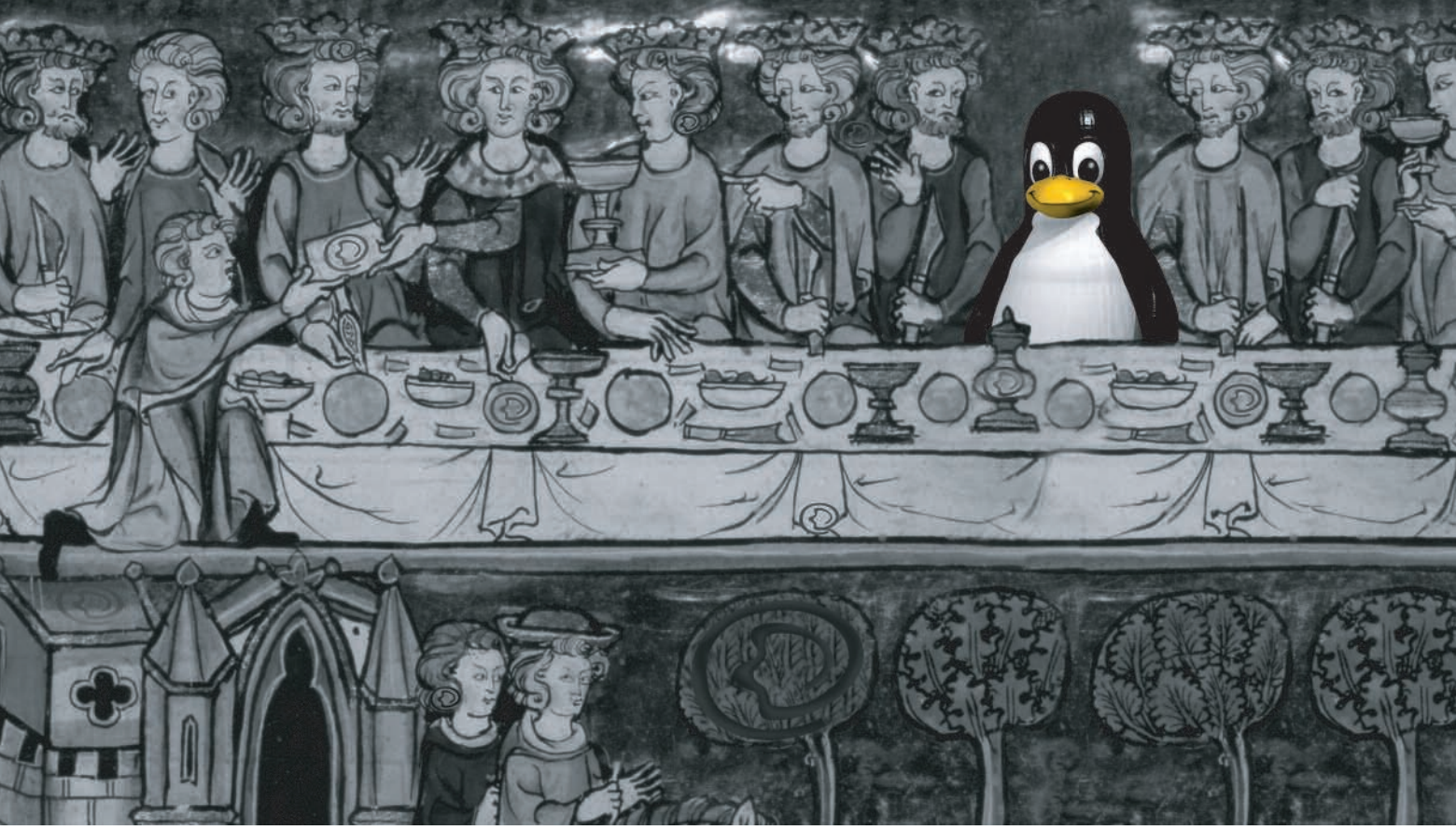


Figure 1. The FTP protocol uses two sockets: one for control messages and one for data. Here's how the first connection, used for commands, gets set up.



The Holy Grail

Easy, high-performance clustering. For years, many searched, but none could find it. Some said it didn't exist. But not the Penguin.

Penguin Computing® made easy, high-performance clustering a quest. Now you can find Linux hardware and software solutions, configured to order, driven by Scyld's commercially supported, industry-leading Linux clustering software.

For the turnkey clusters you need to run even your most important applications, come to Penguin Computing. Penguin Computing's dedicated experts, who are 100% focused on Linux, are waiting to serve.

Powerful, easy clustering. It's the once and future thing. Love what you do.

www.penguincomputing.com

Highly SCYLD

PENGUIN COMPUTING®

Penguin Computing is a registered trademark of Penguin Computing, Inc. Scyld, Scyld Software, Scyld Beowulf, and the Scyld Block Logo are trademarks of Scyld Software, Inc. Linux is a registered trademark of Linus Torvalds. Other names are for informational purposes only and may be trademarks of their respective owners.

(>1024) using `bind()`. The client now `listen()`s on this port for data transfer from the server. The server now has enough knowledge to honor a data transfer request from the client. Finally, the client uses `accept()` to accept connections from the server to send and receive data. For sending and receiving data, the client and server use the `write()` and `read()` or `sendmsg()` and `recvmsg()` system calls. The client issues `close()` on all open sockets to tear down its connection to the server. Figure 1 sums it up.

FTP Commands

Here is a list of a few FTP commands we used. Because our program provides only a basic implementation of the protocol, we discuss only the relevant commands:

- The client sends a `USER <username>\r\n` command to the server to begin the authentication process.
- To send the password, the client uses `PASS password\r\n'`.
- In some cases, the client sends a PORT command to inform the server of its preferred port for data transfer. In such cases, the client sends `PORT <a1,a2,a3,a4,p1,p2>\r\n`. The RFC for FTP requires that the a1–a4 constitute the 32-bit IP address of the client, and p1–p2 constitute the 16-bit port number. For example, if the client's IP address is 10.10.1.2 and it chooses port 12001 for data transfer, the client sends `PORT 10,10,1,2,46,225`.
- Some FTP clients request, by default, that data be transferred in binary format, while others explicitly ask the server to enable data transfer in binary mode. Such clients send a `TYPE I\r\n` command to the server to request this.

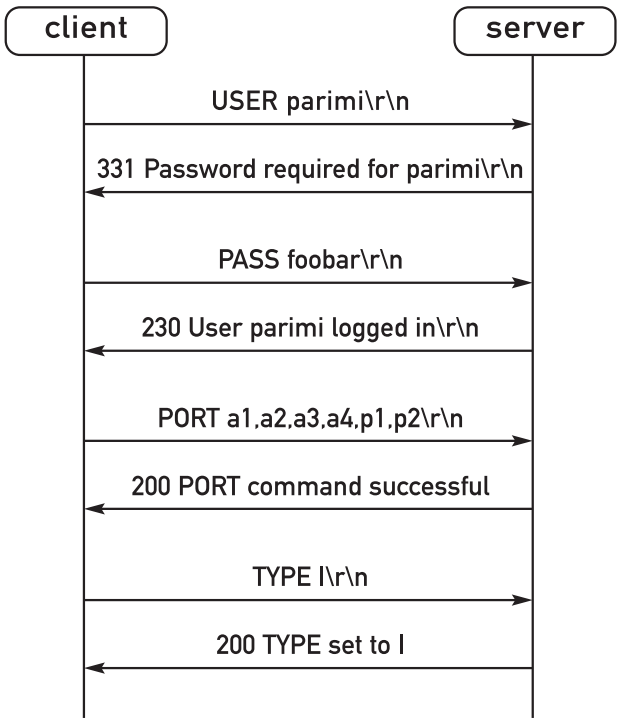


Figure 2. The client issues FTP commands over the control connection to set up the file transfer.

Figure 2 is a diagram that shows a few FTP commands and their responses from the server.

Socket Programming in the Kernel

Writing programs in the kernel is different from doing the same in user space.

We explain a few issues concerned with writing a network application in the kernel. Refer to Greg Kroah-Hartman's article "Things You Never Should Do in the Kernel" (see the online Resources). First, let's examine how a system call in user space completes its task. For example, look at the `socket()` system call:

```
sockfd = socket(AF_INET,SOCK_STREAM,0);
```

When a program executes a system call, it traps into the kernel via an interrupt and hands over control to the kernel. Among other things, the kernel performs various tasks, such as saving contents of registers, making changes to address space boundaries and checking for errors with system call parameters. Eventually, the `sys_socket()` function in the kernel is responsible for creating the socket of a specified address and family type, finding an unused file descriptor and returning this number back to user space. Browsing through the kernel's code, we can trace the path followed by this function (Figure 3).

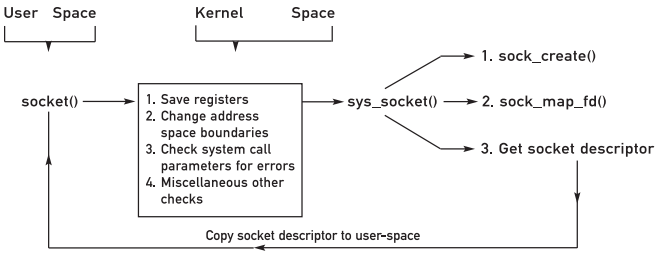


Figure 3. Behind the scenes of a system call: when user space executes socket(), the kernel does necessary housekeeping and then returns a new file descriptor.

Design of an FTP Client

We now explain the design and implementation of a kernel FTP client. Please follow through the code available at the *Linux Journal* FTP site (see Resources) as you read through this article. The main functionality of this client is written in the form of a kernel module that adds a system call dynamically that user-space programs can invoke to start the FTP client process. The module allows only the root user to read a file using FTP. The user-space program that calls the system call in this module should be used with extreme caution. For example, it is easy to imagine the catastrophic results when root runs:

```
./a.out 10.0.0.1 10.0.0.2 foo_file /dev/hda1/*
```

and overwrites /dev/hda1 with a downloaded file from 10.0.0.1.

Exporting sys_call_table

We first need to configure the Linux kernel to allow us to add new system calls via a kernel module dynamically. Starting

DON'T BE SQUARE! GET CUBED!

L Series Laptop - LS1250-L
Light &Thin Performance
Starting at $1,562.54

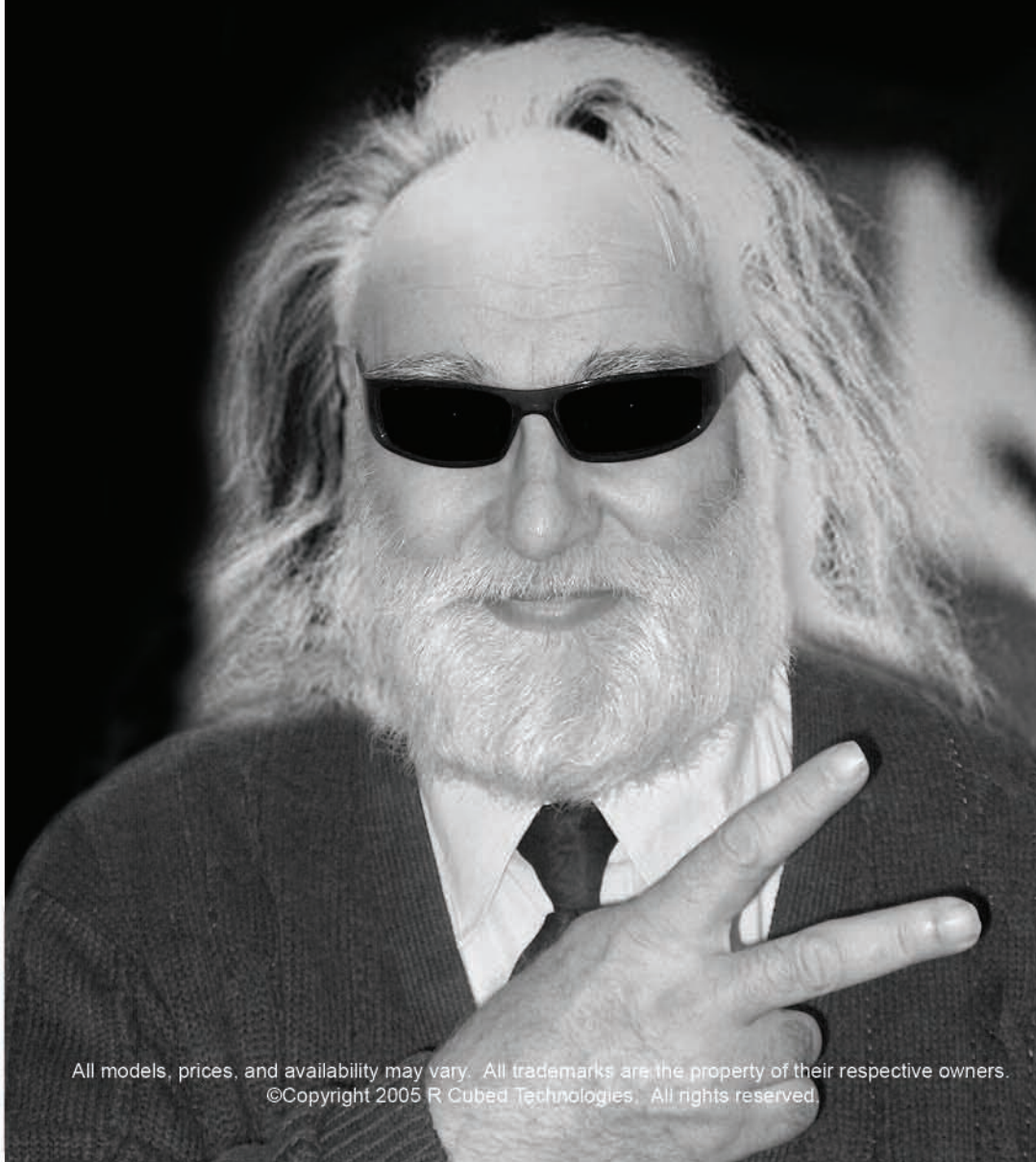
G Series Laptop - GW1550-L
Essential Technology on a Budget
Starting at $1,302.54

X Series Laptop - XW1550-L
Extreme Technology & Performance
Starting at $1,608.04

R Cubed Technologies has provided pre-installed Linux laptops without OS tax since 2003. We customize the Fedora Linux distribution for each laptops configuration providing support for: PCMCIA, USB, FireWire, X, CD/DVD/CDRW/DVDRW, Sound, Power Management, Ethernet, Modem, Wireless, and more. Our laptops are equipped with Intel Centrino Mobile Technology. We also offer Windows dual boot options. All of our laptops come with a one year parts and labor warranty. Visit us online at www.shoprcubed.com or call 309.34.CUBED for details.

R3 Technologies™
Working Hard to Bring Technologies to Life.

309.34.CUBED
www.shoprcubed.com



All models, prices, and availability may vary. All trademarks are the property of their respective owners.
©Copyright 2005 R Cubed Technologies. All rights reserved.

with version 2.6, the symbol sys_call_table is no longer exported by the kernel. For our module to be able to add a system call dynamically, we need to add the following lines to arch/i386/kernel/i386_ksyms.c in the kernel source (assuming you are using a Pentium-class machine):

```
extern void *sys_call_table;
EXPORT_SYMBOL(sys_call_table);
```

After recompiling the kernel and booting the machine into it, we are all set to run the FTP client. Refer to the Kernel Rebuild HOWTO (see Resources) for details on compiling a kernel.

Module Basics

Let's examine the code for the module first. In the code snippets in this article, we omit error checking and other irrelevant details for clarity. The complete code is available from the *LJ* FTP site (see Resources):

```
#include <linux/init.h>
#include <linux/module.h>
#include <linux/kernel.h>

/* For socket etc */
#include <linux/net.h>
#include <net/sock.h>
#include <linux/tcp.h>
#include <linux/in.h>
#include <asm/uaccess.h>
#include <linux/file.h>
#include <linux/socket.h>
#include <linux/smp_lock.h>
#include <linux/slab.h>

...

int ftp_init(void)
{

    printk(KERN_INFO FTP_STRING
    "Starting ftp client module\n");
    sys_call_table[SYSCALL_NUM] = my_sys_call;
    return 0;
}

void ftp_exit(void)
{
    printk(KERN_INFO FTP_STRING
    "Cleaning up ftp client module, bye !\n");
    sys_call_table[SYSCALL_NUM] = sys_ni_syscall;
}

...
```

The program begins with the customary include directives. Notable among the header files are linux/kernel.h for KERN_ALERT and linux/slab.h, which contains definitions for kmalloc() and linux/smp_lock.h that define kernel-locking routines. System calls are handled in the kernel by functions with the same names in user space but are prefixed with sys_. For example, the sys_socket function in the kernel handles the task of the socket() system call. In this module, we are using system call number 223 for our new system call. This method is not foolproof and will not work on SMP machines. Upon unloading the module, we unregister our system call.

The System Call

The workhorse of the module is the new system call that performs an FTP read. The system call takes a structure as a parameter. The structure is self-explanatory and is given below:

```
struct params {
    /* Destination IP address */
    unsigned char destip[4];
    /* Source IP address */
    unsigned char srcip[4];
    /* Source file - file to be downloaded from
       the server */
    char src[64];
    /* Destination file - local file where the
       downloaded file is copied */
    char dst[64];
    char user[16]; /* Username */
    char pass[64]; /* Password */
};
```

The system call is given below. We explain the relevant details in next few paragraphs:

```
asmlinkage int my_sys_call
(struct params __user *pm)
{
    struct sockaddr_in saddr, daddr;
    struct socket *control= NULL;
    struct socket *data = NULL;
    struct socket *new_sock = NULL;

    int r = -1;
    char *response = kmalloc(SNDBUF, GFP_KERNEL);
    char *reply = kmalloc(RCVBUF, GFP_KERNEL);

    struct params pmk;

    if(unlikely(!access_ok(VERIFY_READ,
                pm, sizeof(pm))))
        return -EFAULT;
    if(copy_from_user(&pmk, pm,
       sizeof(struct params)))
        return -EFAULT;
    if(current->uid != 0)
        return r;

    r = sock_create(PF_INET, SOCK_STREAM,
                    IPPROTO_TCP, &control);

    memset(&servaddr,0, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_port = htons(PORT);
    servaddr.sin_addr.s_addr =
        htonl(create_address(128, 196, 40, 225));
```

Flexibility to power the enterprise.

From mail servers to databases, ZT Servers powered by the 64-bit Intel® Xeon™ Processor can run the full range of 32-bit applications and offer extended flexibility for your 64-bit needs. So you can create powerful, all-purpose IT infrastructure that enhances business agility – and the bottom line.

ZT Optimum 1U Server X9493

Intel® Xeon™ Processors 3 GHz
- **Upgradable to Dual Intel® Xeon™ Processors 3.60 GHz, 1MB L2 Cache, 800MHz FSB**
- Intel® E7320 Chipset Server Board
- 512MB ECC Registered DDR 333 SDRAM (Up to 4GB)
- 2 x Seagate® 300GB 10,000rpm SCSI Hard Drive (Raid 1)
- 4 x 1" SCSI Hot-Swappable Drive Bays
- Slim CD-ROM & Floppy Drive
- 2 x Intel® 10/100/1000 Gigabit Network Controller
- 1U Rackmount Chassis w/420W Cold-Swappable Power Supply
- 3-Year Limited Warranty and First Year On-Site Service

$3,999

ZT Optimum 2U Server X9494

Dual Intel® Xeon™ Processors 3GHz
- **Upgradable to Dual Intel® Xeon™ Processors 3.60 GHz, 1MB L2 Cache, 800MHz FSB**
- Intel® E7520 Chipset Server Board
- 1.0GB ECC Registered DDR 333 SDRAM (Up to 16GB)
- 4 x Seagate® 250GB SATA Hard Drive (Total 1TB Storage)
- 6 x 1" Hot-Swap SATA Drive Bays
- Slim CD-ROM & Floppy Drive
- 8 Channel SATA Controller (RAID 0,1,JBOD Support)
- 2 x Intel® 10/100/1000 Gigabit Network Controller
- 2U Rackmount Chassis w/550W Power Supply
- 3-Year Limited Warranty and First Year On-Site Service

$2,499

ZT Optimum 3U Server X9495

Dual Intel® Xeon™ Processors 3GHz
- **Upgradable to Dual Intel® Xeon™ Processors 3.60 GHz, 1MB L2 Cache, 800MHz FSB**
- Intel® E7520 Chipset Server Board
- 1GB ECC Registered DDR2 400 SDRAM (Up to 16GB)
- 8 x Seagate® 300GB SATA Hard Drive (Total 2.4TB Storage)
- 8 x 1" Hot-Swap SATA Drive Bays
- DVD±RW Burner & Floppy Drive
- 8 Channel SATA Controller (RAID 0, 1, 5, 10 , 50 Support)
- 2 x Intel® 10/100/1000 Gigabit Network Controller
- 3U Rackmount Chassis w/550W Power Supply
- 3-Year Limited Warranty and First Year On-Site Service

$4,299

ZT Optimum 4U Server X9496

Intel® Xeon™ Processors 3GHz
- **Upgradable to Dual Intel® Xeon™ Processors 3.60 GHz, 1MB L2 Cache, 800MHz FSB**
- Intel® E7320 Chipset Server Board
- 1.0GB ECC Registered DDR 333 SDRAM (Up to 16GB)
- 4 x Seagate® 250GB SATA Hard Drive (Total 1TB Storage)
- 7 x 1" Hot-Swap SATA Drive Bays
- DVD±RW Burner & Floppy Drive
- 4 Channel Raid Controller (RAID 0, 1, 1/0, 5, JBOD Support)
- 2 x Intel® 10/100/1000 Gigabit Network Controller
- 4U Rackmount Chassis w/550W Redundant Cooling Power Supply
- 3-Year Limited Warranty and First Year On-Site Service

$2,199

New Powerful 64-bit Server Line

- Friendly Server Specialists
- Complete Solution Provider
- Flexibility and Variety : Customize platforms with leading-edge products
- We Build the Future !

1. OEM Computer Manufacturer

- 3 year warranty with lifetime tech support
- Reseller and volume pricing available.

New Accounts Receive Free Gift

- Personal attention (Dedicated Technical Sales Team)
- Call now to customize using the latest technology

Find out how ZT Insider Program can help maximize your Business Solution

Go to ztgroup.com/go/linuxjournal

Call 866- ZTGROUP (866-984-7687)

Prmote code : lj1005

Purchaser is responsible for all freight costs on all returns of merchandise. Full credit will not be given for incomplete or damaged returns. Absolutely no refunds for merchandise returned after 30 days. All prices and configurations are subject to change without notice and obligation. Opened software is non-refundable. All returns have to be accompanied with an RMA number and must be in re-sellable condition including all original packaging. System's picture may include some equipments and/or accessories, which are not standard features. Not responsible for errors in typography and/or photography. All rights reserved. All brands and product names, trademarks or registered trademarks are property of their respective companies. Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, Pentium, and Pentium III Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

```
    r = control->ops->connect(control,
              (struct sockaddr *) &servaddr,
              sizeof(servaddr), O_RDWR);
    read_response(control, response);
    sprintf(temp, "USER %s\r\n", pmk.user);
    send_reply(control, temp);
    read_response(control, response);
    sprintf(temp, "PASS %s\r\n", pmk.pass);
    send_reply(control, temp);
    read_response(control, response);
```

We start out by declaring pointers to a few `socket` structures. `kmalloc()` is the kernel equivalent of `malloc()` and is used to allocate memory for our character array. The array's response and reply will contain the responses to and replies from the server.

The first step is to read the parameters from user mode to kernel mode. This is customarily done with `access_ok` and `verify_read/verify_write` calls. `access_ok` checks whether the user-space pointer is valid to be referenced. `verify_read` is used to read data from user mode. For reading simple variables like `char` and `int`, use `__get_user`.

Now that we have the user-specified parameters, the next step is to create a control socket and establish a connection with the FTP server. `sock_create()` does this for us—its arguments are similar to those we pass to the user-level `socket()` system call. The `struct sockaddr_in` variable `servaddr` is now filled in with all the necessary information—address family, destination port and IP address of the server. Each `socket` structure has a member that is a pointer to a structure of type `struct proto_ops`. This structure contains a list of function pointers to all the operations that can be performed on a socket. We use the `connect()` function of this structure to establish a connection to the server. Our functions `read_response()` and `send_reply()` transfer data between the client and server (these functions are explained later):

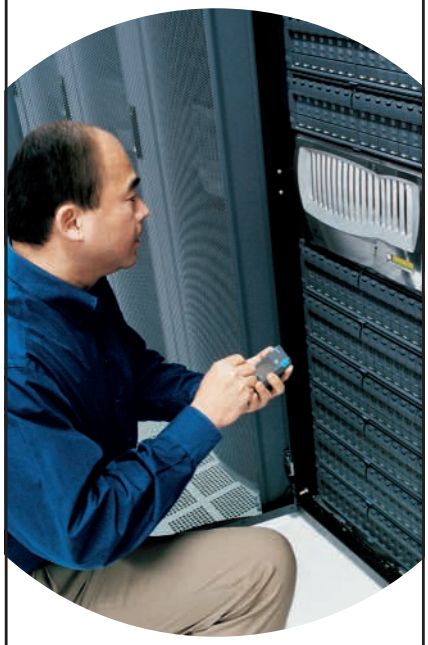
```
r = sock_create(PF_INET, SOCK_STREAM,
                  IPPROTO_TCP, &data);
memset(&claddr,0, sizeof(claddr));
claddr.sin_family = AF_INET;
claddr.sin_port = htons(EPH_PORT);
clddr.sin_addr.s_addr= htonl(
                        create_address(srcip));
r = data->ops->bind(data,
          (struct sockaddr *)&claddr,
          sizeof (claddr));
r = data->ops->listen(data, 1);
```

Now, a data socket is created to transfer data between the client and server. We fill in another `struct sockaddr_in` variable `claddr` with information about the client—protocol family, local unprivileged port that our client would bind to and, of course, the IP address. Next, the socket is bound to the ephemeral port EPH_PORT. The function `listen()` lets the kernel know that this socket can accept incoming connections:

```
a = (char *)&claddr.sin_addr;
p = (char *)&claddr.sin_port;

send_reply(control, reply);
read_response(control, response);

strcpy(reply, "RETR ");
strcat(reply, src);
strcat(reply, "\r\n");

send_reply(control, reply);
read_response(control, response);
```


How do you make high-performance computing even better?

Customize it.



Using Intel®-based servers to maximize power and interoperability, Intel® Premier Providers build high-performance technology solutions customized to your enterprise's needs. As members of the most elite Intel® channel program, Premier Providers deliver the latest IT solutions, and have priority access to parts and technical support.

Insist on the Best.

intel premier PROVIDER 2005

Find the Intel® Premier Provider that's right for you.
www.insistthebest.com

intel®

64

The Intel® Xeon™ processor now provides high availability for your 64-bit applications and clusters.

The Intel® Xeon™ processor now works harder for your business than ever. With innovative features that enable power-saving options, flexible memory, I/O and storage configuration. And, of course, continued support for all your existing 32-bit applications.

How can clusters featuring the Intel Xeon processor serve you?
intel.com/go/xeon

Western Scientific
9445 Farnham Street
San Diego, CA 92123
858-565-6699
www.westernscientific.com/intel

Atipa Technologies
4921 Legends Drive
Lawrence, KS 66049
888-222-7822
www.atipa.com

Bell Computers
187 Pacific Street
Pomona, CA 91768
909-598-1006
www.bellcomputer.com

©2005 Intel Corporation. Intel, the Intel logo, Intel Inside and Intel Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. All rights reserved.

PGI Compilers are building the 64-bit applications infrastructure.

C, C++, F77, F95 and HPF • 32-bit and 64-bit Linux
Optimized for AMD64 and IA32/EM64T • Full 64-bit support
Workstation, Server and Cluster configurations • Fast compile times
Native OpenMP • Native SMP auto-parallelization • Cache tiling
Function inlining • SSE/SSE2 Vectorization • Loop unrolling
Interprocedural optimization • Profile-feedback optimization
Large file support on 32-bit Linux • 64-bit integers and pointers
F77 pointers • Byte-swapping I/O • VAX and IBM extensions
OpenMP/MPI/threads debugging • OpenMP/MPI/threads profiling
Interoperable with g77/gcc/gdb • PDF and printed documentation
Electronic purchase, download and upgrades • Tech support
Network-floating licenses • Academic and volume discounts

Visit www.pgroup.com to download a free PGI evaluation package and see the latest tips and techniques for porting to 64-bit systems.

The Portland Group™
www.pgroup.com ++01 (503) 682-2806

The registered trademarks and marks are the property of their respective owners.

STMicroelectronics

As explained previously, a PORT command is issued to the FTP server to let it know the port for data transfer. This command is sent over the control socket and not over the data socket:

```
new_sock = sock_alloc();
new_sock->type = data->type;
new_sock->ops = data->ops;

r = data->ops->accept(data, new_sock, 0);
new_sock->ops->getname(new_sock,
    (struct sockaddr *)address, &len, 2);
```

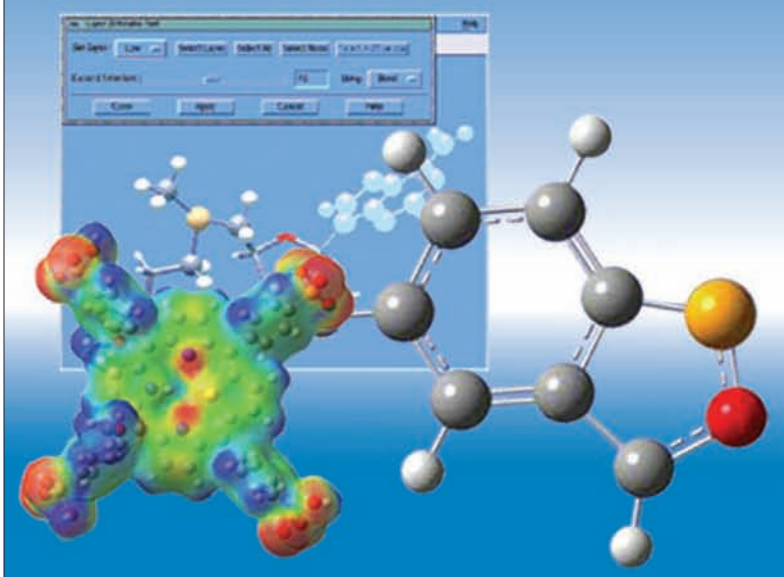
Now, the client is ready to accept data from the server. We create a new socket and assign it the same `type` and `ops` as our data socket. The `accept()` function pulls the first pending connection in the listen queue and creates a new socket with the same connection properties as `data`. The new socket thus created handles all data transfer between the client and server. The `getname()` function gets the address at the other end of the socket. The last three lines in the above segment of code are useful only for printing information about the server:

```
if((total_written = write_to_file(pmk.dst,
             new_sock, response)) < 0)
    goto err3;
```

The function `write_to_file` deals with opening a file in the kernel and writing data from the socket back into the file. Writing to sockets works like this:

```
void send_reply(struct socket *sock, char *str)
{
    send_sync_buf(sock, str, strlen(str),
                  MSG_DONTWAIT);
}


int send_sync_buf
(struct socket *sock, const char *buf,
 const size_t length, unsigned long flags)
{
    struct msghdr msg;
    struct iovec iov;
    int len, written = 0, left = length;
    mm_segment_t oldmm;

    msg.msg_name      = 0;
    msg.msg_namelen   = 0;
    msg.msg_iov       = &iov;
    msg.msg_iovlen    = 1;
    msg.msg_control   = NULL;
    msg.msg_controllen = 0;
    msg.msg_flags     = flags;

    oldmm = get_fs(); set_fs(KERNEL_DS);
```


64-bit GAUSSIAN Compiled With PGI



Gaussian 03 is the premier electronic structure program. Chemists and other scientists use it to study important molecules and reactions related to drug design, materials science, catalysis, and other areas of leading edge and commercial research interest.

See ***www.gaussian.com*** to learn about the latest Gaussian 03 innovations that make it applicable to very large molecules previously out of reach of accurate models.

Gaussian, Inc builds *Gaussian 03* for 64-bit AMD64 and EM64T processor-based systems using ***PGI Compilers and Tools.***

Gaussian

```
repeat_send:
     msg.msg_iov->iov_len = left;
     msg.msg_iov->iov_base = (char *) buf +
                                written;

     len = sock_sendmsg(sock, &msg, left);
     ...
     return written ? written : len;
}
```

The `send_reply()` function calls `send_sync_buf()`, which does the real job of sending the message by calling `sock_sendmsg()`. The function `sock_sendmsg()` takes a pointer to `struct socket`, the message to be sent and the message length. The message is represented by the struture `msghdr`. One of the important members of this structure is `iov` (io vector). The iovector has two members, `iov_base` and `iov_len`:

```
struct iovec
{
     /* Should point to message buffer */
     void *iov_base;
     /* Message length */
     __kernel_size_t iov_len;
};
```

These members are filled with appropriate values, and `sock_sendmsg()` is called to send the message.

The macro `set_fs` is used to set the FS register to point to the kernel data segment. This allows `sock_sendmsg()` to find the data in the kernel data segment instead of the user-space data segment. The macro `get_fs` saves the old value of FS. After a call to `sock_sendmsg()`, the saved value of FS is restored.

Reading from the socket works similarly:

```
int read_response(struct socket *sock, char *str)
{
         ...
         len = sock_recvmsg(sock, &msg,
                  max_size, 0);
         ...
         return len;
}
```

The `read_response()` function is similar to `send_reply()`. After filling the `msghdr` structure appropriately, it uses `sock_recvmsg()` to read data from a socket and returns the number of bytes read.

A User-Space Program

Now, let's take a look at a user-space program that invokes our system call to transfer a file. We explain the relevant details for calling a new system call:

```
...
#define __NR_my_sys_call 223
_syscall1(long long int, my_sys_call,
           struct params *, p);

int main(int argc, char **argv)
{
  struct params pm;
  /* fill pm with appropriate values */
  ...
  r =  my_sys_call(&pm);
  ...
}
```

`#define __NR_my_sys_call 223` assigns a number to our system call. `_syscall1()` is a macro that creates a stub for the system call. It shows the type and number of arguments that our system call expects. With this in place, `my_sys_call` can be invoked just like any other system call. Upon running the program, with correct values for the source and destination files, a file from a remote FTP server is downloaded onto the client machine. Here is a transcript of a sample run:

```
# make
make -C /lib/modules/2.6.9/build SUBDIRS=/home/ppadala/ftp modules
make[1]: Entering directory `/home/ppadala/linux-2.6.9'
  CC [M]  /home/ppadala/ftp/ftp.o
  Building modules, stage 2.
  MODPOST
  CC      /home/ppadala/ftp/ftp.mod.o
  LD [M]  /home/ppadala/ftp/ftp.ko
make[1]: Leaving directory `/home/ppadala/linux-2.6.9'
# gcc do_ftp.c
# ./a.out <local host's IP address> 152.2.210.80 /README /tmp/README anonymous anon@cs.edu
Connection from 152.2.210.80
return = 215 (length of file copied)
```

Conclusions

We have seen a basic implementation of an FTP client within the kernel. This article explains various issues of socket programming in the kernel. Interested readers can follow these ideas to write various network applications, such as an HTTP client or even a Web server in the kernel. Kernel applications, such as the TUX Web server are used for high-performance content serving and are well suited for environments that demand data transfer at high rates. Careful attention has to be paid to the design, implementation and security issues of such applications.

Resources for this article: www.linuxjournal.com/article/8453.

Pradeep Padala is a PhD student at the University of Michigan. His general interests are in distributed systems with specific emphasis on scheduling and fault tolerance. He is the author of the NCurses Programming HOWTO and contributes to various open-source projects. More about him can be found on his Web site at www.eecs.umich.edu/~ppadala.

Ravi Parimi has a Master's degree in Computer Engineering and currently works in Silicon Valley, California. His main interests are in operating systems, networking and Internet security. He has been using Linux since 1998 and aspires to be a kernel hacker. In his free time, he pursues Vedic studies and Chess.

Where Open Minds Meet...

LINUXWORLD® CONFERENCE & EXPO

October 5-6 • 2005 • Olympia 2 • London

The UK's only event for business and the technical community

Come along to LinuxWorld Expo and tackle IT business issues, gain real-time Linux and open source solutions, meet key suppliers, ask technical points and get answers, discover 'how-to', see new technologies, source products and network with the entire community; experts, colleagues and suppliers;

Register NOW at **www.linuxworldexpo.co.uk** for **FREE** entry into the exhibition, featuring;

- The Great Linux Debate
- OSC/Open Source Academy
- FREE Product Briefings and Demonstrations
- FREE Showcases of Technology
- FREE Case Study Presentations
- FREE Open Forum Europe Advice Centre
- .org village
- Internet Café with Wireless Connection
- Pre-registered visitors can take the LPI exam for ONLY £25 – Saving £100

CONFERENCE PROGRAMMES

Technical and Enterprise:
Two streams, two days of informative and in-depth sessions covering key issues and topics for technical and enterprise.

Linux in Business:
Practical applications, benefits and analysis of Linux and open source in business:
SPEAKERS INCLUDE:
Klaus Knopper, *Engineer, Consultant and Software Developer, KNOPPER.NET & Maker, Knoppix*
Mark Shuttleworth, *President, Ubuntu Foundation*
Rasmus Lerdorf, *Engineer, Yahoo! Inc.*
Paul Everitt, *Founder, Zope Europe Association*

Linux for the Technical team:
What's here now; and what's coming next.
SPEAKERS INCLUDE:
Andrew Eddie, *Senior Systems Integrator, Toowoomba City Council, Australia*
Kevin Carmony, *President and CEO, Linspire Inc*
Michael Tiemann, *Vice President Open Source Affairs, Red Hat*
Brian Green, *Director of Solutions Management, Novell*

Delegate places are limited, so book today and also benefit from:
- FREE LPI Examination – Saving £125
- GUARANTEED seating at The Great Linux Debate
- FREE Entry to the exhibition

View the full conference programme and book your place TODAY! - **www.linuxworldexpo.co.uk**

Register NOW at www.linuxworldexpo.co.uk

GOLD SPONSOR
DELL

PLATINUM SPONSORS
Novell suse ORACLE Sun microsystems

OWNED BY
IDG WORLD EXPO

ORGANISED BY
turret GROUP International Business Publications & Exhibitions

All trademarks acknowledged. E&OE. Programme may be subject to change. Correct at time of press.

Trekking through the Desktop Jungle

Is it easier to find a document on a faraway Web server than one on your own hard drive? Try some search programs to dig up the files you need.

BY MARCEL GAGNÉ

That certainly does make it difficult, François. When I asked you to locate the wine order from last month and you told me it was somewhere on your disk, I didn't expect that it was sitting "somewhere on your disk" in quite this way. This is possibly the most disorganized home directory I've ever seen. Every document is in the same folder, and all the files are cryptically named. What were you thinking, *mon ami*?

Quoi? Well, of course there is a way to find it. If the document still exists somewhere on your disk, we'll find it. We just need to use the right tools. Later, though—our guests will be here any moment and...too late, François, they are already here! Welcome, everyone, to *Chez Marcel*, home of fine Linux fare and exquisite wines. Please sit and make yourselves comfortable. François will fetch your wine *immédiatement*. François, head to the east wing of the wine cellar and bring back that 2001 Nuits Saint George Pinot Noir we've been tasting, er, I mean, subjecting to quality control. *Vite!*

That wine, *mes amis*, just happens to represent part of an order lost in one of François' documents on his computer. Trouble is, he doesn't remember which document. What we need to do, is set him up with a desktop search engine. Luckily, this just happens to be the basis of tonight's menu, so we all will profit from my faithful waiter's lack of organization.

The original desktop search engine, *mes amis*, is something that's been around in Linux from the beginning, and that's the find command. This is an amazingly powerful tool and one that is easily overlooked in this age of cutting-edge graphical desktops. In its most basic form, find is used like this:

```
find starting_dir [options]
```

One of those options is -print, which makes sense only if you want to see any kind of output from this command. You easily could get a listing of every file on the system by starting at the top and recursively listing the disk:

```
find / -print
```

Of course, it makes more sense to search for something, for instance, all the MP3-type music files sitting on your disk. Because you know that the files end in a .mp3 extension, you can use that to search:

```
find / -name "*.mp3" -print
```

This is also great for locating big files you haven't looked at in forever. Maybe it's time to do a little archiving of those old files, but how do you find only them? Say you want to look for anything that has not been modified (this is the -mtime parameter) or accessed (the -atime parameter) in the past 12 months. The -o option is the "or" in this equation:

```
find /home/marcel -size +1024 \( -mtime +365 -o -atime +365 \) -ls
```

In case you are curious, the back-slashes in front of the parentheses are escape characters; they are there to make sure the shell does not interpret them in ways you do not want it to—in this case, the open and close parentheses on the second line. The preceding command also searches for files that are greater than 500KB in size. That is what the `-size +1024` means, because 1024 refers to 512-byte blocks. The `-ls` at the end of the command tells the system to do a long listing of any files it finds that fit the search criteria. So far so good?

The find command is fairly simple to use on the surface, but it also has many command-line options and (as you can see) interesting ways of passing the results of a search to other commands, so that the results can be narrowed down or fine-tuned. Getting to know find is a great idea, but there are alternatives that are a little friendlier.

Many people out there have grown up in the graphical world of KDE or GNOME, so desktop tools have been created in each of these environments. Even so, my experience indicates that these excellent tools are, for many users, as equally overlooked as find. Let's have a look at those now.

Let's begin our search for search tools under KDE. Click the application launcher and look for a submenu labeled Find. The Find menu has two options, one for files and one for Web search (which, by default, launches Konqueror on the Google Web site). You also can fire up the files search tool by using the Alt-F2 quick launch (program name: `kfind`). When the application starts, the Find Files/Folders dialog appears. It contains three different tabs, and each is designed to help you locate the information you need. They are labeled Name/Location, Contents and Properties.

Under the Name/Location tab, specify the starting folder, either by entering it manually or by clicking the Browser button and navigating over to it using the KDE file navigator. There's also a field labeled Named where you enter part of a filename using Linux metacharacters. For instance, if I wanted to find all the files with Cooking anywhere in the title, I would enter `*cooking*`. By default, this is a case-insensitive search, so upper- and lowercase don't matter in terms of the search results. You can, however, override this behavior by clicking the Case-sensitive search check box.

Under the Contents tab, the real action takes place. Generally speaking, I don't have a problem locating a file by name. It's the content that is the real issue. Which of your several hundred documents contains a reference to a particular

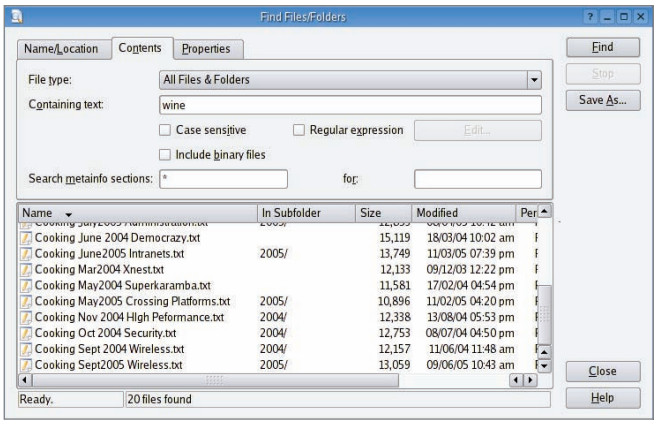


Figure 1. KFind makes it easy for Marcel to locate all those columns that mention "wine".

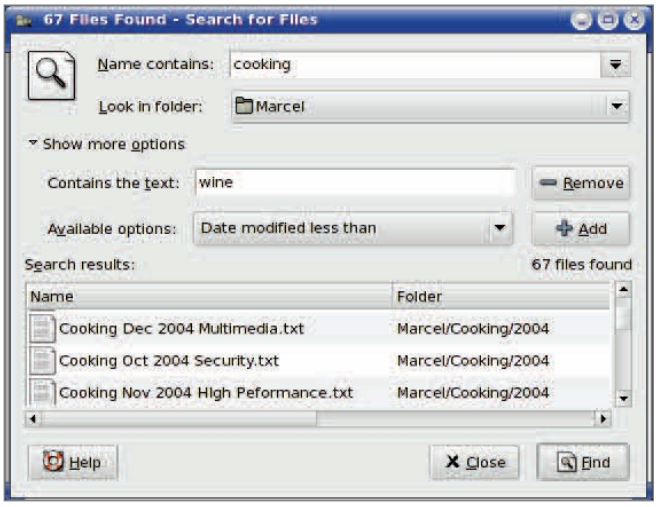


Figure 2. The GNOME search tool allows you to search by name as well as text within a file.

word or phrase is a more difficult search than which has a particular word in the name. The Contents tab lets you enter your search text (again, case-insensitive by default), regular expression searches and so on. You even can specify that Kfind search through binary files and not only documents (Figure 1). There's also a meta-info search feature for things like MP3 files that contain embedded information, such as title and artist.

Finally, the Properties tab provides a means of searching for files or folders based on creation or modification date, ownership and more.

Similarly, GNOME users have access to the GNOME search tool (program name: `gnome-search-tool`), a similar program that lets you search based on filename, file content (text search) and date. Choose Search for files in the GNOME Places menu (I'm running 2.10 in this example), and this brings up the file find dialog (Figure 2).

When the dialog first appears, there isn't much to see. The defaults are to search for a file by name, which you enter in the Name contains field. Below that is your starting folder for the search, the default being your home directory. To get the full power of the GNOME search tool, click on the arrow next to the label that says Show more options. A new field appears through which you can specify some text in the file itself.

Finally, directly below the text search field, is one other option that can be quite complex. A drop-down box labeled Available options includes size, date and ownership search criteria that can be applied to narrow down your search results even further.

If you've been following search technology in any way, you'll know that there's a lot of excitement concerning desktop search engines these days—think Google for your desktop. In fact, Google does provide such a tool, but alas, only for non-Linux operating systems. However, this is not to say that desktop search tools don't exist for Linux.

One such tool is Roberto Cappuccio's Kat, a desktop search engine and indexing tool that makes it easy and fast to do full-text searches in a variety of document formats (for example, PDF, OpenOffice.org, KWord and so on). You also can search for images using thumbnails and more.

The Kat Web site (see the on-line Resources) provides binary packages for a number of distributions, so you may not need to build from source. Should you need to, however, the process is nothing more than the classic extract-and-build five-step. In terms of prerequisites, you need the SQLite database and its development libraries.

To use Kat, simply start the program (name: `kat`) and a plain three-pane window appears where you will do your work

We've got problems with your name on them.

At Google, we process the world's information and make it accessible to the world's population. As you might imagine, this task poses considerable challenges. Maybe you can help.

We're looking for experienced software engineers with superb design and implementation skills and expertise in the following areas:

- high-performance distributed systems
- operating systems
- data mining
- information retrieval
- machine learning
- and/or related areas

If you have a proven track record based on cutting-edge research and/or large-scale systems development in these areas, we have brain-bursting projects with your name on them in Mountain View, Santa Monica, New York, Bangalore, Hyderabad, Zurich and Tokyo.

Ready for the challenge of a lifetime? Visit us at http://www.google.com/lj for information. EOE

Google™

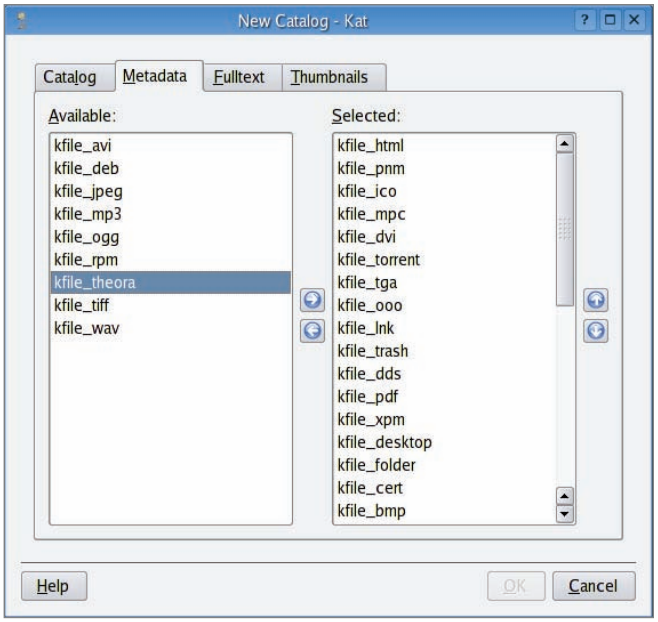


Figure 3. Using kfile hooks, Kat can index almost anything.

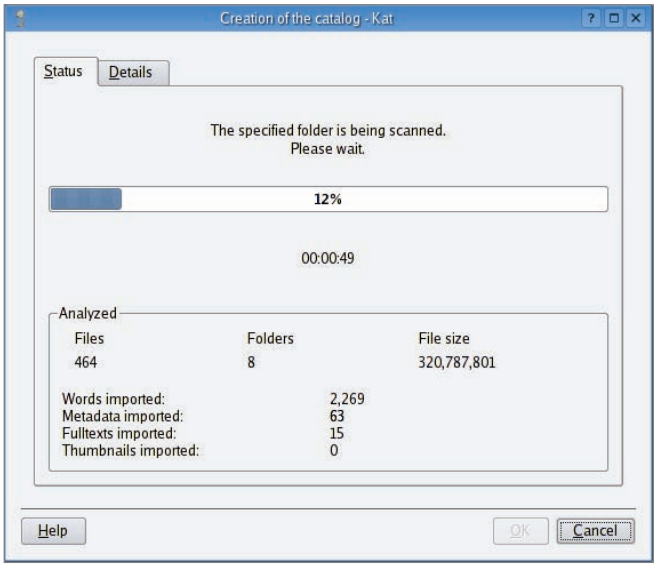


Figure 4. As Kat creates the new catalog, the program reports statistics on the process.

and your searching. The first step is to create a catalog. To do this, click File on the menu bar and select New.

When creating a new catalog, a four-tabbed window appears. The first tab, labeled Catalog, is where you enter the starting directory, the name of the catalog and other identifying information. On the second tab, labeled Metadata, you'll see a list of the various metadata engines that are available to Kat for indexing. You can remove different formats, but most likely, this will stay as is (Figure 3). Similarly, the Fulltext tab. Under Thumbnails, you can select the size of the thunmbnails created during the index process.

A status window keeps you abreast of the number of files and folders scanned, as well as the size of the collection (Figure 4).

This brings us to the one big drawback of a tool like this. If

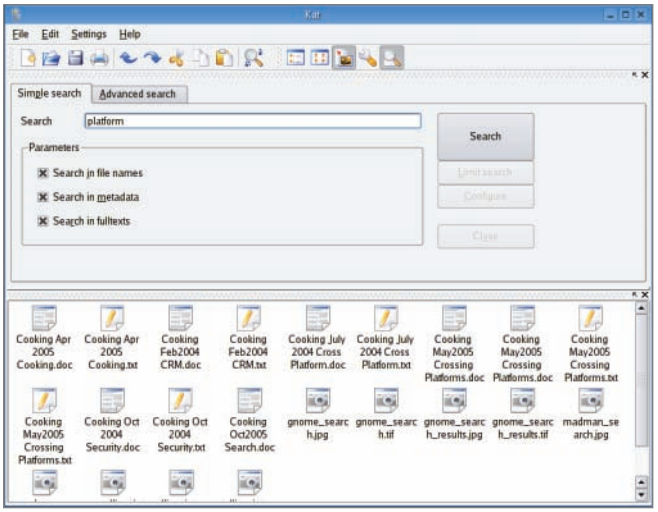


Figure 5. Although the initial indexing can take some time, Kat searches are blazingly fast.

the folder for which you are creating a catalog is large, this can take an amazing amount of time. Be prepared or keep your catalogs confined to a reasonable collection of files. I tried to index my own home directory in its entirety at nearly 6.6GB of data—suffice it to say, that was a mistake.

Once a catalog has been created, finding information is blazingly fast. Simply click on the search icon on the far right (the magnifying glass), enter your search term and Kat returns the results of the search almost instantly (Figure 5).

According to the clock on the wall, it would appear, *mes amis*, that closing time has arrived. Before we leave this topic of desktop search engines, I'd like to mention another package with the friendly, puppy-dog name of Beagle. Beagle is built on Mono (the open-source .Net implementation) and requires an inotify-enabled kernel. Neither is uncommon in the more modern distributions. Beagle also shows promise in that it is very fast and works silently in the background, keeping an eye on what you tell it while automatically updating its catalog of information. Unfortunately, Beagle is very much alpha code and not quite ready for prime time, as they say (although it is included with the new SUSE Linux Professional 9.3). Nevertheless, Beagle is a tool to watch, and I've included the link in the on-line Resources.

Please raise your glasses, *mes amis*, and let us all drink to one another's health. *A votre santé! Bon appétit!*

Resources for this article: www.linuxjournal.com/article/8456.

Marcel Gagné is an award-winning writer living in Mississauga, Ontario. He is the author of *Moving to the Linux Business Desktop* (ISBN 0-131-42192-1), his third book from Addison-Wesley. He also makes regular television appearances as Call for Help's Linux guy. Marcel also is a pilot and a past Top-40 disc jockey. He writes science fiction and fantasy and folds a mean Origami T-Rex. He can be reached via e-mail at mggagne@salmar.com. You can discover a lot of other things (including great Wine links) from his Web site at www.marcelgagne.com.

Best Price. Best Performance. Best Support.

Evolution of the Revolution

The TransPort™ is ImageStream's best-selling router. With its small footprint, business-class features, and competitive price, the TransPort is an ideal router for T1 and E1 applications that demand low latency wire-speed performance.

The TransPort includes 128 MB RAM, three 10/100 ethernet ports, and an expansion slot for add-on cards. It also features the ImageStream Linux™ router distribution, which supports most WAN protocols and advanced features including NAT firewall, peer-to-peer traffic control, bridging, bandwidth limiting, QoS, dynamic routing, VPN, and more.

NO HASSLE 31 DAY RETURN GUARANTEE

Like all ImageStream routers, the TransPort includes 12 months of free 24/7 technical support, a full 12-month warranty on parts and labor, free software upgrades for life, and the industry's only money back performance guarantee.

Enterprise Router™

The Enterprise features four 8 Gbps buses, dual CPUs, and redundant AC or DC power supplies. It can route multiple OC3 or OC12 circuits at wire speed.

Gateway Router™

ImageStream's Gateway is the industry's lowest cost OC3 router. The upgraded dual bus Gateway 64™ can route multiple DS3/E3 or OC3 circuits at wire speed.

Rebel Router™

The Rebel is the industry's lowest cost DS3/E3 router. The 1U Rebel can route one DS3/E3 or up to 16 T1/E1 circuits at wire speed.

R1™ Router

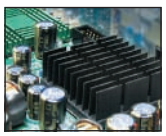
The R1 is designed for extended temperature applications including outdoor installations. The R1 has a small footprint, and can be installed in set-top, wall-mount, and rackmount applications.

Well Connected

The TransPort provides three 10/100 ethernet ports for flexible LAN connectivity.

Fanless CPU

The TransPort's highly efficient CPU minimizes power consumption and cooling requirements.



Easy Indicators

The TransPort provides front panel LEDs to show LAN connection status.

Sized Right

The TransPort's small footprint allows it to be installed just about anywhere.

- **Best Software** The ImageStream Linux™ router distribution has everything you need to deploy advanced network applications including secure shell, menu-based configuration, real-time monitoring, IP firewall, bridging, interface bonding, dynamic routing, QoS, IPsec VPN, free software upgrades for life, and more.
- **Best Support** ImageStream routers include a full year of free 24/7 support. When you consider total cost of ownership, ImageStream routers cost less up front and over time.
- **Guaranteed Performance** Our 31-day performance guarantee ensures your router will function as promised. If your ImageStream router does not work as specified in writing, and our support team cannot correct the problem, you can return your router for a full refund of the purchase price. See our web site for details.

ImageStream™
Routers for the Real World™

800.813.5123
www.imagestream.com
1.574.935.8484

*ImageStream, Enterprise Router, Gateway Router, Gateway 64, Rebel Router, R1, TransPort, ImageStream Linux and "Routers for the Real World" are trademarks of ImageStream Internet Solutions, Inc. Linux is a registered trademark of Linus Torvalds. Specifications are subject to change without prior notice. *Please refer to ImageStream's Web site for more information on wire-speed specifications and the performance guarantee.*

Limitations of shc, a Shell Encryption Utility

The shell script compiler, shc, obfuscates shell scripts with encryption—but the password is in the encrypted file. Could an intruder recover the original script using objdump? BY NALNEESH GUAR

Shc is a popular tool for protecting shell scripts that contain sensitive information such as passwords. Its popularity was driven partly by auditors' concern over passwords in scripts. shc encrypts shell scripts using RC4, makes an executable binary out of the shell script and runs it as a normal shell script. Although the resulting binary contains the encryption password and the encrypted shell script, it is hidden from casual view.

At first, I was intrigued by the shc utility (www.datsi.fi.upm.es/~frosal/sources/shc.html) and considered it as a valuable tool in maintaining security of sensitive shell scripts. However, upon further inspection, I was able to extract the original shell script from the shc-generated executable for version 3.7. Because the encryption key is stored in the binary executable, it is possible for anyone with read access to the executable to recover the original shell script. This article details the process of extracting the original shell executable from the binary generated by shc.

shc Overview

shc is a generic shell script compiler. Fundamentally, shc takes as its input a shell script, converts it to a C program and runs the compiler to compile the C code. The C program contains the original script encrypted by an arbitrary key using RC4 encryption. RC4 is a stream cipher designed in RSA laboratories by Ron Rivest in 1987. This cipher is used widely in commercial applications, including Oracle SQL and SSL. Listing 1 demonstrates running shc.

The two new files, named with the .x and .x.c extensions to the name of the source shell script, are the executable and an intermediate C version. Upon executing pub.sh.x, the original shell source is executed. shc also specifies a relax option, -r. The relax option is used to make the executable portable. Basically, shc uses the contents of the shell interpreter itself, such as /bin/sh, as a key. If the shell binary were to change, for example, due to system patching or by moving the binary to another system, the shc-generated binary does not decrypt or execute.

Listing 1. Running shc

```
[user1@shiraz test]# cat pub.sh
#!/bin/sh
echo "Hello World"
user1@shiraz test]# ./pub.sh
Hello World
[user1@shiraz test]# shc -v -r -f pub.sh
shc shll=sh
shc [-i]=-c
shc [-x]=exec '%s' "$@"
shc [-l]=
shc opts=
shc: cc  pub.sh.x.c -o pub.sh.x
shc: strip pub.sh.x
[user1@shiraz test]# ls
pub.sh pub.sh.x pub.sh.x.c
[user1@shiraz test]# ./pub.sh.x
Hello World
```

I inspected the shell executable using strings and found no evidence of the original shell script. I also inspected the intermediate C source code and noted that it stores the shell script in encrypted octal characters, as depicted in Listing 2.

Listing 2. The original shell script becomes an RC4-encrypted string in the C version.

```
static  char text[] =
    "\223\004\215\264\102\216\322\060\300\070\101\217\277\161\033\130"
    "\217\145\370\170\106\257\176\301\057\132\172\044\217\247\276\222"
    "\203\076\334\201\323\107\064\334\120\132\001\241\267\052\203\216"
    "\116\232\156\337\121\145\235\003\156\244\142\246\117\200\206\014"
    "\004\153\372\152\030\262\171\275\137\342\247\367\231\315\353\151"
    "\264\241\230\105\344\053\034\247\342\142\156\305\327\255\036\111"
    "\234\061\013\355\300\336\324\257\175\124\222\044\132\040\276\067"
    "\007\002\371\063\021\320\060";
```

The C source code also includes as arrays the password as well as other encrypted strings. Therefore, anyone with access to the source code easily can decrypt and view the contents of the original shell script. But what about the original shell binary executable generated by shc? Is it possible to extract the original shell script from nothing but the binary executable? The answer to this question is explored in the next section.

Extraction Approach

I generated and reviewed the C source code for several shell scripts to better understand how the shell source is encrypted and decrypted. Fundamentally, shc uses an implementation of RC4 that was posted to a Usenet newsgroup on September 13, 1994. I set off by first identifying the encryp-

tion key and the encryption text. The objdump utility came in handy for this. bjdump, part of GNU binutils, displays information about object files. First, we use objdump to retrieve all static variables, for this is where the encryption key and the encrypted shell text are stored. Listing 3 provides a brief overview of objdump.

The first column of the output in listing 3 specifies the starting addresses in hexadecimal, followed by the stored data in the next four columns. The last column represents the stored data in printable characters. So somewhere in the first four columns of the output is the array of characters that form the encryption key (password) and the encrypted shell script.

Comparing the original C source code and Listing 3, you can see that the password most likely begins at address 0x804a540. After comparing other executables, I determined that the first address after the zeros leading the "Please contact your provider" text usually is the starting address. To retrieve these arrays, such as the one depicted in Listing 2, we also need to look at the disassembled code. We use objdump again here, except this time with the -d option, for disassemble, as shown in Listing 4.

The last two columns represent assembly instructions. The movl instruction is used to move data—`movl Source, Dest`. The Source and Dest are prefixed with $ when referencing a C constant. The push takes a single operand, the data source, and stores it at the top of stack.

Now that we have the basics of objdump, we can proceed to extract the encryption password and eventually the shell code.

In the intermediate C code produced by shc, about nine arrays are referenced by the variables pswd, shll, inlo, xecc, lsto, chk1, opts, txt and chk2. The pswd variable stores the encryption key, and the txt variable stores the encrypted shell text. shc hides the useful information as smaller arrays within these variables. Thus, obtaining the actual array involves two steps. First, identify the length of the array. Second, identify the starting address of the array.

The objdump output needs to be looked at in detail to obtain the actual

C-TREE PLUS® DATABASE TECHNOLOGY OPENS UP YOUR OPTIONS

SQL offers a convenient and easy-to-use database interface. ISAM provides powerful performance with precision indexing control in a small footprint. With c-tree Plus you can simultaneously enjoy BOTH! Superior ISAM indexing technology PLUS an industry-standard SQL interface provide blazing fast data management for every environment. Break the limitations of a single solution and open up your database options. Experience the benefits c-tree Plus can deliver to your application!

FEATURED HIGHLIGHTS

- Fast, reliable, and portable
- Low deployment cost
- No DBA required
- Professional technical support
- Source code
- 64-bit support
- 16-exabyte file support
- Memory files
- Embeddable database
- Full OLTP support

CUSTOMER TESTIMONIAL

"We have reviewed Oracle and some of the other big relational databases and chose FairCom for our database development needs. With c-tree Plus, we see transactional volume that is 8 to 10 times faster than what we can get with other databases. I have been using c-tree based solutions since the 80's and highly recommend it…"

Visit our Web site for more testimonials about c-tree!

FairCom®
Database your way.

See for yourself — download c-tree Plus® Today!

Go to www.faircom.com/go/open for a FREE evaluation of c-tree Plus!

Other company and product names are registered trademarks or trademarks of their respective owners. © 2005 FairCom Corporation

Listing 3. objdump browses the object file for interesting-looking strings.

```
/usr/bin/objdump --section=.data -s pub.sh.x

pub.sh.x:     file format elf32-i386

Contents of section .data:
 804a4e0 00000000 00000000 3ca80408 00000000  ........<.......
 804a4f0 00000000 00000000 00000000 00000000  ................
 804a500 00000000 506c6561 73652063 6f6e7461  ....Please conta
 804a510 63742079 6f757220 70726f76 69646572  ct your provider
 804a520 00000000 01000000 00000000 00000000  ................
 804a530 00000000 00000000 00000000 00000000  ................
 804a540 e554f49f 93dcd6dc bb0bdc9b ad60edd0  .T...........`..
 804a550 7a9beb67 60277cb2 dd9e0886 0797aeec  z..g`'|.........
 804a560 eba28b7e 7e615a3a 6d37d51a 97c2ea11  ...~~aZ:m7......

...
```

Listing 4. The output of `objdump -d pub.sh.x` shows information needed to find the encrypted script. Lines in parentheses were added.

```
8048e52: 68 28 01 00 00  push   $0x128
                         (Length of encryption key)
8048e57: 68 40 a5 04 08  push   $0x804a540
                         (Key address)
8048e5c: e8 17 fb ff ff  call   0x8048978
8048e61: 83 c4 10        add    $0x10,%esp
8048e64: 83 ec 08        sub    $0x8,%esp
8048e67: 6a 08           push   $0x8
                         (Length of shll)
8048e69: 68 72 a6 04 08  push   $0x804a672
                         (shll address)
8048e6e: e8 a0 fb ff ff  call   0x8048a13
8048e73: 83 c4 10        add    $0x10,%esp
8048e76: 83 ec 08        sub    $0x8,%esp
8048e79: 6a 03           push   $0x3
                         (length of inlo)
8048e7b: 68 8a a6 04 08  push   $0x
8048e80: e8 8e fb ff ff  call   0x8048a13
```

array length and the starting address. My first hint here is to look for all addresses that are within the data section (Listing 2) of the disassembled object code. Next, seek out all the push and mov commands in Listing 4. Addresses will be different for different scripts, but when you encrypt a few scripts and read the resulting C code, the patterns become familiar.

The 804a540 address seems to correspond to the pswd variable, the encryption key. The length of the useful portion of the encryption key is represented by 0x128, or 296 in decimal form. Similarly, the next variables, shll and inlo, have useful lengths of 0x8 and 0x3 and starting addresses of 804a672 and 804a68a, respectively. This way, we are able to obtain the starting addresses and lengths of all nine variables. Next, we need to be able to decrypt the original shell script using only the binary as input.

In shc, before the shell script itself is encrypted, many other pieces of information are encrypted. Furthermore, the RC4 implementation maintains state between encrypting and decrypting each individual piece of information. This means that the order in which shc encrypts and decrypts information must be maintained. Failure to do so results in illegible text. To extract the original shell script, we need to perform several decryptions. For this step, I wrote a small program called deshc, using the existing code from one of the intermediate C files. The program reads two files as its input, the binary executable and an input file that specifies the array lengths and addresses. deshc executes the following four steps:

- Reads binary executable.
- Extracts data section from the disassembled output.
- Retrieves individual arrays based on input file.
- Decrypts individual arrays in order, so that the RC4 state is maintained.

Based on the objdump output, I have arrived at the following array lengths and addresses for the pub.sh.x executable:

```
pswd 0x128 0x804a540
shll 0x8   0x804a672
inlo 0x3   0x804a68a
xecc 0xf   0x804a68e
lsto 0x1   0x804a6a4
chk1 0xf   0x804a6a6
opts 0x1   0x804a6be
txt  0x76  0x804a6e0
```

All of these parameters are used in an input file to deshc, which then decrypts and prints the original shell script.

Conclusion

An approach to extract the shell source code successfully from shc version 3.7 generated binary executable was demonstrated. The pub.sh script was used for illustrative purposes only. I have indeed tested the deshc program on executables that I did not create and without access to the source code or the original shell script.

Francisco Garcìa, the author of shc, recently released version 3.8. It uses somewhat different data structures and improves upon the security of the previous version. Nevertheless, I believe that embedding the encryption password within the binary executable is dangerous and prone to extraction as discussed in this article.

Nalneesh Gaur, CISSP, ISAAP, works at Diamond Cluster International as a BS7799 Lead Auditor.

servers ▶ DIRECT™
More products, better service, guaranteed.

GO STRAIGHT TO THE SOURCE!

Introducing the Servers Direct Blade System with the power of Intel® Xeon™ Processor

Increased computing power in a smaller footprint and simplified maintenance help you expand your enterprise solution to meet the most intense application demands.

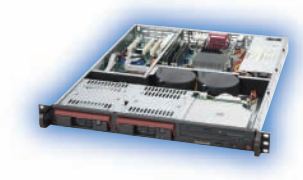
SDB-1100H Servers Direct Blade System

Featuring a Server Direct Server Compute Blade powered by dual ® Xeon™ 800FSB Processors.

Benefit of using Servers Direct Blade System:

- More power, bandwidth, and processing performance to meet the demanding requirements of departmental workloads
- Deliver world-class performance for peak server workloads
- Future 64-bit-enabled applications
- High performance small form factor SCSI hard drives (RAID 1 with 2 HDDs)
- Ethernet I/O for demanding, data-intensive applications
- Ability to easily add hotswappable SCSI hard drives, and additional Ethernet, or Fibre Channel I/O for increased application performance (RAID 1E requires use of the HDDs on the SBX82 as well as the HDDs on the SBESCSI)

1U Xeon Entry Level Server
SDR-1300T



Highest performing with Dual Xeon 800MHz. Excellent with general purpose applications and provide the most power.

- Intel Xeon Processor 2.8Ghz with 800FSB 1MB Cache (Dual Processor Option)
- Intel Extended Memory 64 Technology
- 1U Chassis with 420W power supply
- Supermicro server board with Intel® E7320 (Lindenhurst VS) Chipset
- Kingston 512MB DDR333 ECC Reg. RAM (2x256MB)
- 1pc x Seagate 80GB SATA 7200RPM hard drive
- 2 x 1" Hot-swap SATA drive bays
- Integrated ATI Rage XL SVGA PCI video controller
- 2x Intel® 82541GI Gigabit Ethernet Controllers
- 2x SATA Ports via 6300ESB SATA Controller RAID 0, 1 Supported

$999

2U Xeon Processing Server
SDR-2103T

High-density 2U platform optimized for performance and flexibility; ideal for Web hosting, data center, terminal services and High Performance Computing (HPC)

- Intel Xeon Processor 3.0Ghz with 800FSB 1MB Cache (Dual Processor Option)
- Intel Extended Memory 64 Technology
- 1U Intel Chassis with 700W PFC power supply
- Intel® Server Board SE7520JR2
- Kingston 512MB DDR333 ECC Reg. RAM (2x256MB)
- 6pcs xSeagate 160GB SATA/150 W/ncq 7200rpm 8MB Cache
- Intel SRCS16 6Channel SATA RAID Controller Card
- Integrated ATI Rage XL SVGA PCI video controller
- 2x Intel® PRO/1000 MT Server Network Connections (Intel® 82546GB controller)

$3,499

5U File Server
SDR-5301S

Outstanding performance, excellent data protection, and advanced management for departmental servers.

- Intel Xeon Processor 3.0Ghz with 800FSB 1MB Cache (Dual Processor Option)
- Intel Extended Memory 64 Technology
- Intel SC5300LX Chassis with Redundant 730W Power Supply
- Intel server board w/Intel® E7520 (Lindenhurst) Chipset
- Kingston 1024MB DDR400 ECC Reg. RAM (2x512MB)
- Adaptec 2200S SCSI RAID Controller Card
- Include 6-Drive SCSI Hot-Swap Cage Kit
- 6 x Seagate 36GB SCSI 10K RPM U320 SCA hard drive
- ATI Rage XL SVGA PCI video controller with 8MB of video memory
- Dual Intel® PRO/1000 Server Network Connections

$4,999

3U Clusterable SATA SAN Nodes
SDR-3303T

Provides a flexible, price and performance advantages to the storage needs of Small to Medium Business (SMB) market segment.

- Intel 3U SAN Storage Enclosure w/700W Redundant Power Supply
- Integrated Intel SE7501HG2 Server Board
- 2x Integrated SAN Intel Xeon 3.06Ghz/533FSB Processor
- Integrated SAN 2x256MB Compact Flash Memory Cards w/SAN Mgt Software
- 3 X Integrated Intel SRCS16 6-Channel SATA RAID Cards
- Kingston 1024MB DDR266 ECC Reg. RAM (2x512MB)
- 16pcs x Western Digital WD2500SD RAID SATA
- Integrated SAN Management Software & Storage System

$13,599

Big business power, small business price tag.

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**

Intel, Intel Inside, the Intel Inside logo, and Intel Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Development of a User-Space Application for an HID Device, Using libhid

When it's time to get a new device working on Linux, every piece of information helps, whether it's reading the hardware documentation, snooping data, reading sample code or even running utilities on a non-Linux OS. **BY EOIN VERLING**

The Matrix is a USB bill validator, sometimes known as a note reader or bill acceptor, made by Validation Technologies International. The bundled software was developed for Microsoft Windows, but fortunately the device comes with low-level technical documentation that defines device-specific aspects, such as flow control, status bytes and local status LEDs.

The device is a Human Interface Device (HID), as identified by an enumeration process upon connection. The Windows device manager reports the device as such, as does usbfs on Linux. This article is specific to this particular HID device, so including all of its code probably is unnecessary, but it should provide help for developing other HID-class devices.

After some initial research, I decided to develop user-space code using an in-development library called libhid, which provides a cross-platform way to access and interact with USB HID devices. libhid is implemented on top of libusb, so it does not depend directly on the kernel's USB support.

Another option for driving the Matrix is to use libusb directly, but doing so would be re-inventing the libhid wheel. A third option is to implement the Matrix driver as a kernel module, but it would incur the large overhead of learning kernel particulars. This option also would render the code platform-specific.

Investigation

USB devices are categorized into device classes. A modem is in the communications class, and a speaker falls into the audio class. The HID class mainly consists of devices that people use to control computers. Examples of HID devices are mice, joysticks and force-feedback game controllers. Also included in the HID class are devices that may not require human interaction but do provide data in a similar format to HID-class devices, such as bar-code readers and, in my case, the Matrix note reader.

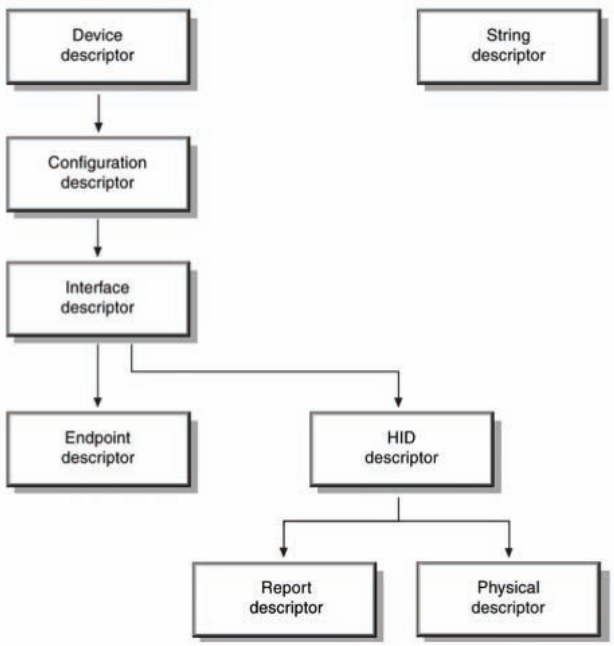


Figure 1. A USB device's descriptors, stored in its ROM, hold information about it.

Information about a USB device is stored in segments of its ROM called descriptors. A diagram of the descriptor structure is provided in Figure 1, where an overall view of the hierarchy can be seen. When a USB device is attached to a USB bus, an enumeration process takes place that equates to the descriptors on the device being read into memory. Information about an HID-class device is contained in its HID report descriptors.

I plugged the device in to the Linux box in order to read the descriptors and monitor the device, the machine and the communications. I did this to try to get as much information as possible so I could have a better understanding of how to write code for the device.

A key component of these report descriptors is the usage information, which is defined in the USB HID Usage Tables (see the on-line Resources). Usage values describe three basic types of information about the device:

- Controls—information about the state of the device, such as on/off or enable/disable.
- Data—all other information that passes between the device and the host.
- Collections—groups of related controls and data.

Taken together, the usage page and usage number define a unique constant that describes a particular type of device or part of that device. For example, on the Generic Desktop usage page (page number 0x01), usage number 0x05 is a game pad, and usage number 0x39 is a hat switch.

Because my device is unique—it isn't a mouse, joystick or something commonly found in the examples of HID-class devices—the usage page is set to 65,440, which is a vendor-defined value. In comparing outputs of lsusb for other HID-class devices, they all had a defined usage page, such as Generic Desktop Controls or Game Controls. Because libhid still is in development, few previous examples of code are available to browse for reference. My work was much like an exploratory investigation.

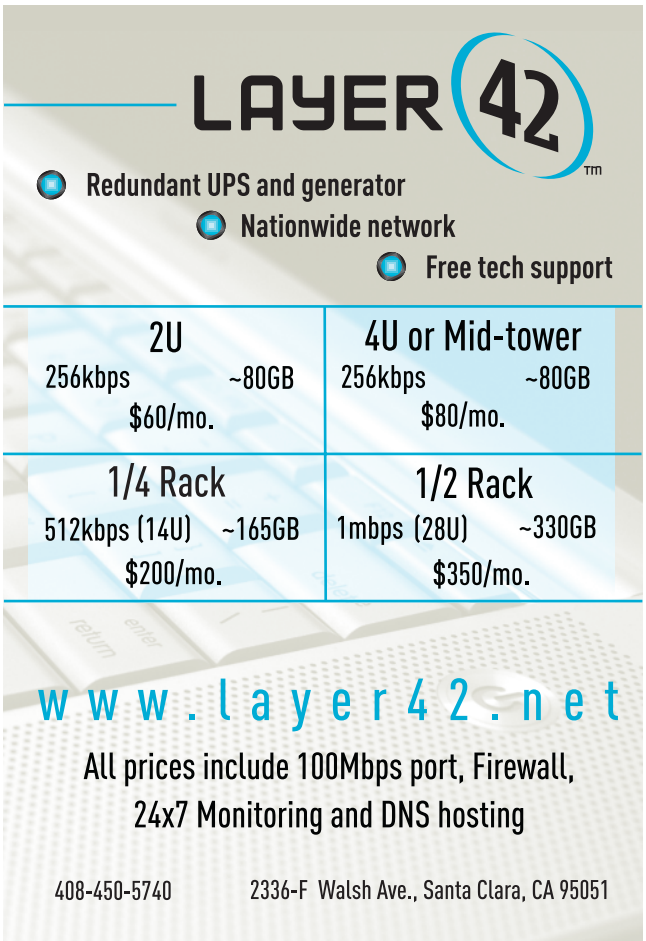
On Linux, with a standard Debian 2.6.9 kernel and usbutils, I was able to see that Linux recognises the device as a USB HID device, bInterfaceClass = HID, and loads the hiddev kernel module. This module, or piece of kernel code, is a generic driver for HID devices. It is not specific to our needs—it mainly is used for mice, joysticks and the like—so it needs to be detached from the device or disabled (see the Communicating with the Device section).

The device, like all USB devices, is enumerated upon connection to the USB bus. So looking at the output of `lsusb -vvv`, run as root, for more information is helpful in determining what the device capabilities are. lsusb parses the usbfs filesystem into a more readable format:

```
[sample lsusb -vvv]

Bus 001 Device 004: ID 0ce5:0003
Device Descriptor:
...
 idVendor            0x0ce5
 idProduct           0x0003
  ...
  Configuration Descriptor:
  ...
   Interface Descriptor:
   ...
     bNumEndpoints         1
     bInterfaceClass       3 Human Interface Devices
     bInterfaceSubClass    0 No Subclass
     bInterfaceProtocol    0 None
     ...
     HID Device Descriptor:
     ...
      Report Descriptor: (length is 32)
       Item(Global):Usage Page,data=[0xa0 0xff]65440
                              (null)
       Item(Local ):Usage, data= [ 0x01 ] 1
                              (null)
       Item(Main  ):Collection, data= [ 0x01 ] 1
                              Application

       Item(Local ):Usage, data= [ 0x03 ] 3
                              (null)
       Item(Global):Logical Minimum,data=[ 0x00 ] 0
       Item(Global):Logical Maximum,data=[ 0xff ]255
       Item(Global): Report Size, data= [ 0x08 ] 8
       Item(Global): Report Count, data= [ 0x05 ] 5
       Item(Main  ): Input, data= [ 0x02 ] 2
        Data Variable Absolute No_Wrap Linear
        Preferred_State No_Null_Position
        Non_Volatile Bitfield
       Item(Local ): Usage, data= [ 0x05 ] 5
                              (null)
       Item(Global):Logical Minimum,data=[ 0x00 ]0
       Item(Global):Logical Maximum,data=[ 0xff ]255
       Item(Global): Report Size, data= [ 0x08 ] 8
       Item(Global): Report Count, data= [ 0x05 ] 5
       Item(Main  ): Output, data= [ 0x02 ] 2
        Data Variable Absolute No_Wrap Linear
        Preferred_State No_Null_Position
        Non_Volatile Bitfield
       Item(Main  ): End Collection, data=none
```

The above output—some of the information has been omitted—follows the hierarchy depicted in Figure 1. Some values of note are:

- idVendor and idProduct—unique identifiers for all USB devices, used for identifying and accessing the device in code.
- bNumEndpoints—lists the number of endpoints available in a device. This value actually means the number of endpoints in addition to the default endpoint, endpoint 0, available in every USB device.



- bInterfaceClass—the value that determines that a device is an HID-class device.
- bInterfaceSubClass—the subclass of a device, in this case, HID. For example, the boot interface subclass of the device must be bootable or available to the BIOS, such as a mouse or keyboard.
- bInterfaceProtocol—the protocol used. Possible values are 0 for none, 1 for keyboard or 2 for mouse; additional information is available in the HID spec.

Communicating with the Device

A block diagram depicting the flow of control of data is shown in Figure 2. It may help in picturing where your code fits in with respect to the libraries and the device. From my investigation, I know that control messages periodically are written by way of the control pipe, and interrupt reads are made through endpoint 0.

The control pipe is used for three tasks: receiving and responding to requests for USB control and class data; transmitting data when polled by the HID class driver, using the Get_Report request; and receiving data from the host. The Interrupt pipe is used for two tasks: receiving asynchronous, or unrequested, data from the device and transmitting low-latency data to the device.

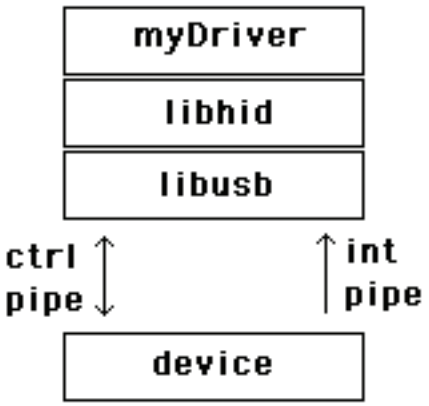


Figure 2. The new driver uses libhid, which depends on libusb.

The kernel has a DEBUG feature that can be activated in order to log extra information about what is happening when communicating with the device. To do this, a file in the kernel source needs to be modified. In the /usr/src/linux/drivers/usr/input/hid-core.c file, these two lines need to be changed from:

```
#undef DEBUG
#undef DEBUG_DATA
```

to:

```
#define DEBUG
#define DEBUG_DATA
```

The module needs to be recompiled and installed. Once this is done, the modules should prove helpful in determining whether your code is working and doing what you expect.

Sample code containing some helpful comments comes with libhid. The file test_libhid.c in the libhid/test directory is a good place to start writing code for the device. Below is a snippet of that code, along with some more explanation of the functions; details are omitted for brevity:

```
HIDInterface* hid;
hid_return ret;

HIDInterfaceMatcher matcher =
         { 0x0ce5, 0x0003, NULL, NULL, 0 };
ret = hid_force_open(hid, 0, &matcher, 3);

int const PATH_LEN = 2;
int const PATH_IN[2] = { 0xffa00001, 0xffa00003 };

int const WRITE_PACKET_LEN = 8;
char write_packet[8] =
         { 0x04,0x7f,0x7f,0x00,0x02,0x00,0x00,0x00 };

int const READ_PACKET_LEN = 5;
char read_packet[5];

ret = hid_set_output_report(hid,
 PATH_IN,
```

```
 PATH_LEN,
 write_packet,
 WRITE_PACKET_LEN);

ret = hid_interrupt_read(hid,
 USB_ENDPOINT_IN+1,
 read_packet,
 READ_PACKET_LEN,
 0);
```

The first thing to do is identify the particular device we want to talk to. This is done with the HIDInterfaceMatcher call simply by entering the vendor ID and the product ID. These two identifiers are all that is required to identify any USB device. If you have more than one identical device, it is possible to separate them by serial number, that is, two Matrix note readers would have the same vendor ID and product ID but different serial numbers. The HIDInterfaceMatcher call can do this; see the comments in the test_libhid.c file.

After some variable setup, the next step is to detach the kernel driver from the HID device. Upon insertion of the HID device, the kernel usually loads the usbhid module, which we don't want. We do have a few options, however, for unloading it or for not loading it in the first place. One such way is to enter this command:

```
root@localhost #> modprobe -r usbhid
```

When the hid_force_open function runs, it attempts, n times, to detach the device before it fails. The device is now free from any control, so our code now "opens" the device. As with any USB device, it is necessary to send control information to the device to activate it. This information must be sent periodically in order for the device to remain active. If the control poll stops, the device deactivates after a certain timeout.

Writing to a device requires the HID usage path and its length, plus a packet and its length. To find this out, we need to parse the usage tree—the output of `lsusb -vvv`—and obtain the path to the interface we want. As with everything else, there are various methods for determining the path. At this stage, a lot of time was spent determining what path to write to, and a number of tools are helpful here, such as:

1. The test_libhid.c code: when the correct vendor and product ID are entered in the code, the function hid_dump_tree, which uses the MGE hidparser (see Resources), which parses the HID usage tree and places the available usages at its leaves, outputs the available paths.

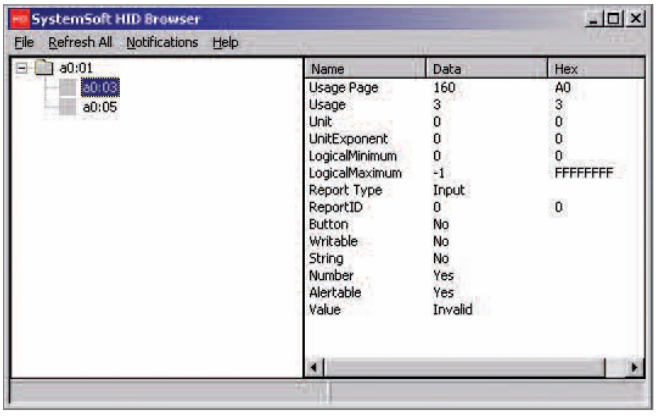
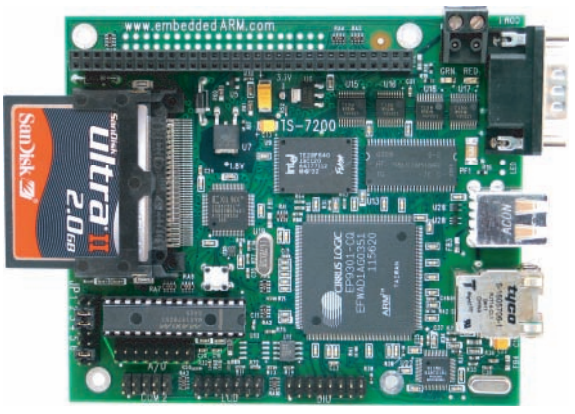


Figure 3. Understanding a device: one way to browse the available nodes of the HID tree is to use the SystemSoft HID Browser.

2. A Windows application available from Arnaud, one of the libhid authors, also parses the usage tree and produces a nice GUI output, as shown in Figure 3.

3. By parsing the output of `lsusb -vvv`, run as root, it is possible to parse the tree manually to determine the path. This process is explained in the comments of test_libhid.c code.

$119 qty 100
200 MHz ARM9
10/100 Ethernet
PC/104 bus
TS-7200 ARM9 Single Board Computer
Linux 2.4


Shown with optional Compact Flash
$149 qty 1
32 MB SDRAM (64 MB optional)
8 MB Flash (16 MB optional)
Compact Flash
10/100 Ethernet
2 USB ports
20 Digital I/O
2 Serial Ports
Options:
RS-485
8 ch 12-bit A/D
RTC (battery-backed)
Boots Debian stable from Compact Flash
Boots TS-Linux from on-board Flash
Technologic SYSTEMS
Many x86 and ARM based SBCs and peripherals available
Call for custom designs
We use our stuff.
Visit our TS-7200 powered website at
(480) 837-5200
www.embeddedARM.com

From the above methods, we now have a path value we can use for the hid_set_output_report. Once we know where to write to, it's a matter of what to send. This information should be in the technical documentation that comes with the device, and it can be verified with the USB-sniffing tools. As with the particular device I was using, verifying the format of a packet with the sniffing tools turned out to be important, as the information in the documentation didn't match what the sniff log reported (see the Snooping section).

Once the control message or output report is sent, we can start to read from the read pipe, endpoint0. The function needed is an interrupt read function. It already exists in libusb, but a corresponding libhid function doesn't. The developers of libhid simply hadn't come across a device that required it yet, so I studied the format of the other functions and implemented it appropriately. I also added a new error code to the existing list. These additions are being considered for inclusion in the latest version of libhid.

At this stage, once the interrupt read value is stored, I then parse this value, as per the Matrix documentation, to display the results to the user. For this device, that equates to information such as, "A ten-euro note has been inserted" or "The cash box is disconnected" and other such device-specific information. The details are unnecessary for the purposes of this article, but if anyone requires this detail, feel free to contact me.

This process is repeated for as long as the driver is running. We must keep polling the device to keep it active. There is a status LED on the device that turns green when the device is active and remains orange when inactive. The goal for quite some time was to make the little light go green.

Snooping

Snooping can be done with a number of utilities. This is where I learned about the discrepancies between what the Matrix documentation says and what actually happens:

```
[5037 ms]  <<<  URB 647 coming back  <<<
-- URB_FUNCTION_CONTROL_TRANSFER:
  PipeHandle            = 8180c814
  TransferFlags         = 00000002 (DIRECTION_OUT)
  TransferBufferLength  = 00000005
  TransferBuffer        = 92a137ed
  TransferBufferMDL     = fe9876e8
  UrbLink               = 00000000
  SetupPacket           =
    00000000: 21 09 00 02 00 00 05 00

[5038 ms]  <<<  URB 645 coming back  <<<
-- URB_FUNCTION_BULK_OR_INTERRUPT_TRANSFER:
  PipeHandle            = fe9876a0 [endpoint 0x81]
  TransferFlags         = 00000003 (DIRECTION_IN)
  TransferBufferLength  = 00000005
  TransferBuffer        = fefeef08
  TransferBufferMDL     = 81a18f48
    00000000: 00 20 00 00 1a
  UrbLink               = 00000000

[5038 ms]  >>>  URB 648 going down  >>>
-- URB_FUNCTION_BULK_OR_INTERRUPT_TRANSFER:
  PipeHandle            = fe9876a0 [endpoint 0x81]
  TransferFlags         = 00000003 (DIRECTION_IN)
  TransferBufferLength  = 00000005
  TransferBuffer        = fefeef08
  TransferBufferMDL     = 00000000
  UrbLink               = 00000000
```

From the snoop log, we see the control message sent to the device at the start, followed by a series of interrupt reads. According to the documentation, "The Host sends [a] poll to request information from Matrix at a periodic rate. Matrix answers to the poll and reports all the happening events." So, my interpretation of this was to send periodic control write messages to the device and read the responses from the interrupt endpoint. Also according to the documentation, the format of the write message is five bytes in length, so with this information, I used the test_libhid.c file included with libhid to see what happens. I found that functions within libhid give error codes if they fail and that the /var/log/messages file, with the extra DEBUG information from the modified kernel file, reports useful errors.

Upon closer inspection of the snoop log, I saw that the control write was, in fact, eight bytes in length. See SetupPacket in snoop log output. The five bytes described in the documentation seemed to represent the first five bytes of the packet, and the last three bytes seemed to be padding. That is, changing these last three bytes doesn't seem to affect the operation. Subsequent error-free testing, with the packet set to eight bytes, confirmed that the documentation had been misleading.

Conclusion

In terms of where to start with this project, I found the mailing list for libhid to be helpful. The libusb mailing list also provided guidelines. The Linux usbutils are quite useful in determining what interfaces are available on the device and the meaning of the descriptors.

The libhid source code, still in constant development, also is a source of help. Because the code constantly is being developed, it is a good idea to keep an eye on the Subversion repository for changes, including documentation changes such as helpful comments in the code.

Acknowledgements

Special thanks to Charles Lepple and Arnaud Quette, the original authors of libhid, and also to Martin F. Krafft, who later joined and led the rewrite. They all provided me with a lot of help, and without them I certainly wouldn't have gotten my little light to go green.

Also, thanks to my supervisor, Dr Paul O'Leary, at WIT, for his encouragement and analytical skills. It always is good to have an experienced pair of eyes to guide me in the right direction.

libhid uses the HIDParser framework made available by MGE.

Resources for this article: www.linuxjournal.com/article/8275.

Eoin Verling (everling@theverlings.com) qualified in 1998 and has been a sysadmin since. He currently is undertaking a research Master's in parallel computing at Waterford Institute of Technology, Ireland. There's nothing he likes better than a bit of ceol agus craic!

The Only Silo

"Edison's light bulb was important not because he was the first with the idea; as many as ten others envisioned similar schemes. Rather it was significant because he conceived not just a bulb but a whole electrified world."—Teresa Riordan, *US News*

BY DOC SEARLS

We've been fighting closed and proprietary software for a long time now. And we've had lots of success—enough, I think, that we need to move to the next stage: to the marketplace. We can see the problem when we look at how many closed systems have open foundations: Google and Amazon on Linux, Apple's Mac OS X and Yahoo's search infrastructure on BSD. Also, countless closed Web habitats served up by Apache.

Am I being unfair here? Perhaps a little. You can't be open in every possible respect, right? Some stuff needs to be locked down or closed off. Customer data, future product plans, trade secrets and "secret sauces" of one kind or another. But those aren't the issue.

The real issue is silos: closed habitats that serve as private marketplaces that lock customers in and competitors out.

Dick Hardt of Sxip Networks gives a killer talk about "Identity 2.0". As Dick puts it, Identity 1.0 is a province of walled gardens. Amazon, eBay, Flickr and Skype are all walled gardens. They may be lovely places to hang out, but they are also enclosed and private market spaces, as false in their own way as the faux Venetian canals and Parisian streets in Las Vegas hotels.

What makes them most different from closed systems of the traditional sort is not a lack of interoperability—often they have that—but the lock-in of personal data. You can't take

and then it hits you://

LINUX ISN'T A POLITICAL STATEMENT.

IT'S A PROFIT STATEMENT.

Novell®

find out more at novell.com

©2005 Novell, Inc. All rights reserved. Novell is a registered trademark of Novell, Inc. in the United States and other countries.

your eBay reputation, or the business you've built inside eBay's walled garden, over to Amazon. Even trivial data, such as your Skype contact list, isn't portable. It's locked inside a space that is not your own.

To hear Dick describe it, Identity 1.0 is barely past medieval. It's a country of duchies and city states. But since we're so used to it, we can barely think outside its walls. Yet that's where we belong, he says. The world we want—the Identity 2.0 world—is one of independent actors: free-range customers, conducting business and building relationships in ways that each individual controls and that work with many different vendors.

The problem with the walled garden metaphor is that all the familiar examples are native to the Web. Silos, on the other hand, are everywhere, both on the Web and off. Nearly every familiar business category you can name—banking, hospitality, retailing, commercial aviation, car rental...even office equipment, such as copiers and printers—is a forest of silos. Take airlines. I am a registered frequent flyer with United, American, Delta, America West and Southwest. Yet the only common way I can relate to any of the five is money. None of my data in the United silo is available for my dealings with American or Delta.

We've lived in a world packed with silos for so long that we now confuse them with a free marketplace. We actually believe that a choice of silos comprises all the conditions required for a free market. We can see how limited this is when we look at the market category we call computers. A quarter century ago, we thought the category was free and open because we had a choice of silos from IBM, Digital, Data General, Wang and HP. We thought the same way about networks when our choices were OmniNet, WangNet, IBM Token Ring, Sytek, Corvus, 3Com and Ungermann-Bass.

I remember a long conversation I had with Ralph Ungermann about how his company's goods were "open" because they interoperated with other networks. In a relative sense, they may have been. But the market was essentially a field of silos. What he offered was inter-silo-operability. Good for its day, but nothing like the Net that was to come—and which didn't come from any one vendor at all.

I remember Earl Gillmor talking about silos, way back in 1980. Earl enjoyed a small measure of notoriety as a member of a splinter group at Data General that lost "the shoot-out at the Holiday Inn" in Durham, North Carolina—an event immortalized in the early pages of Tracy Kidder's book, *The Soul of a New Machine*. After the shoot-out, Earl left DG and formed a new systems integration company in Raleigh called BAS, for Business Application Systems. BAS's goal was to produce what they called "machine-independent software". To be independent, Earl explained, you needed software that was independent of every hardware vendor's silo.

One day, early in my company's relationship with BAS, Earl explained the ideals of machine independence, all of which are familiar to anybody acquainted with open source today. (Although, naturally, BAS's code was proprietary.) When he was done with his rap, my business partner asked the impolite question, "So how do you make your money?" "We're whores", Earl replied. "We walk the streets with the rest of them." His point: they had no choice—except among silos. (BAS, it turned out, bet on the wrong silo: Texas Instruments' DS990s.)

Silo was just one container metaphor kicking around in those days. Others were smokestacks and stovepipes. Today those metaphors have fallen behind silo in popularity. I suspect that's because silos are completely contained. Unlike smokestacks and stovepipes, they don't have an opening at the top.

The prototypical office building is a silo of sorts. With its security systems, its employee and visitor badges, it comprises what David Weinberger calls "Fort Business":

The world we want—the Identity 2.0 world—is one of independent actors: free-range customers, conducting business and building relationships in ways that each individual controls and that work with many different vendors.

> This fort is, at its heart, a place apart. We report there every morning and spend the next eight, ten or twelve hours inaccessible to the "real" world. The portcullis drops not only to keep out our enemies, but to separate us from distractions such as our families. As the drawbridge goes up behind us, we become businesspeople, different enough from our normal selves that when we first bring our children to the office, they've been known to hide under our desks, crying.
>
> Within this world, the Web looks like a medium that exists to allow Fort Business to publish on-line marketing materials and make credit-card sales easier than ever....The Web isn't primarily a medium for information, marketing or sales. It's a world in which people meet, talk, build, fight, love and play. In fact, the Web world is bigger than the business world and is swallowing the business world whole. The vague rumblings you're hearing are the sounds of digestion.
>
> The change is so profound that it's not merely a negation of the current situation. You can't just put a big "not" in front of Fort Business and say, "Ah, the walls are coming down." No, the true opposite of a fort isn't an unwalled city. It's a conversation.

David wrote that more than six years ago, in Chapter 6 of *The Cluetrain Manifesto*. We still aren't having the conversation required to bring the walls down. True, there are some significant conversations growing out of employee blogs. For example, nothing has done more to bring down Microsoft's walls than interaction with outsiders by Robert Scoble, Kim Cameron (a subject of last month's column) and about 2,000 other blogging Microsoft employees.

But the problem isn't communication. It's the structure of markets themselves. I'm not talking about structure in an architectural sense, but in a deeper way that's more like geology. Because the Internet is geological, not just architectural. It has a nature that goes deeper than whatever structures private efforts can provide. But that nature is hard to see when your frames of reference are closed and proprietary.

Like many in the Linux community (including my good friend Eric Raymond), I have strong Libertarian sensibilities. I understand the liberating advantages of private property to societies and their economies. Ownership matters, and ownership works. But we in the Free Software and Open Source communities also know there are some things that are beyond the scope of ownership and the control ownership naturally implies. Earth below the crust is as beyond the practical scope of ownership as the weather and the stars. Yet they provide us with services so fundamental we couldn't live without them. One of those services is a deep and easily ignored context for property: gravity. Real estate would be meaningless without the gravitational pull provided by a mass we'll never see. The Net's geology is like that.

I've written many times about the NEA nature of the Net, and of all free and open-source software: Nobody owns it, Everybody can use it and Anybody can improve it. The same applies to markets, and it's time we started improving the ones we've got, by putting silos in a context that makes clear their limited advantages.

The Supreme Court missed a chance to do that with the Brand X case. In a 6–3 decision that was handed down on June 27, 2005—the same day as Grokster, which is a big reason why not much of a fuss was made about it—the Supremes upheld a 2002 FCC ruling that classified cable broadband as a deregulated "information service" rather than a "telecommunications service". Unpacked, that means the cable and telephone companies can (and will) be exclusive Internet service providers. Independent ISPs like Brand X and Earthlink, which don't own physical connections to homes and businesses, are out of luck if the cable and phone companies want to keep captive customers to themselves.

More important, the FCC's understanding of the Internet achieved the stature of law with the Brand X decision. That understanding is basically feature-rich broadcast. It's a concept of service anchored on the supply side of the highly asymmet-

and then it hits you://

YOU'RE WORKING WITH LINUX 24/7 IN

EIGHT TIME ZONES. AND SO ARE WE.

Novell®

find out more at novell.com

©2005 Novell, Inc. All rights reserved. Novell is a registered trademark of Novell, Inc. in the United States and other countries.

What makes Linux so different, and so successful, is that it's not designed as a silo.

rical distribution system the FCC has governed for most of the last century.

Former FCC Chairman Michael Powell, in a speech at the VON (Voice On the Net) conference one year ago, said, "To realize the innovation dream that IP communications promises, however, we must ensure that a willing provider can reach a willing consumer over the broadband connection." He generously acknowledged "the importance of consumer empowerment" and rights such as "Freedom to Access Content", which he explained with "Consumers should have access to their choice of legal content." Generous as that may have been, it was no less top-down than anything owned by Rupert Murdoch. Nowhere did he acknowledge the Net's most profound commercial grace: supporting the ability of people to go into business, and to do business, with anybody they please, anywhere.

Thomas Madsen-Mygdal, a young serial entrepreneur in Denmark who hosts the delightful annual reboot conference in Copenhagen, recently told me he likes and appreciates Flickr—the on-line photo gallery phenomenon that has taken the world by storm (and which was built on LAMP)—but that it has "lock-in" issues:

> I don't mean "a total lock-in" in the traditional BigCo IT sense. More like that if open data standards existed, the collective value would be in the commons—not on one photo sharing site. The London bombings wouldn't be about the "flickr tag", but about the "photo tag" or just the *tag*—which in my book is much more aligned with our values and the society we want to create.

Thomas' higher-level concern is that "we're selling out on values of open standard and decentralization". What Tim O'Reilly calls the "architecture of participation", Thomas says, is turning into something that is "based on silos" in practice. So, he adds, "I'm gonna try and push some open standards in the photo sharing space to level the competition."

He'll do that, he says, by "dividing what is the commons from what is the product. That way, thousands of photo sharing products can create a collective value that's a lot greater. Competition will be on the product side, rather than on who aggregates most of the commons." Thomas' site, 23people.com, is open for beta.

In September 2005, O'Reilly put on its second Web 2.0 conference. (Shouldn't they call it Web 2.1?) In September 2004, Tim O'Reilly described Web 2.0 as "the Internet as a platform". Then he added:

> We heard about that idea back in the late 1990s, at the height of the browser wars, but that turned out to be a false alarm. But I believe we're now starting the third age of the Internet—the first being the telnet-era command-line Internet, the second the Web—and the third, well, that tale grows in the telling. It's about the way that open source and the open standards of the Web are commoditizing many categories of infrastructure software, driving value instead to the data and business processes layered on top of (or within) that software; it's about the way that Web sites like eBay, Amazon and Google are becoming platforms with rich add-on developer communities; it's about the way that network effects and data, rather than software APIs, are the new tools of customer lock-in; it's about the way that to be successful, software today needs to work above the level of a single device; it's about the way that the Microsofts and Intels of tomorrow are once again going to blindside established players because all the rules of business are changing.

That was a lead-in to the Web 2.0 conference. After the conference, in an interview with Richard MacManus, Tim said:

> I actually ran a couple of panels on this at our Open Source convention, a year and a half or two years ago—called "Open Data—Do We Need a Bill of Rights for Web Services?" We had people from Amazon, eBay and others trying to answer that question: what does it mean when we're investing our on-line data in these sites? Will we end up with something like the Open Source movement because the companies have ended up locking in their users?
>
> ....But the actual data ownership is maybe less important, in some areas, than people think. When we talk about user-contributed data, we're not just talking about my data proper (as in having your mail stored on Gmail or Yahoo! Mail or whatever). We're also talking about a kind of content that users are contributing to a collective work. So for example, Amazon Reviews—people don't really care about that in the same way. They're not saying, "Oh I created that review and I want to be able to export it to Barnes & Noble as well." They're creating it in a particular context of that community.
>
> ....Despite what I've said...data lock-in absolutely should be a concern. I believe that data lock-in of various kinds is going to be one of the key tools of business advantage in the Internet era. I think that as companies realize this, they will figure out how to be evil—so to speak (to use Google's terminology)—and I predict that we will in fact have some major battles in that area.

As I said last month, one answer is to create ways to do what Drummond Reed calls "Company Relationship Management" (or CoRM), which should look far more interesting and useful to companies than their own Customer Relation Management (CRM) systems, which by

nature have no view outside the company's own silo. In fact, CRMs are one of the main ways companies *maintain* their silos.

Another is to pay more attention to where the Net's deep, almost geological market-making infrastructure comes from. It's not from the physical cables that run to homes or from the "services" available exclusively from cable and phone companies, but from the open protocols that define the Net's environment. It's also not from fancy private services inside corporate walled gardens but from the raw building materials that make deploying those services so free and easy.

Which brings us back to the L in the LAMP suite that makes possible the last phrase above.

What makes Linux so different, and so successful, is that it's not designed as a silo. Linux didn't come from a silo, and it had no ambitions to be a silo. At one point, Linus talked about "world domination", but his tongue was in his cheek—even if he was indulging in prophesy that would prove out in the long run.

Linux was never a business. It was, and remains, a great way to build anything, to support anything, for anybody. That's the fundamental virtue we need to fight for when we go to battle.

Our battle, however, is not with the companies that use open code to build walled gardens and silos. Our battle is with the closed, top-down silo-oriented value system that has been with us since the dawn of the Industrial Age. It's that lame old value system that prevents us from imagining how we can improve markets that nobody owns and anybody can improve.

The best way to shed the old mentality is to embrace our customers and not only their money. Today the preponderance of inventiveness and productivity is out in the free world, in the hands of free-range individuals. Linus Torvalds is the prototypical example of one of those individuals. There are countless more like him, producing all kinds of goods, expressing all kinds of demand—much of which they are able to supply for themselves, as Linus did, and with the help of others, as the Linux community has done.

In fact, the only silo that matters is the most fundamental and indivisible unit in the marketplace, the individual. What we need is to create and support independence, not dependence.

Work to free individuals, and to take advantage of what they do with that freedom, and you'll have a winning strategy in the new marketplace we're all making together.

Doc Searls is senior editor of *Linux Journal*.

and then it hits you://

LINUX WORKS WITH ANY PLAN.

ESPECIALLY THE FY '06 BUDGET.

Novell®

find out more at novell.com

©2005 Novell, Inc. All rights reserved. Novell is a registered trademark of Novell, Inc. in the United States and other countries.

Fixing Web Sites with GreaseMonkey

Who says "View Source" on a Web page has to be a read-only proposition? Re-mix your favorite Web sites by changing styles, adding and removing elements, and more. BY NIGEL MCFARLANE

Here's a strange thing: hacking open source isn't done only at midnight, in the spare room, hunched over the protocol analyser, the breadboard, source code control and some helpless device. No, sometimes it's done inside a different crucible entirely: a public world of shameless posturing and self-promotion. A lurid and neon habitation of signs, shops, styles and stuff populated by the babble of conversations both informed and banal. It's a place of great joy and great angst; a place of towering conservatism and the last bastion of the radical voice. Within it, a good hairdo or a radically cut legline can get you as far as a symbolic debugger, possibly even further. Devices they may be, but of a different cut entirely from those of hardware. Its denizens slip hyperactively in and out of view like character actors with coffee addictions and inspired agents. Of course, I refer to the World Wide Web.

In this article, you learn how to code in a new way, a way that's about changing media, not about changing programs. To enter this nightclub and experience the beat, you need the right gear, and the right gear is Mozilla Firefox and GreaseMonkey. Alfred Bester and William Gibson are waiting, so ready your Mojo and prepare for cyberspace insertion. But first, a bit of background.

Web Pages as Open Source

We tend to forget that the Web is open source, in a way. Some of the Web's infrastructure, browsers and servers, is traditional open-source software, but the idea also applies to Web page content. Appropriation of code is an everyday occurrence. Every day, Web developers and designers use the View Source browser feature to appropriate (industry term: steal) code and design from other people's pages. It was ever thus, and it remains so. Ideas and code are shared freely and often; it's an art design sensibility.

Most technical people have dabbled with Web development, and dabbling is an easy way to have a bad experience. The big three technologies—HTML, CSS and JavaScript—were riddled with bugs for many years after their inception. That's the experience that probably looms large for early adopters who first tried it out in the 1990s and walked away in disgust. Cross-browser code? No, thank you.

Fortunately, matters have improved tremendously as of late, and the Web is reviving as a technology platform. Better standards support, more standards support and the decline of hoary old misgivings, such as Netscape Communicator 4.x and Internet Explorer 5.0, have left Web developers with a nearly clear shot at real portability, a shot frustrated only by the once shiny but now fairly rusty Internet Explorer. In 2005, the buzz is about Modern DHTML, Layout without Tables, Semantic Markup and Asynchronous JavaScript and XML (AJAX). Client-side Web development is coming back, and these are the things of which it's made. This time, the Web is backed by professionals with formal Web training and veterans with ten years of experience. These people have their acts together, and it's possible to say things about Web technology that are no longer drowned out by the static of incompatibility issues.

Supporting and colonizing this trend is the Mozilla Firefox Web browser, and Mozilla technology in general. Of course, Mozilla is fully open source, as open as a religious movement can be, and so there's plenty of room for experimentation. The critical bit of Mozilla and Firefox is its interpreted nature. On top of a big, bad, networked C++ rendering engine is a thin skin of JavaScript scripts and XUL, an XML dialect. This makes Mozilla a distant cousin to Emacs or Tcl/Tk, as it provides the whole Firefox user interface by way of interpreted code. By writing an extension, you can enhance this user interface and drop it in to thousands of willing people's daily experience. Go to update.mozilla.org to see the endless possibilities made real by this system. Every variant hardware device requires Linux kernel driver support; every variant human expectation about user interfaces requires a Firefox extension. That's a lot of extensions.

Grabbing GreaseMonkey

GreaseMonkey is a Firefox extension (see the on-line Resources). You have to click on the link twice, once to trust www.mozdev.org and once afterward to install the extension. GreaseMonkey differs from the other extensions because it provides no specific user-interface enhancements of its own other than a configuration dialog box. Instead, it creates a macro-like scripting environment into which you put JavaScript scripts. Those scripts operate on Web pages that you specify. When such a page loads, your script goes to work on the page content, no matter who provided it. You're intercepting a content provider's content and modifying it before it hits you. No wonder GreaseMonkey's been called "TiVo for the Web". I wrote about page modification tactics in *Rapid Application Development with Mozilla* (Prentice Hall, 2004), but GreaseMonkey has moved that idea into the mainstream by supporting traditional Web-scripting techniques and by packag-

Turn Control Freaks Into Remote Control Freaks.

Take charge. Win the battle and take control, right from your comfy chair. It's easy to conquer the challenges of managing serial devices in the data center with the CCM serial console manager.* When used with DSView® 3, AVWorks®, or industry-standard SSH/Telnet client software, you can remotely control servers, network gear, telco and power devices from a single interface. With proactive alerts and offline buffering you can remotely diagnose failed devices and reduce downtime without setting foot in the data center. Visit us at www.avocent.com/serialcontrol. And start looking for a new chair.

Avocent®
The Power of Being There®

** Recliner recommended, but not included.*

Avocent, the Avocent logo, DSView, AVWorks and The Power of Being There are registered trademarks of Avocent Corporation. © 2005 Avocent Corporation.

ing it all up into a digestible product.

For all Firefox extensions, you must restart Firefox completely to finish the install. Use File→Exit to do that safely.

Bucket-loads of pre-existing GreaseMonkey scripts are available (see Resources). Before you get too excited though, note that such scripts are tied to one Firefox installation and have no effect on any server. On a Linux or UNIX box, such scripts might affect a large user population, but they're primarily a personal thing. For those readers switched on to people problems, the broader implications should be obvious.

Spotlight on LinuxJournal.com

To see all this at work, in this article I hack the *Linux Journal* Web site with GreaseMonkey. My esteemed editor, Don Marti, even asked me to do this. A brave man indeed. [*Maybe next time they'll invite me to the Web site meeting.—Ed.*]

Give me a hill and I'll climb it. First up is a bit of scrutiny of the site due for surgery. Recall it's www.linuxjournal.com, if you're reading this in print. This is also the fun part; personal tastes differ, and for user-side drivers—which effectively is what GreaseMonkey scripts are—it's entirely valid and professional to be picky and subjective. In Mozilla-land, dogfood means testing your fixed bugs for technical correctness, and catfood means testing your inventions against unreliable and subjective people who might spring in any direction. It's all catfood here, and there's no right or wrong. After reading this article, *LJ*'s long-suffering site maintainer will likely glare at me venomously or perhaps change the site before this sees print. Design sensibilities, you see. Sorry mate, they made me do it. Hard-core engineers should look away; you might find this analysis distressing. On to the site.

Here's a handful of observations.

- The site icon, which appears in the location bar and on the current tab if you use tabbed browsing is dinky and uninspired. Oh well.
- There's advertising everywhere.
- *Linux Journal*'s supposed to be the granddaddy of technical journals in open source, excluding academia and professional bodies. Where's that indicated?
- The headings are red. What's with red? I'm not in a hurry.

On the plus side—my survival as a critic is at stake—the site has a robust three-column layout and is clean overall. Someone knows his or her stuff. Viewing the source, the layout is all done with CSS, so that's relatively modern; many industry sites still pump out the worst HTML you can imagine. The excessive use of <DIV> tags shows that *LJ* is halfway through modernisation; there's still some Semantic Markup work to go, where meaningful tags are used as content descriptors instead of the meaningless <DIV>. That update might improve the site's search engine performance or so it's claimed.

The Right Tool for the Right Job

Now, of the above personal observations, some are simple to rectify and do not require GreaseMonkey. If you dislike advertising, then the AdBlock extension is for you; there's nothing, or at least little, to code. Similarly, for a long time, all browsers have supported user-specified stylesheets. If you install the ChromEdit extension, you can get at that stylesheet without having to grovel through the filesystem looking for it. Bring it up via Tools→Edit User Files, click the userContent.css tab and start typing. To make headings blue, you might add:

```
h1.title a { color : blue !important; }

@-moz-document domain(linuxjournal.com) {
 h1.title a { color : blue !important; }
}
```

The first rule applies to all Web sites; the second is a Mozilla special that applies only to the Web site specified. Browser-specific is okay here, because we're working purely on and in the client side.

You can get a lot done in these stylesheets, especially if you know CSS well. You can hack the page's layout to bits by reordering, hiding or floating columns and other content. All of these options are possible via GreaseMonkey as well, but GreaseMonkey is better suited to bigger stuff. In other words, don't go to GreaseMonkey if page changes are easily solved with a stylesheet; it's overkill.

For this article, we'll make one simple change. We'll bring some gravitas to the site by replacing some content with fancy calligraphy drawn from another site.

Illuminated Drop-Caps for Paragraphs

The CSS :first-letter pseudo-selector lets you take an ordinary paragraph of text and make the first letter big, so that several lines of text flow around it. It's a self-important feature and what we're looking for. We simply could apply that feature, but most computers don't have fancy medieval fonts installed. And, a big Times Roman letter F isn't that exciting. It would be better if we could get the LinuxJournal.com Web site properly illuminated, like the *Book of Kells*, with extra fancy calligraphy.

Here are a couple of screenshots showing the before-and-after looks, taken on Windows XP Professional. This is a timely reminder that the user experience is what's important here. It also emphasizes that open source means cross-platform when stated in Mozilla terms. Everything described in this article

Figure 1. A Regular Linux Site

Figure 2. That's better. We didn't need monks to illuminate this manuscript, simply a GreaseMonkey script.

works identically on Windows, Macintosh, Linux and various obscure Mozilla platforms, such as Solaris.

In the second screenshot, you can see that the first letter of each paragraph has been replaced with a fancy illuminated letter. Because I don't have access to the back end of the *LJ* Web site, that's something of a feat. In fact, these images come from the Australian National University's Medieval Studies image server.

I've used the thumbnail images only. It's a bit naughty to serve up some other Web site's images, and these images aren't perfectly cropped, registered scans, but for the purposes of, well, illustrating a technique, they'll do. Let's hope some parsimonious old sod doesn't take them down by the time you read this.

Spinning Up the Script

To make this embellishment work, you need a GreaseMonkey JavaScript script. To make such a script, proceed as though this were any other Web page project. I saved to local disk the *LJ* home page and then added this to the end of the <head> section:

```
<script src="illuminate.js></script>
```

Now I'm free to develop that script in pure Mozilla JavaScript, with no cross-browser constraints, because GreaseMonkey works only on Firefox. Let me tell you, it's real 200%-proof pleasure to charge forward in JavaScript without once having to trip over document.all or other MSIE aberations. More than that, there's a bit of now-established rigor we can bring to the code. Here's a skeleton of the job at hand, in the shape of a JS object signature, which is the bit of syntactic dogma that I like to propagate:

```
var illuminate = {
  caps : { ... },
  load : function () { ... },
  image : function (text) { ... },
  mask : function () { ... },
```

Hurricane Electric Internet Services...Speed and Reliability That Gives You A Lap On the Competition!

Flat Rate Gigabit Ethernet
1,000 Mbps of IP
$13,000/month*

Full 100 Mbps Port
Full Duplex
$2,000/month

Colocation Full Cabinet
Holds up to 42 1U servers
$400/month

Order Today!
email sales@he.net or call 510.580.4190

he.net

*Available at PAIX in Palo Alto, CA; Equinix in Ashburn, VA; Equinix in Chicago, IL; Equinix in Dallas, TX; Equinix in Los Angeles, CA; Equinix in San Jose, CA; Telehouse in New York, NY; Telehouse in Los Angeles, CA; Telehouse in London, UK; NIKHEF in Amsterdam, NL; Hurricane in Fremont, CA; and Hurricane in San Jose, CA

```
  insert : function () { ... },
  getElements : function (node) { ... },
  getElementsRecursive : function (l, n) { ... },
};
```

There's no static typing and barely any forward declarations in JavaScript, so there's nowhere to declare an object. Instead, we work with an object literal. This approach creates a packaged set of functionalities that expose only the illuminate option to the page's namespace. So it's both a reuse strategy and a namespace non-pollution strategy. All the methods of the object are expressed as anonymous functions, and caps is a sub-object in which we put data. Anonymous functions also save you from forcing a function name into the page's namespace. That's a vast improvement on early scripting techniques.

Once defined, this object does nothing. You need a line such as this to make it go:

Listing 1. Illuminating the Current Object

```
var illuminate = {
    caps : {
                "a" : "102.PNG", "b" : "103.PNG",
                "c" : "104.PNG", "d" : "105.PNG",
                "e" : "106.PNG", "f" : "107.PNG",
                "g" : "108.PNG", "h" : "109.PNG",
                "i" : "110.PNG", "j" : null,
                "k" : "111.PNG", "l" : "112.PNG",
                "m" : "113.PNG", "n" : "114.PNG",
                "o" : "115.PNG", "p" : "116.PNG",
                "q" : "117.PNG", "r" : "118.PNG",
                "s" : "119.PNG", "t" : "120.PNG",
                "u" : null,
                "v" : "121.PNG",
                "w" : null,
                "x" : "122.PNG",
                "y" : null,
                "z" : "123.PNG"
    },

    load : function () {
      this.mask();
      this.insert();
    },

    image : function(text) {
      var a = text.substring(0,1).toLowerCase();
      var link = "";
      if (a && this.caps[a]) {
        link = 'http://rubens.anu.edu.au/htdocs/' +
               'bytype/prints/ornament/0001/' +
                this.caps[a];
      }
      return link;
    },

    mask : function () {
      var head = document.getElementsByTagName(
                        'head')[0];
      var rules = document.createElement('style');
      var text = document.createTextNode(
        'div.node > div.content > img[ill] : ' +
        '{ display: inline; float:left; }\n' +

        'div.node > div.links : ' +
        '{ clear : left; }\n' +

        'img[ill] : { display : none; }\n'
      );
      rules.appendChild(text);
      head.appendChild(rules);
    },

    insert : function () {
      var list = this.getElements(window.document);
      var img;
      var text;
      for (var i=0; i<list.length; i++) {
        text = list[i].firstChild;
        if (text.nodeType == 3 ) {
          img = document.createElement('img');
          img.setAttribute('ill','true');
          img.setAttribute('width','64px');
          img.setAttribute('height','64px');
          img.setAttribute('src',
                this.image(text.nodeValue));
          text.nodeValue =
                text.nodeValue.substring(1);
          list[i].insertBefore(img, text);
        }
      }
    },

    getElements : function (node) {
      var rv = [];
      this.getElementsRecursive(rv, node);
      return rv;
    },

    getElementsRecursive : function (list, node) {
      for (var i=node.childNodes.length-1;i>=0;i--)
      {
        var child = node.childNodes.item(i);
        var klass = null;
        if ( child.nodeType == 1) {
          klass = child.getAttribute("class");
          if ( klass && klass == "content") {
            list.push(child);
          }
          this.getElementsRecursive(list, child);
        }
      }
    }
}
```

```
window.onload = function () { illuminate.load(); }
```

That causes the load() method to run when the page is finished loading. The anonymous function that wraps it provides an extra scope that makes illuminate the current object. Done that way, the reference points to the illuminate object, which means this can be exploited from inside the methods of the object. That saves the object from ever having to use the illuminate variable name—more namespace non-pollution.

Listing 1 shows this object fully implemented, so let's go through it. You also can grab the complete script from the *Linux Journal* FTP site.

The caps associative array points to the individual letters available at the ANU Web site. Because they're from medieval times, new-fangled letters such as J, U, W and Y are nowhere in sight. We simply have to do without those for now and also resist the urge to use I for J biblically. The alphabet's constantly changing, albeit slowly, anyway. The way I hear it, if radio is a true reflection of the street argot, Double-u is next to undergo change. Evidently it's being replaced with an identical letter named Dub, as in: "Go to Dub Dub Dub dot sell you something dot com". Think what you will of that despicable trend. But I digress.

The load() method does all the work. It calls mask(), which inserts a <style> tag as the last thing in the head of the current page. Careful study of the neatly designed *LJ* home page lets one create styles that fit like extra jigsaw pieces in the existing layout regime. This first style acts on the new illuminated letters, allowing text to flow around them:

```
div.node > div.content > img[ill]
        { display : inline; float : left; }
```

This next style stops the float effect so that the next news item doesn't flow around it as well:

```
div.node > div.links { clear : left; }
```

That's all standard CSS2 stuff. Finally, the rest of the JavaScript code is a bit over-enthusiastic in its page hacks, as you will see. So, here's a style to shut up the accidental extras:

```
img[ill] : { display : none; }
```

In all cases, the ill bit is simply a custom tag attribute added to identify the images specific to this script, so that they can be picked out easily with a style rule.

The second thing that the load() method does is call the insert() method, which adds <img> tags to the main content of the page. To those who dabble only in client-side scripting, perhaps an onclick handler or two, this looks fairly formidable, but it's pretty routine stuff for quality client-side scripting.

The insert() method acquires a list of important nodes in the page using the now robustly supported DOM APIs. It then uses a loop to run through that list, adding an <img> tag of this kind:

```
<img ill="true" width="64px" height="64px" src="">
```

This is added to every node found that has a Text node as its first child. That amounts to adding a new child node to any tag that's immediately inside a tag. That's a big assumption about the page's structure. Also, there are many unwanted examples of that combination, for instance, in the advertising column on the right and in miscellaneous content outside the list of articles. That's why I had to shut up some images with an extra style—too many are inserted. It keeps the code simple to use a broad brush, though.

While developing this, I also noticed that on one instance of the home page, someone had added extra <p> tags to the deck of one article. The deck is the lead-in remarks that draw the reader to the full article content. That's a simple typo or random act of innovation on some editor's part. For that one article, displayed in a layout marginally different to the rest, the script failed to do anything. At least that's better than generating an error or an exception and halting. It does go to show, though, how fragile GreaseMonkey scripts can be if one's not circumspect enough and has ignored the matter of graceful degradation, in which scripts melt away to NO-OPs if things go pear-shaped. Any assumptions made about the page's expected structure should be as general and as flexible as possible. Tread lightly.

Back in the code, insert() also uses standard JavaScript string operations to chop the first character off the deck's text. So that's one plain textual character gone, one image of a character added. Between the Web and Unicode, saying the word character without caution is to flush out in a trice all the lexicographical pedants lurking in the woodwork. Let them come, I say.

The rest of the object is some routine processing leveraged

by the insert() method. The image() function is the easier utility: it merely performs a dictionary look-up on the caps object, which is effectively an associative array. JavaScript allows literal strings to be used as array indices and object member names. The retrieved filename is concatenated into a full URL and returned. It's simple data-driven programming.

The other utility is the remaining two methods, getElements() and getElementsRecursive(). They implement a standard prefix tree-walking algorithm that acts on the whole DOM of the page and that is wrapped up in the neater façade of getElements(). They are page-scanning routines and not overly general as there are logic tests inside specific to *Linux Journal* content. Someone should write a set of qsort(3)-like navigation routines so one simply can plug in a comparator functor or two. Probably that's already been done, but I haven't tracked down such a thing for this article.

As the DOM tree is walked, any nodes are appended to the list of discovered nodes. There's no copying at work; it's all nodes by reference. Walking a whole DOM tree is a bit ambitious. For more focused GreaseMonkey hacks, it's more efficient to go straight to the page element at issue, perhaps with a document.getElementById() call. When you're not sure about the exact structure of the page, though, it's better to grope blindly through all the content with a minimum of assumptions. How directly you proceed simply depends on what kind of leverage you're looking for.

Now that the script is developed, all that needs to be done is to configure it into the GreaseMonkey extension. Recall that so far it has been developed on a static and locally held test page. That configuration task is, to be frank, a bit weird, at least at GreaseMonkey 0.3.3.

To get it in place, make sure the script is named illuminate.user.js. Next, using Firefox with GreaseMonkey installed, navigate to the local directory where the script is. On Linux that is something like:

```
file:///home/nrm/
```

On Windows it may start with:

```
file:///C|/Documents%20and%20Settings/nrm/Desktop/
```

Notice the three forward slashes. The file URI scheme is similar to NFS or SMB and, in theory, can retrieve files located anywhere, for example:

```
file://www.example.com/something.txt
```

Omit a domain and the default is localhost, which generally is what you want.

Once that directory listing appears, you should see a link for the illuminate.user.js file. Right-click on it (context-click on Mac OS X), and the magic option is revealed: "Install User Script ...". Pick that, because no amount of fiddling with the GreaseMonkey options on the Tools menu can bring you equal joy. The GreaseMonkey configuration dialog box appears next, with the new script lodged on the left. Click Add on the right, and type in the *Linux Journal* URL, like so:

```
http://www.linuxjournal.com/*
```

Click OK and the script's installed. Now it can be reached via the Tools menu for subsequent administration. Reload the *LJ* page and everything should work, with illuminated capitals in place. If not, it's time to open the JavaScript Console and go back to script debugging, testing with 1.0.4 and GreaseMonkey 0.3.3.

Illumination Postmortem

The tale of illuminated capitals thus is told, and it's a tale of content aggregation. Of course, this is but a trivial example. You're not restricted to patching-in a single, grubby <img> tag, nor must you be so sanguine about the existing page content. GreaseMonkey scripts can hack the page content to bits, and you can stick any amount of extra content into the page, from any source. The redoubtable XMLHttpRequest object is available to such scripts, and it can be used to load any content in the world that's accessible by HTTP—content that then can be put in the current page. You also can send bits of the current page elsewhere with this object, but that's another article. Here I've attempted a graceful addition to the page.

Now you might say, "this is an exercise in folly, no one will see my work but me." That, however, simply is a distribution problem, one solvable by many different IT deployment techniques, not the least of which is hypertext links.

Such enhancements are not so silly either. Imagine the Web interface to your favourite network device, perhaps a router. Wouldn't it be nice if the host status lights from the open-source and Web-enabled Big Brother LAN-monitoring application appeared next to the IP addresses of the matching hosts in the filtering rules in the Web pages served up by that router? At least then you wouldn't be trying to fix a route for a box that's not even running in the first place. You'd see a big red light next to the entry in the router's configuration pages. GreaseMonkey is exactly the right tool for such problems, especially since no one has access to the source pages generated by the router's embedded Web server.

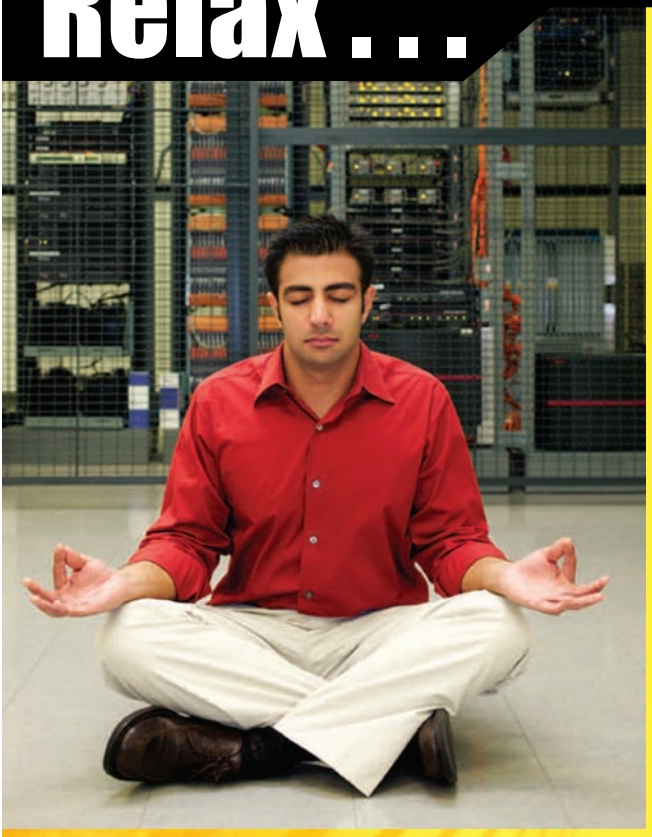
Furthermore, many Web pages are busy places, full of navigation widgets and data entry fields. GreaseMonkey scripts can hack all that to bits, removing or adding elements to the page that streamline the user's individual surfing behaviour. Don't like that menu bar at the top? Hide it. Can't remember how to fill in that form? Add some reminder text that floats above it. You get the idea.

Finally, because GreaseMonkey is content-based, analogies with other content media are worth considering. If there are hit records and hit movies, then a hit GreaseMonkey script no doubt will emerge in time. What political orientation it has with respect to the Web site it hacks will set a very interesting precedent. Will it be a script that protests, deconstructs, graffitis, supports or censors the site in question? Only time will tell. In the meantime, GreaseMonkey is a handy tool for Web content that's otherwise difficult to change.

Resources for this article: www.linuxjournal.com/article/8458.

Nigel McFarlane is the author of *Firefox Hacks* (O'Reilly Media) and *Rapid Application Development with Mozilla* (Prentice Hall PTR). For more information about Nigel's contributions to the Open Source community, see page 12.

Relax . . .



STORIX®

System Backup Administrator for Linux V5.2

As the leader in system and data backup and disaster recovery for Linux on Intel and IBM pSeries platforms, Storix announces the release of System Backup Administrator (SBA) Version 5.2.

Storix SBA provides centralized management of system level and data backup for one system, or the entire network, and includes the most complete and flexible bare-metal recovery capabilities available today.

NEW FEATURES in Storix SBA v5.2

- Microsoft Windows (SMB) client data backup
- 128/192/256-bit backup data encryption
- Network boot for bare-metal recovery
- Multiple boot loader support for x86/Linux
- User access rights management
- Enhanced exclude list support
- Significant performance enhancements

TIME-PROVEN FEATURES of STORIX SBA

- Complete system and data backup and restore
- Flexible disaster recovery
- System migration to different hardware
- System cloning from disk, tape or SAN
- Snapshot backups of live data

We Can Rebuild it!

PRODUCT EXCELLENCE AWARDS
San Francisco 2004
LINUXWORLD CONFERENCE & EXPO
WINNER
Best Data Backup Solution

STORIX® SOFTWARE

INGRAM MICRO

LIFEBOAT DISTRIBUTION

Tech Data

www.storix.com

1 877-STORIX1 sales@storix.com

©1999-2005 Storix, Inc. All trademarks are the property of their respective owners.

The Linux for Kids Experiment

This Linux dad got the young members of his family set up with educational software, art tools and games.

BY PAUL BARRY

When pressed to answer truthfully, most parents agree that raising kids is a big experiment. In the December 2004 issue of *LJ*, Diego Betancor's letter motivated me to experiment with something I've been meaning to do for some time. Diego wanted to see more content in *LJ* aimed at kids, and his suggestion was the inspiration for the next phase of my child-rearing experiment: moving the kids to Linux.

My wife Deirdre and I have three young children: Joseph, age nine; Aaron, age seven; and Aideen, age five—see Figure 1. As shown in the photo, indoctrination starts early in the Barry household: there's a fluffy Tux in the foreground and an electronic Tux on the screen.

Figure 1. The "Linux for Kids" Kids, (left-to-right) Joseph, Aideen and Aaron

With his dad being a longtime computer geek, it came as no surprise when Joseph took to the computer at a young age. For years, our home computer was a first-generation iMac, running Mac OS. A great 3-D shoot'em-up game came with the iMac, *Nanosaur*, that Joseph just loves. Despite this, our household software policy always has been to try to ensure that any software brought into the house is classified as educational. Therefore, Joseph also has a bunch of *Land Before Time* and *Zoombini* titles, as well as kiddie-strategy games, such as *Darby the Dragon*. Other software includes the usual encyclopedia, dinosaur and space-exploration titles.

Aaron is the sporty child in the house as well as the artist, and he has been happy to sit and play with the paint application integrated into ClarisWorks, the simple office suite that came with the iMac. Aaron also likes to play with Joseph's software, as well as some of Aideen's titles, which include *Green Eggs and Ham*, *Sammy's Science House* and *Thinking Things*.

As long as there are a lot of bright colors and funny sound effects, Aideen's happy, even though this five-year-old's attention span is not at all lengthy.

As great as it is, the iMac had been showing its age for some time. It also has become increasingly difficult to find original software titles for its effectively discontinued OS version. Trying to upgrade to Mac OS X or any modern version of Linux was not an option for the iMac; it's simply too underpowered. Without new titles, the kids were getting bored with the iMac and had been asking for a new computer. They also constantly bugged both me and their Mum to install various Windows titles on our laptops—especially the demo software that comes free inside various cereal packets. As Deirdre has to run Windows 2000 for work, her laptop was the one infected with a growing collection of these types of titles.

A few months back, a new computer arrived in the form of a Dell Optiplex GX270, with 512MB of RAM, a 40GB hard disk and a flat-panel monitor. As I'd rather eat the new PC than allow the kids to use the factory-installed Windows XP, I looked for a family-friendly Linux distribution to install instead. Having recently experimented with Ubuntu Linux as my office desktop, I downloaded and burned a copy of the Warty Warthog release for use at home.

Like most big kids, I love experiments, and now my experiment had a plan: replace the Dell's factory-installed OS with Ubuntu, pack it full of kid-friendly software, let the kids at it and see how they get on.

Going Cold Turkey

I deliberately decided against installing any type of emulation that would have allowed the kids to run any of their existing software titles, even though such technology is well established within the Linux community. My main reason for doing this was to see if the kids would identify any titles that they missed.

If they did, I'd try to find native alternatives, install them and see if the yearning subsided.

Installing and Configuring Ubuntu

Ubuntu installed easily on the Dell, taking about one hour from start to finish. Once the base OS was up and running, I installed a bunch of stuff for the kids to use. I created a user ID called kids with a password of dinosaur and then set up a window in Nautilus to mimic the look and feel of the Mac OS Launcher program, as shown in Figure 1. Nautilus hopefully would provide a familiar look and feel for my pint-sized, Mac-loving user community.

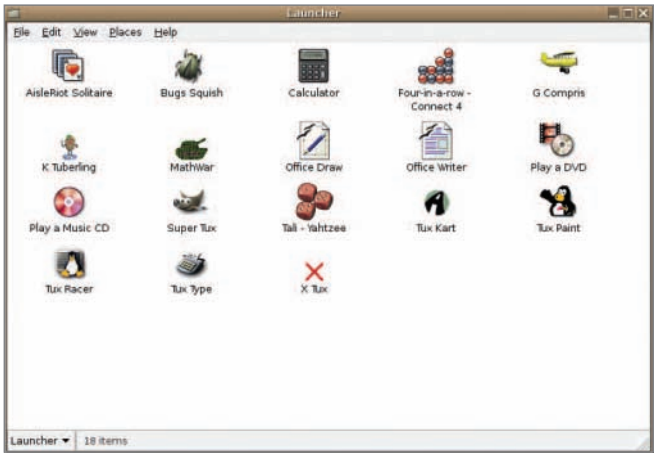


Figure 2. The Linux for Kids Launcher, Courtesy of Nautilus

Software for the Kids

In an attempt to ease the introduction of a new—and somewhat different—computer into the house, we decided to relax our household software policy and install a few nice Linux games along with the educational software. Here's a quick rundown of the titles we decided to make available on the desktop launcher. Unless stated otherwise, these titles were downloaded into Ubuntu using the included Synaptic Package Manager. It helps to refer to Figure 2 while working through this list.

- *AisleRiot Solitaire* (/usr/games/sol) is a Linux version of the classic solitaire game. It came pre-installed on Ubuntu and was elevated to the Launcher in an attempt to provide a familiar piece of software on the new desktop.
- *Bug Squish* (/usr/games/bugsquish) is a bit mindless but fun all the same. Little bugs drop down and try to land on an arm. Your mission—should you accept it—is to squish as many bugs as you can by clicking your mouse on them. As I said, it's mindless, but it does allow little people to practice their mouse skills while having some fun.
- Calculator (/usr/bin/gcalctool) is the GNOME calculator.
- *Four-in-a-Row/Connect 4* (/usr/games/gnect) is just like the board game. You can play against another human opponent or an increasingly more skillful computer user.
- G Compris (/usr/games/gcompris) has to be the real find of

the experiment. This is a single program that has many, many parts. It is an entire suite of educational tools packaged together and aimed at 3–8 year olds. Within the suite are—among many other things—word and number games, color-matching and memory exercises and geography quizzes. There are loads of educational functionality in G Compris, and it is graduated, which means children cannot proceed to a later exercise until they have mastered the earlier ones. G Compris can be installed in one of a number of languages and has a friendly soundtrack and voice-over. I initially thought Aaron and Aideen would spend a lot of time in this program and was surprised to find Joseph enjoying it too. There's so much in G Compris that it really needs to be experienced to be believed.

- *K Tuberling, the Potato Guy* (/usr/games/ktuberling) is a simple little program that provides a blank picture upon which you can place, for example, ears, eyes, noses, spectacles, hats and hair. The default blank picture is a potato, but a blank Tux also is provided. Aideen loves this program, as do the boys. The boys love it so much that they used *K Tuberling* to create a gallery of Tux and his family. Check out Tux's mother-in-law, as shown in Figure 3.

- *MathWar* (/usr/bin/mathwar) is a simple X-based math-drill program.

- Office Draw (/usr/bin/oodraw) and Office Writer (/usr/bin/oowriter), both part of the OpenOffice.org suite, were included primarily for Aaron, who likes to draw with the computer as well as write short stories and poems. I'd recently convinced the kids' schoolteacher to try OpenOffice.org for Windows in their school, in an effort to fix file format-compatibility problems she was having with the school's existing choice of office suite. So, making OpenOffice.org available on the kids' PC made perfect sense.

- Play a DVD (/usr/bin/xine) allows the kids to view any of the DVDs that they own. To get DVD playing to work on Ubuntu, I had to search Google for the libdvdcss library, which allows for the DVD movie encoding to be deciphered. Once the library was installed, DVD viewing worked. Xine was a big hit, not only because it supports DVD menus and the like but also because it allows viewers to capture snapshots of the currently playing movie. Once he discovered this Xine feature, Joseph wasted no time and created a gallery of snapshots of his current DVD favorite, *The Incredibles*. An added bonus to being able to view DVDs on the new computer is that the main household TV and DVD player are freed-up for Mum and Dad to use. Xine was chosen over the Ubuntu-installed Totem, which did not work as well as Xine in any of my tests.

- Play a Music CD (/usr/bin/gnome-cd) turns the PC into a CD player, with the default GNOME CD player popping up whenever an audio CD is popped in to the CD drive.

- *Super Tux* (/usr/games/supertux) is a classic, Mario-style, jump-and-bump-level game that should be familiar to many readers. Saying that the boys love this game would be a complete understatement: they are totally besotted with it. A little animated Penguin jumps and bumps his way through increasingly difficult levels in search of his goal. The soundtrack to this game is great, as are the effects and configurability. If anything, it's a little too addictive and, of all the programs described in this article, *Super Tux* is the program most likely to be on-screen when I enter the playroom. This has caused Deirdre to worry that the boys are playing it *too much*. However, as the game allows players to design and use their own levels, and as the boys have started to do just that, I've been happy to let *Super Tux* survive. I figure that building a level is the first tentative step toward getting the computer to work the way the kids want it to, which isn't a huge leap away from that other popular customization technique: programming. So, highly addictive or not, *Super Tux* stays for now—unless the boys are cheeky to their Mum, in which case it'll be wiped from the PC faster than they can say "yahtzee!"

- *Tali/Yahtzee* (/usr/games/gtali) is a nice implementation of the classic dice game. The iMac had a great version of this game that the boys always liked to play, and the GNOME version is similar and familiar.

- *Tux Kart* (/usr/games/tuxkart) is an arcade-type racer game. Little Tux sits in a go-kart and races around one of a selection of pre-built tracks. The music is fun, and the game is not too hard to play, which means that even Aideen can play without too much trouble. I've seen some games of this type that take the physics to the extreme, making them incredibly hard to play well. *Tux Kart*, thankfully, does not fall into this category.

- *Tux Paint* (/usr/bin/tuxpaint) is a great kids-targeted drawing program. The sound is great, the effects are wonderful and it is easy to use. Aideen spends more time in *Tux Paint* than in all of the other installed programs combined, and Aaron enjoys using it too. The built-in collection of stamper shapes especially are appreciated by our budding Picassos.

- *Tux Racer* (/usr/games/tuxracer) is the one program that's fired-up and shown-off whenever either of the boys have a friend over to play. *Tux Racer* is, quite simply, one very cool program. Watching Tux slide on his belly at 90km/h in stunning, realistically rendered graphics remains—for me, anyway—one of the best examples of just how far Linux has come as a multimedia platform.

- *Tux Type* (/usr/games/tuxtype) is a fun typing tutor. All three of the kids play it, and Aideen loves the way Tux eats the letters as they drop from the sky and correctly are identified on the keyboard. Aideen especially likes the cartoon-type sound effects and animation that occur when Tux eats a fish at the last possible moment, which usually results in Tux making a mad dash across the screen.

- *X Tux* (/usr/games/xtux) is a 2-D, *Pac Man*-type game that works well and is fun to play. Although not as popular with the boys as *Tux Racer* or *Super Tux*, it still is played quite often.

RoboNexus

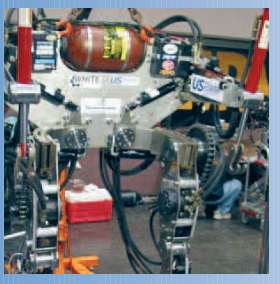
IDG WORLD EXPO

roboticstrends

The Largest Robotics Event in the Western Hemisphere

October 6-9, 2005
San Jose Convention Center, San Jose, CA

EXPO Magazine's best new show 2005 Winner



The International Business Development, Educational and Consumer Event for Personal, Service and Mobile Robotics

- 50 Robotics Visionaries and Thought Leaders
- 50,000 Square Foot Exposition
- 5 Comprehensive Professional Development Conferences:
  - Business Development and Opportunities Conference
  - Emerging Robotics Technologies and Applications Conference
  - Robotics Design, Development and Standards Conference
  - Robotics Education and Instruction Conference
  - Consumer Robotics and Entertainment Event

New for RoboNexus 2005!

- Business-to-Business and Consumer Entertainment Expo Areas
- Service Robotics Summit
- Service Robotics Pavilion
- "Robotics at Home" Demo Stage
- Robotics Innovators Awards –the 'Robi'

www.robonexus.com

Founding Sponsor

iRobot

Premier Sponsor

Premier Media Sponsor

POPULAR science

Gold Sponsors

HiTEC

Corporate Sponsors

evolution robotics

ESS

LEGO education

mBus machineBus.com

OLogic

Openware Robotics

ShopBot

ROGUE robotics

SYSTRONIX Embedded Java Spoken Here

Media Sponsors

BATTERY POWER PRODUCTS & TECHNOLOGY

CIRCUIT CELLAR

e-DRIVE

Entertainment Engineering

GLOBAL FUTURES

VEHICLE

Nuts & Volts

LINUX JOURNAL

NetworkWorld

PC WORLD

Popular Mechanics

TUX

SERVO

Vision Systems DESIGN

Association Sponsors

Acceleration Studies Foundation

GETI

Figure 3. A Rather Cool-Looking Granny Tux

Problems

Thankfully, there are no show-stopper problems to report. The Warty Warthog release of Ubuntu did have some problems with sound. After a restart or a new login, the sound configuration would be lost, resulting in no more sound. Upon investigation, I discovered that the GNOME volume controls were being set automatically to zero. To fix this temporarily, I popped a shortcut to the GNOME Volume Control applet on the desktop and then used it to reset all the volume sliders. This fixed the sound problem, until the next restart or login, of course. I planned to research a permanent fix but then quickly realized that the complaints about the new computer having no sound had stopped. It turned out that Aaron had watched me fiddle with the volume controls, he'd told his siblings what to do, and all three of them had developed the habit of sliding up the volume controls immediately after logging in.

Upgrading Ubuntu

In the last few weeks, I upgraded the PC to the most recent release of Ubuntu, Hoary Hedgehog. This resulted in much merriment, primarily because of the inclusion of a newer release of *Super Tux* that, I'm told, is much better, has improved graphics, animation and sound. Speaking of sound, this Ubuntu release is better but still has a few problems. Any that surfaced were all fixable, permanently, and all I needed to do was search the Ubuntu support wiki for sound and the name of the program that was misbehaving. The fixes found in the wiki worked, and sound is no longer a problem.

With the upgrade, Joseph asked if the shared login ID could be replaced by individual IDs, which I did. This is less to do with privacy and more to do with his little sister's fondness for pressing the Delete key when viewing Joseph's *K Tuberling* Tux family collections. By the way, Tux's family has been extended to include cousins, friends and neighbors.

Once the novelty factor started to wear off, I began to get requests for some of their older software titles. Most of these, despite being targeted to Mac OS, did come in dual-install format, in that they can be installed on Windows too. In an effort to see how much work was involved, I decided to play around with Wine in an attempt to install some of the titles the kids were asking for. After a few hours of research on the Internet and some reading, I spent about a day trying to get the latest release of Wine to work on Ubuntu. I managed to run the installers successfully for a lot of the Windows titles that the kids had, but none of the programs would run properly once installed, so I had to abandon the effort. Since giving up—and since the upgrade to the latest Ubuntu—the requests for the older titles have become less frequent; although Aaron misses one of the freebie, cereal-pack soccer games that he used to play on his Mum's laptop. As I finish off this article, I'm in the process of downloading and evaluating a small collection of Linux soccer games from *The Linux Game Tome*. The *Eat The Whistle* technology looks the most promising. If this does not satisfy Aaron's craving for a soccer game, I plan to dedicate additional time to configuring Wine.

Is Linux Ready for Kids?

The answer is yes, of course it is! It's not that Linux is a better platform than the others for kids to use, it's that Linux is as good as any other. Children are happy to sit down and play with most any computer as long as the software titles provided are engaging and fun. This is true of Linux, Windows and Mac OS. Of course, the point to make is that if Linux is as good as the others, there's nothing stopping anyone from using Linux as a primary OS for children. It's not a case of "is Linux ready for kids?" but rather "why not Linux for kids?"

The Barry household has made the move to Linux and won't be turning back. The wealth of software available on the Internet and within Ubuntu's Debian archives has been only scratched. There are loads out there for me to evaluate and install for my kids as they grow out of the programs they currently are enjoying. If you have any suggestions for programs you think they might like, drop me a note and we'll take a look.

Acknowledgement

Thanks to Peter Garrett from Marcel Gagné's WFTL-LUG mailing list for suggesting I use Nautilus to mimic the Mac OS Launcher application.

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.

Why is LPI the Global Standard in Linux Certification?

Trusted.

All Linux Professional Institute certification programs are created using extensive community input, combined with rigorous psychometric scrutiny and professional delivery. We test the whole continuum of important Linux skills - we don't just focus on small, subjective tasks. LPI exams are not simply an afterthought used to help sell something else. LPI is a non-profit group that does not sell software, training or books. Our programs and policies are designed to meet educational requirements, not marketing.

Accessible.

LPI exams are available in seven languages, at more than 7,000 locations, in more than 100 countries. You take LPI exams when you want, where you want. In addition, special exam lab events around the world make our program even more affordable. And because we don't make exclusive partnerships, LPI is supported by a broad range of testing centers, book publishers and innovative suppliers of preparation materials.

Independent.

You switched to Linux to get away from single-vendor dependence. So why trade one form of vendor lock-in for another? LPI's program follows the LSB specification, so people who pass our tests can work on all major distributions. Because of its strong grass-roots base and corporate support both inside and outside the world of open source, LPI goes beyond "vendor-neutral" to truly address community needs.

LPI is IT certification done RIGHT!

For more information, please contact us at Info@lpi.org or visit us at www.lpi.org.

Linux Professional Institute

Project Utopia

Users—what will they plug in next? Robert is making the computer make sense of hardware, so you don't have to. BY ROBERT LOVE

For the better part of a decade, Linux enthusiasts have waxed poetic on the inherent greatness and looming success of Desktop Linux. Our kernel is so robust! Our applications are infinitely configurable! This is the year of Desktop Linux! Who would ever use Microsoft Windows? These claims and similar—particularly when made back in the 20th century—seem, in retrospect, so trite. Not that I stand righteous—oh no, I laid the praise down as thick as anyone else did. I too was a convert.

At least, I did until I realized that hardware support in Linux was *awful*. Like a deck of cards, my rosy view of Desktop Linux came crashing down, making a 180-degree turn from glowing to ghastly. Heartbroken, I cried myself to sleep every night and went on an inexplicable diet consisting only of cheese and pudding.

But this did not last long. One day, the Linux community decided to do something about it. We outlined a plan not only to reach feature parity with the other desktop operating systems, but also to surpass them. Hardware support has come a long way in the last year and a half. This is the story of just how far.

A Past Since Forgotten

The steps for installing a new hardware peripheral on a Mac might go a bit like this:

- Step 1: plug hardware in to Mac.
- Step 2: begin using hardware.

Most of us would not even include these two items as steps. The first is a physical necessity; the second is the original and ultimate goal. Lost, somewhere between steps one and two, are 39 other steps, right? Kernel modules? Configuration files? Rebooting? Extensive mastery of sed and awk?

At some point in Linux's history, support for new hardware could easily require compiling a new kernel module, becoming root, editing configuration files, loading said module, checking dmesg, cursing, removing the module, unplugging the hardware, plugging the hardware back in, reloading the module and so on.

Forgotten, perhaps clouded by a love for free software and the invigoration of *do-it-yourself*, is the notion that *stuff should just work*. As fun as writing my own kernel module might be—and I use the term fun here loosely—sometimes I just want to plug in my camera, get my photos and be done with it.

A Call for Change

In late 2003, the Linux system was well primed for the emergence of a new architecture for managing hardware on the desktop. The 2.6 Linux kernel was out and rapidly gaining adoption. It brought, among numerous other new features and improvements, a new mechanism for handling device drivers, called the device model. The device model allowed, for the first time, the kernel to build an in-memory tree of the devices it supported. For example, both my mouse and my keyboard are connected to my USB hub, which is connected to my third USB port, which is on my first PCI bus. Such a rich hierarchy provides all sorts of opportunities to the kernel. One of the most promising, however, was sysfs.

sysfs exports this device hierarchy as a filesystem. One directory lists all the buses on a system. For each bus, another directory lists all the devices on a given bus. Files for a given device could link to the associated module. Walking the sysfs tree, therefore, would allow user space to build a comprehensive picture of the system's physical device hierarchy, exactly as the kernel sees it.

That alone is incredibly useful. But another kernel feature, hotplug, broadened the horizon even more. The kernel's hotplug infrastructure notifies user space whenever a device is added to or removed from the system. This allows applications to become aware of changes to sysfs in real time. It also allowed for the creation of udev.

udev is a user-space implementation of devfs—an automated and dynamic manager of device nodes. Instead of a /dev created once, statically, udev updates /dev on the fly, in response to the exact hardware available to the system. More important, however, is that udev places intimate knowledge of devices and their device nodes in user space. Hotplug, sysfs and udev together allow user space a complete view of the system's hardware.

Now user space needed to capitalize on the opportunity.

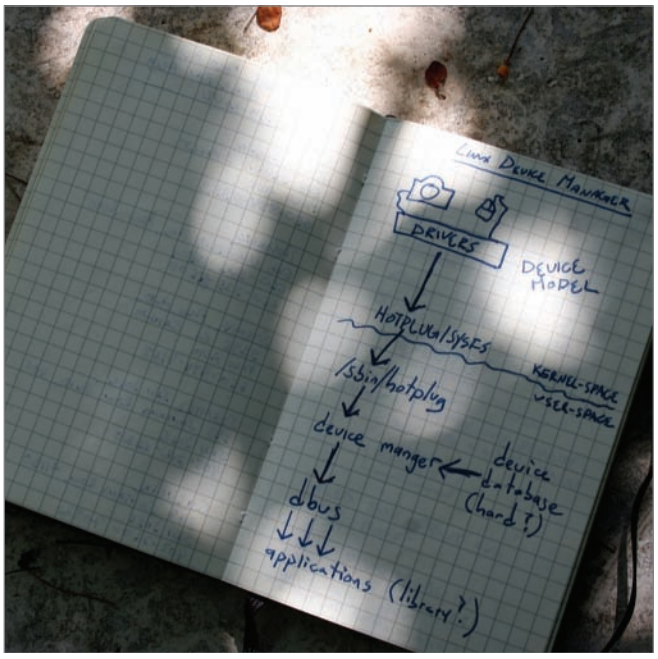


Figure 1. Early Project Utopia Notes

Enter HAL

This was 2003. That summer, I attended a BOF at the Ottawa Linux Symposium on improving the Linux desktop by Robert Sanford Havoc Pennington. In the BOF, Havoc referenced a whitepaper of his entitled "Making Hardware Just Work", in which he unveiled a utopian view of hardware management on the Linux desktop. Intrigued, I took notes—see Figure 1.

We ended up speaking to the group on this utopia and discussing possible implementations. The BOF ended without much traction from the audience, but Havoc and I had a firm understanding of the situation and potential solutions. Other responsibilities kept me from immediately acting on my crude sketches, and so they sat idle on the pages of my notebook.

Two things happened that lifted the pages to life without my immediate realization.

First, David Zeuthen, then living in Copenhagen, decided to bring Havoc's documents to life by beginning the HAL Project. HAL, originally hardware abstraction layer but now not an abstraction of anything whatsoever, is a system-level dæmon that ties together hotplug, sysfs and udev in order to provide a Linux system with a single, comprehensive view of hardware, accessible via a standardized interface. HAL makes it possible for an application to say, "give me the device nodes of all input devices" or to ask, "is there a camera connected to this computer?" With HAL, what was once a hundred lines of hacks, operating on hard-coded device nodes with intimate knowledge of Linux internals, is now a single, elegant HAL request. David's HAL, in effect, brought a 21st-century hardware infrastructure to Linux.

The second disruptive event came in December of the same year, when I accepted a job with Ximian, recently acquired by Novell, as a kernel hacker dedicated to the desktop. My first mission was to figure out the hardware situation. I teamed up with a colleague, the inimitable Joey Shaw, an Ohio native, and we sat down and hashed out our utopian view of hardware management.

Both Joey and I recognized the strong foundation that the 2.6 Linux kernel, sysfs, hotplug, udev and now HAL supplied. We concluded that the missing pieces were the layers on top of HAL. We had a rich infrastructure in place; we just had to do something with it.

HAL uses a then-nascent but always-promising project called D-BUS as its communications mechanism. On one side, D-BUS is a run-of-the-mill interprocess communication (IPC) system—like CORBA, but a lot easier to use. On the other side, however, D-BUS introduces the concept of

the system-wide message bus. In addition to per-user process-to-process communication, D-BUS allows components in a Linux system to send out signals, announcing events or providing information to all who care to listen. Signals can announce when a network connection is obtained or when the laptop battery is running low. Interested applications higher up the stack can listen for these signals and, upon receipt, react.

Our plan was literally to flood the system with D-BUS signals. HAL and other lower-level components of the Linux system were to generate numerous useful signals and have higher-level components respond, evolve and react. In effect, our goal was to make the Linux system much more dynamic and, ultimately, make hardware just work.

A Project All about Utopia

Joey and I decided to create an umbrella project—a meta-project. The plan was to spur development of HAL-aware applications that can provide hardware policy on the desktop. Never should a user need to configure hardware. It should happen automatically in response to the user plugging the hardware in. Never should the user (or even the programmer) have to mess with device nodes and esoteric settings. HAL should provide all of that, on the fly, to the applications. Never should the user have to guess how to use new hardware. If I plug in a camera, my photo application should run. If I insert a DVD, it should start playing. All of this should happen magically, automatically and cleanly.

I coined the name Project Utopia. It was, after all, a bit utopian.

We did not have a central Web site or source repository or cute logo. Project Utopia was a cause and a way of thinking. We had a goal and a set of use cases and a growing disgust toward things not working. We blogged and spoke at conferences and wrote code. One by one, piece by piece, we started to build a set of policy pieces on top of HAL, guided by the following rules:

- Make hardware just work.
- Use HAL, udev, sysfs and 2.6 Linux kernel as our base.
- Tie it all together with D-BUS.
- No polling, no hacks—everything should be event-driven and automatic.
- Carefully divide infrastructure into system and user level.
- System level should be platform-agnostic; user level, GNOME-based.

GNOME Volume Manager

I began writing GNOME Volume Manager in late December 2003. It was originally a proof of concept—a test bed for my ideas. I wanted to see how feasible hardware management on top of HAL could be. The plan was to respond to events such as “new hardware” or “audio CD inserted” with specific actions. GNOME Volume Manager is nothing but a simple finite state machine, receiving hardware-related

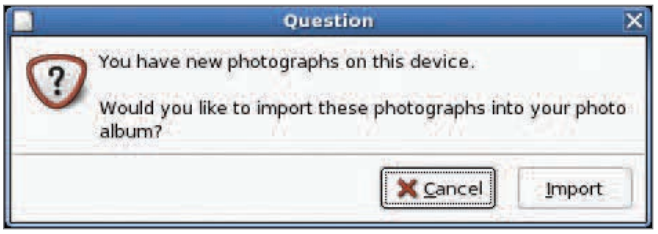


Figure 2. GNOME Volume Manager Prompting on Discovery of New Photos

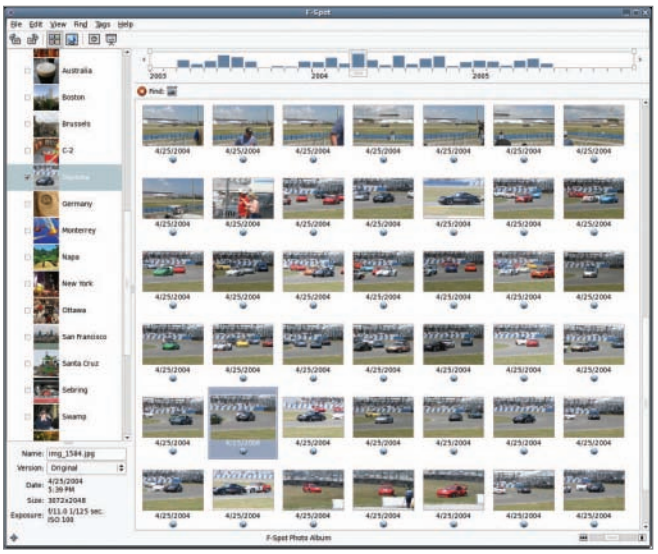


Figure 3. F-Spot, a Photo Management Application

events on one end and replying with hardware-induced actions on the other. The tricky part was to do it all with HAL: no polling, no hacks.

GNOME Volume Manager implemented the Project Utopia policy related to block devices. When the user inserted an audio CD, GNOME Volume Manager would play it. When the user inserted a USB keychain device, GNOME Volume Manager would mount it and open a Nautilus window. When the user plugged in a camera, GNOME Volume Manager would ask if it should automatically import the photos into the user’s photo management application (Figures 2 and 3). A recently added feature even found GNOME Volume Manager managing iPods!

The Rest of the Puzzle

The next step was bringing HAL support to more applications, a process Joey and I call halification. The following months witnessed additional policy pieces, such as automatic printer configuration and seamless network management (Figure 4).

For printers, Joey wrote a HAL back end for CUPS, the Common UNIX Printing System, allowing CUPS to query HAL on the availability of printers. The result: plug in a printer and configure it automatically, on the spot.

The ambitious NetworkManager Project, started by hackers at Red Hat, aimed to solve networking woes. Seth Nickell, an early designer on the project, described the

Figure 4. NetworkManager's Network-Switching Applet

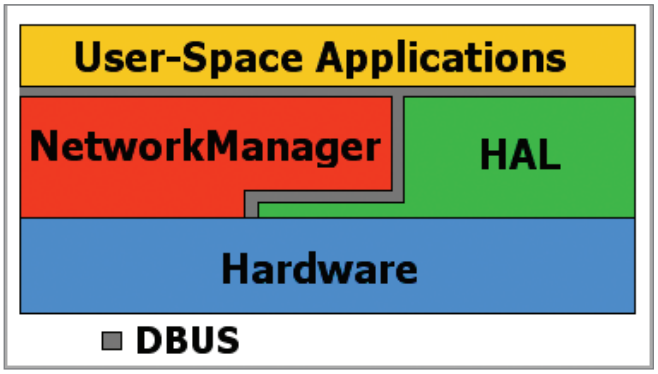


Figure 5. NetworkManager Architecture

intended use case as an electrical outlet: "you plug it in and [it's] on." For example, plug a laptop in to a docking station, and it instantly switches to the station's Ethernet. Walk into your favorite coffee shop and instantly begin using the wireless networking. NetworkManager made networking simple, automatically choosing the optimal solution for networking connectivity.

NetworkManager's architecture is two-part. First, a root-level dæmon sits alongside HAL, responding to HAL events and communicating with the system's networking hardware. Second, one or more user-level components implement policy and provide a user interface. Together, the components provide a complete solution for networking. Figure 5 is a diagram of the architecture.

Today

Today, the Project Utopia mindset continues to foster new applications, interesting hacks and fresh projects aimed at making hardware just work. Linux distributions from Novell, Red Hat and others sport powerful HAL-based infrastructures. The GNOME Project is integrating HAL and D-BUS across the board. The Project Utopia cause is spreading beyond GNOME too, as other platforms implement HAL-based solutions in a similar vein.

Linux development has never stood still, however. Like a rabid cheetah, development sprints forward toward better, faster, simpler solutions. Support for new hardware continues to roll in, and solutions in the spirit of Project Utopia are continually implemented to provide a seamless user experience.

Cute hacks such as having your music player mute when your Bluetooth-enabled cell phone receives a call are not a dream but the reality in which we live. What cute hacks will tomorrow bring? What new hardware will we support next? What application will be halified next? Join in and answer those questions yourself!

Resources for this article: www.linuxjournal.com/article/8459.

Robert Love is a kernel hacker in Novell's Ximian Desktop group and the author of *Linux Kernel Development* (SAMS 2005), now in its second edition. He holds degrees in CS and Mathematics from the University of Florida. Robert lives in Cambridge, Massachusetts.

ospi
Open Source Professional Institute
Preparing Business for Tomorrow

Linux Certification

Train Now for LPIC-1 Linux Certification
Not for the faint of heart...
Obtaining Linux Certification will set you apart from the multitude of Linux IT professionals and prove to potential clients and future employers that you've got what it takes.

~ 2005 ~
10/17 Detroit
10/31 Cincinatti
11/14 Atlanta

~ 2006 ~
1/16 Pittsburg
1/30 Memphis
2/13 Miami

LPIC-1 Training
Mon – Fri 8am – 4pm

Additional Offerings:
Building an Enterprise Mail Server
2 days 5 – 7pm
Building an Enterprise Web Server (LAMP)
2 days 5 – 7pm

Call NOW to Reserve Your Space
1-800-316-7912
www.ospinstitute.com

Supermicro SUPER PDSG4 and SUPER PDSGE

Supermicro Computer announced the launch of Intel dual-core products, the SUPER PDSG4 and SUPER PDSGE motherboards, which support PCI-X 133/100 expansion cards. Based on the Intel 955X chipset, the SUPER PDSG4 ATX form factor board supports one Pentium Processor Extreme Edition, featuring two processing cores with a 1066/800/533MHz system bus. It also offers 8GB of ECC unbuffered DDR2-667/533/400 SDRAM; a user overclock feature in the system BIOS; PCI-Express x16/x1; three 32-bit PCI, two PCI-X 133/100 and four SATA ports (3Gbps); RAID 0, 1, 10 and 5; eight USB 2.0 ports; onboard AC97 audio; single PCI-Express Gigabit LAN; and U320 single-channel SCSI. The SUPER PDSGE is based on the 945G/P Express and supports one Pentium D processor, featuring two processing cores with a 1066/800/533MHz system bus; 4GB of unbuffered DDR2-667/533/400 SDRAM; a user overclock feature in the system BIOS; PCI-Express 1x16/2x1; four 32-bit PCI and four SATA ports (3Gbps); eight USB 2.0 ports; onboard AC97 audio; single PCI-Express Gigabit LAN; and integrated Gfx graphics. Both boards are RoHS-compliant, lead-free and optimized for Supermicro's SC733T-645 and SC733i-645 mid-tower chassis.

CONTACT Supermicro Computer, Inc., 980 Rock Avenue, San Jose, California 95131, 408-503-8000, www.supermicro.com.

PetaBox

Capricorn Technologies introduced the PetaBox Product Family, designed for massive data storage. The PetaBox supports petabyte-class storage with state-of-the-art density, low power consumption and a low total cost of ownership. The PetaBox is scalable from individual terabyte nodes to a full petabyte cluster. A single 19-inch rack can support up to 64TB of raw disk space, a density achieved through a design that consumes as little as 50 watts per terabyte. Four models of PetaBox currently are available: the GB1000, a 1.0TB node; the GB1600, a 1.6TB node; the TB40, a 40TB rack; and the TB64, a 64TB rack. Each node has four hard drives per node, an ATA interface, rotational vibration compensation, 8MB of cache, 8.5ms of typical latency and an EZ-Latch disk mounting system. Nodes also feature a 1GHz VIA C3 CPU, up to 1GB of DDR266 RAM, two USB 2.0 ports, 10/100 or 10/100/1000 Ethernet and an optional 16x2 character LCD.

CONTACT Capricorn Technologies, 1021 Mission Street, San Francisco, California 94103, 415-722-2149, www.capricorn-tech.com.

Qt 4

Trolltech released version 4 of its Qt cross-platform development software. New features for Qt 4 include improved heavy-duty graphics capabilities. Qt's painter now supports semi-transparency, anti-aliasing, optional floating-point coordinate system, painter paths and gradients. Support for interchangeable underlying paint engines and off-screen rendering also has been added. Trolltech also extended Qt's multithreading capabilities, along with its database integration and XML support for building both desktop and server-side applications. In addition, Qt 4 offers seamless integration with Microsoft Visual Studio .NET, allowing Visual Studio .NET developers to create applications that can run on Linux, Mac OS and other desktop platforms. Furthermore, Qt 4 for Microsoft Windows is available under the GPL. Three editions of Qt 4 are available: Qt Console, Qt Desktop and Qt Desktop Light.

CONTACT Trolltech, Inc., 1860 Embarcadero Road, Suite 100, Palo Alto, California 94303, 650-813-1676, www.trolltech.com.

Equalizer SI Series

Coyote Point recently introduced a new line of Web server performance appliances designed to address core availability and performance requirements of Web sites and server farms. The Equalizer SI Series systems integrate an enhanced version of Coyote Point's adaptive server load-balancing and traffic management software, consolidated switch intelligence and Intel processor-based performance. Three systems are available: the enterprise-class E450si, the mid-range E350si and the entry-level E250si. Their features include consolidated switching capacities of up to 16 ports; support for up to 8,000,000 concurrent connections; incrementally scalable load balancing and traffic management for an unlimited number of virtual servers and up to 64 servers per cluster; adaptive protection against DoS attacks; built-in Flash memory for zero-downtime reliability; SSL acceleration for up to 4,000 encrypted transactions per second; and an enhanced Web interface for point-and-click operation of configuration options.

CONTACT Coyote Point Systems, Inc., 675 North First Street, Suite 975, San Jose, California 95112.

Motorola E895

Motorola announced the release of the E895 multimedia clam-shell handset, built on Motorola's EDGE technology, Linux and Java. The E895 offers a suite of intuitive multimedia tools, including a 1.3 megapixel camera, video record and playback options and an optional removable memory. E895 features include 3GPP video streaming and the ability to view files as they download, Bluetooth wireless technology and Bluetooth Stereo Headset, SyncML, streaming audio, shared media player with multiple audio codecs and stereo through the enhanced mini-USB headset jack, up to 10MB of embedded memory, and TransFlash removable memory for up to 512MB of optional memory. The E895 also offers a full HTML/XHTML Web browser, multimedia messaging service (MMS) and instant messaging.

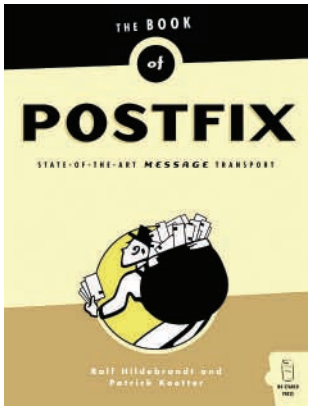
CONTACT Motorola, Inc., www.motoinfo.

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.

The Book of Postfix

by Ralf Hildebrandt and Patrick Koetter

No Starch Press, 2005 | ISBN 1-59327-001-1 | $44.95 US



Anyone who has tried to run an e-mail server knows that mail isn't a polite relay race anymore. It's a game of smash-mouth football. Nine out of ten times someone opens an SMTP connection to you, it's not with something you want. And as if coping with spammers, viruses and other people's misconfigured mail software wasn't enough, now e-mail is a mission-critical company IT service and is expected to plug in to the LDAP directory. We can't blame you if you decide to outsource mail entirely.

If you do decide to stay on as postmaster@ and fight it out, whatever you do, don't try it with one of last decade's mail books. Although any of the current mail servers, correctly configured, can put up a good fight against the spammers and other bad people, *The Book of Postfix* by itself is a good reason to make Postfix your mail server of choice. Look here for a good explanation of the SMTP protocol, essential for any mail admin, along with enough detail on the architecture of Postfix to help you really understand the config files. It also offers real-world advice for putting together a mail server setup that is reliable in the face of the spam and virus blitz.

Postfix offers you a lot of choice in where to add filters, sanity checks and other protective countermeasures to your mail server. For example, do you want to set up a content_filter or an smtpd_proxy_filter? Besides offering a cookbook for each solution, *The Book of Postfix* helps you consider the pros and cons of each feature you're considering. A helpful plus is diagrams illustrating exactly where countermeasures fit into the Postfix architecture.

Postfix is complicated enough on its own, as it divides functionality among multiple processes for security. In order to add spam-fighting tools and have everything work, you need a good understanding of what plugs in to what and how, and this book is a great way to get it.

Downloadable scripts and errata, some of which could save you a late night of troubleshooting, are available at www.postfix-book.com.

—DON MARTI

TEN YEARS AGO IN *LJ*

October 1995

The October 1995 issue covered "Text Processing", and feature articles introduced groff, LaTeX and Linuxdoc-SGML, which was an early document format at the Linux Documentation Project. All three document formats are still in use today.

Making the transition to 64 bits is IT news today, but it was a hot topic for us ten years ago. Jon "maddog" Hall, then still at Digital, covered Linux on Alpha and its advantages for computer science education:

> Over time, this meant that to get all the sources to our Unix products, 15 separate licenses were necessary, at a cost of thousands of dollars, and even then the sources were restricted to a "need to know" basis and were not for consumption by curious students.

Publisher Phil Hughes, in a "Stop the Presses" item, pointed out that Microsoft Windows 95 overwrites a PC's Master Boot Record—the first Windows version to do so and a FAQ for dual-booters ever since. More help came in the form of an ad for the "System Commander" boot manager, which offered an easy solution for multi-OS systems, with the bonus feature of fixing boot sector virus infections.

—DON MARTI

Building a Call Center with LTSP and Soft Phones

Need to equip an office with terminals and phones, all on a small budget? With LTSP and KPhone, you can do it with only terminals, sound cards and headsets. BY MICHAEL GEORGE

A new customer approached us with a need to provision the office. The customer was receptive to open-source software and was interested in using Linux. Being a nonprofit organization, the budget for the project was tight.

We provisioned the new office with a server running software from the Linux Terminal Server Project (LTSP) to make the desktop economical from the start. We then installed an Asterisk server as a PBX for the call center. To make things easier for the staff, we wanted to have a working soft phone on their terminals with headsets for hands-free operation.

This article discusses the installation and use of the LTSP build environment to build Qt and KPhone so the staff members could run KPhone locally on their terminals. I do not discuss the installation of Linux or Asterisk here, but I have included the relevant context for KPhone, which resides in the Asterisk sip.conf file. We used Gentoo for this particular LTSP server, but any Linux distribution can do the job.

Software Needed

The main software packages needed for this project were LTSP, KPhone and the LTSP build environment (LBE). LTSP easily provides thin-client access to a main server. We often recommend LTSP as an economical way to equip an office, because it focuses monetary resources on the main server rather than on the individual stations. The incremental cost of adding a new user to the office is relatively small, and administration is simplified.

The customer's new office is intended to be a small call center, so hands-free phone operation is a big benefit. We wanted to try using headsets and amplifiers that use a computer sound card for their connectivity rather than hardware phones. These headsets, coupled with software SIP phones on each user's local station, allowed us to meet their phone needs without having to buy separate phone equipment.

Because we already were using Asterisk (see the on-line Resources) as the PBX for the office, it seemed logical to use an open-source software phone. We decided to use KPhone (see Resources) as the software SIP phone, because it had proven reliable on standalone systems previously tested. One of the drawbacks of every SIP soft-phone package we investigated at the time was none supported a network-enabled sound protocol. As a result, they were required to run locally on the station that physically has the sound card. As these stations are thin clients that boot from the main server, KPhone needs to be resident in the filesystem on each station. When a user runs KPhone from the desktop, which runs on the server, the KPhone process needs to start in the local terminal environment.

KPhone is not a standard part of the LTSP package, so we needed to build it inside the local stations' root filesystem that is NFS-mounted from the server at boot time. Building software for the terminals' root filesystem requires LBE (see Resources). Building software in LBE also requires that all necessary libraries be present in the filesystem. One of the other benefits of KPhone is that the Qt library is the only library required beyond those already in LTSP.

Installation and configuration of LTSP are detailed in the LTSP documentation (see Resources). One deviation from the standard install of LTSP is that the DHCP configuration file must reference the root filesystem that LBE builds rather than the root filesystem installed with the LTSP package (Listing 1).

Listing 1. Our LTSP Section of dhcpd.conf

```
# LTSP Path Options
option root-path
   "192.168.42.254:/usr/local/src/lbe/opt/ltsp/i386";

#LTSP boot image (relative to the TFTP root)
filename
   "/pxe2/pxelinux.0";
```

Technically, we did not need the LTSP package because LBE includes the necessary boot image and root filesystem. However, if you are not already familiar with LTSP, I recommend you install that package first and get it operational. Deploying LTSP involves the configuration of other standard software included with almost all Linux distributions: DHCP for assigning IP addresses, boot images and root filesystem information for the stations; TFTP for client stations to retrieve their boot images; and NFS for thin clients to remote-mount their root filesystems and the /home filesystem for running remote applications. Installing LTSP provides demo configurations for all of these packages that make setup much easier for a novice.

The main LTSP documentation describes well most of the preparation for running applications locally on the clients. Their installation and configuration also are covered on the LTSP 4.1 Web page. In addition to the software mentioned above, you also need to configure SSH client and

SugarCRM™ solves all these problems that proprietary and hosted solutions created.

Get hooked now. Implement a Linux-based CRM system in under 15 minutes.

Start using SugarCRM today.

Open Source Customer Relationship Management

SUGARCRM™

www.sugarcrm.com/swap

1.877.842.7276
+1.408.454.6941

Copyright © 2005 SugarCRM, Inc. All rights reserved. SugarCRM and the SugarCRM logo are trademarks of SugarCRM, Inc. in the United States, the European Union and other countries. Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

NIS on the server.

SSH is the means we used for starting the process on the remote client. Notice that the LTSP 4.1 documentation demonstrates the use of rsh for launching the applications. Although that would work, the required dæmons for rsh no longer are part of the LTSP package. SSH is now the norm for launching local applications. You can find information about preparing for SSH launching of local applications in the Local Applications section of the LBE documentation.

NIS is needed because the thin clients need to authenticate users through SSH as they launch the applications. NIS configuration is guided by the NIS HOWTO. One item that was not immediately obvious from the documentation was that NIS would complain that /etc/publickey was not present. Creating that file with `touch /etc/publickey` solved the problem.

Once all the supporting software is in place, configuring LTSP to run local applications is easy: set `LOCAL_APPS = Y` in /etc/lts.conf within the LTSP root filesystem. This causes the clients to mount the /home directory from the server with NFS. Also, NIS is made active by /var/yp/nicknames, /etc/yp/conf being created on the clients, domainname being run with the value of the NIS_DOMAIN entry in the lts.conf file and ypbind being run. The sshd dæmon also is activated on the client.

For SSH operations to be transparent to users, we need SSH keys created without expecting users to do it themselves. To accomplish that, we installed superadduser in Gentoo, which is reported to be adduser from Slackware (see Resources) and modified it to generate the SSH keys automatically for the user when the user is created (Listing 2).

Listing 2. Additions to /usr/sbin/superadduser

```
# su to the user and generate their SSH keys
su - "$LOGIN" -c "ssh-keygen -q -t dsa -C '' -N '' -f "$HME"/.ssh/id_dsa"
#
# cp the new public key to the authorized_keys file
cp "$HME"/.ssh/id_dsa.pub "$HME"/.ssh/authorized_keys
chown "$LOGIN":"$( echo $GID | awk '{print $2}')" "$HME"/.ssh/authorized_keys
#
# update the NIS stuff
(cd /var/yp; make > /dev/null)
```

Aside from configuring local applications to run on the client terminals, we also need to make sure the sound cards are active when the thin clients boot. Normally, one would set `SOUND = Y`, `SOUND_DAEMON = <nasd or esd>`, `VOLUME = <default volume level>` and possibly `SMODULE_01 = <ISA configuration string>`. However, doing so not only causes the sound driver to be loaded into the kernel, but it also starts the sound dæmon, which we do not want. We need the sound card to be available for KPhone when it starts on the terminal.

What we do instead is set `SOUND = N` to keep the normal sound system from being activated and `MODULE_01 = <kernel module for the PCI soundcard>`, because LTSP does not have isapnp support, so audio needs a PCI audio device. We also set `RCFILE_10 = "kphone"` to run the initial configuration script to ready the system for KPhone by using the audio device. Then, we put the KPhone script (Listing 3) in /etc/rc.d in the clients' root filesystem to enable access to the /dev/sound/* files. -rwrwrw access is not the most secure, but because only one user is running processes on the terminal at a time, it works fine. Finally, we turn on the microphone and adjust the gain and volume levels.

Listing 3. <LTSP root>/etc/rc.d/kphone

```
#!/bin/bash
echo Setting up the system for using kphone locally

echo change the permissions on the audio files...
/bin/chmod 666 /dev/sound/*

echo Turn on the microphone, adjust gain and volume
/bin/aumix-minimal -m R

echo Turn gain and volumes up to maximum
/bin/aumix-minimal -m 85
/bin/aumix-minimal -p 100
/bin/aumix-minimal -v 100
```

Building Qt and KPhone

Now that you have the LTSP environment configured and operational, you can build the LBE. Getting LBE from CVS is as simple as:

```
cvs -d :pserver:anonymous@cvs.ltsp.org:/usr/local/cvsroot checkout -s
```

You then need to `su` to root—using `sudo` with the LBE doesn't reliably work—and run `./build_all`. You can take a break here, as the build of LTSP in LBE takes some time to complete.

Once you have the new root filesystem for the terminals built, change your DHCP configuration to refer to that boot image and root filesystem, and restart your DHCP server. You probably want to move /etc/lts.conf from your old LTSP root filesystem to the new one. You also should move the system-wide SSH known-host keys—the ones you created as per the Local Applications section of the LBE document—to the new filesystem.

Now we need to build the Qt libraries and then KPhone inside the clients' root filesystem. The LTSP build environment (LBE) makes this much more manageable. Adding packages for building in the environment amounts to creating a package.def file in a directory named for the package. The package.def files describe how to get, verify the download, unpack, configure, build and install the package software. The build script in the ltsp-src directory then does a chroot and executes the build process.

Through trial and error and discussions on the LTSP IRC channel (see Resources), we were able to construct the required package.def files (see Resources for those files). Constructing the package.def file for building Qt, in ltsp-src/qt under the LBE root, was a straightforward process. Each build exported the same variables to the build environment. Notice also that threading is turned on explicitly at the CONFIGURE stage.

KPhone builds much more easily if Qt has threading enabled, but it is not enabled by default in Qt.

Building KPhone was a bit more complicated. The package.def file (see Resources) works well enough, but the x-includes configuration option does not seem to change the resulting Makefiles. This would cause compilation errors when building trayicon.cpp. Manually adding `-I/usr/X11R6/include` to CFLAGS in kphone/kphone/kphone/Makefile (Listing 4) after the configuration stage seemed to fix the problem, however. The steps to build KPhone in LBE are then:

```
ltsp-src# ./build --configure --only=kphone
ltsp-src# vi kphone/kphone/kphone/Makefile
   (Add "-I/usr/X11R6/include" to CFLAGS)
ltsp-src# ./build --only=kphone
```

We also noticed that the icons were not being located properly by KPhone at first. Making a link to ../../share/kphone in opt/ltsp/i386/usr/share from the LBE root—/usr/share from the clients' root—allowed KPhone to find the icons correctly.

Listing 4. Modified CFLAGS in kphone/kphone/kphone/Makefile

```
CFLAGS=-I/usr/X11R6/include -I/usr/qt/3/include \
   -Wall -O3 -I. -I../gsm -I../ilbc -I../dissipate2 \
   -D_REENTRANT=1 -DQT_THREAD_SUPPORT=1 \
   -DHAVE_LIBX11=1 -DHAVE_LIBXEXT=1 -DHAVE_LIBXT=1 \
   -DHAVE_LIBICE=1 -DHAVE_LIBSM=1 -DHAVE_LIBPNG=1 \
   -DSTDC_HEADERS=1 -DHAVE_FCNTL_H=1 \
   -DHAVE_SYS_IOCTL_H=1 -DHAVE_UNISTD_H=1 \
   -DHAVE_SELECT=1 -DINCLUDE_STDLIB_H=1 $(MOREDEFS)
```

To run KPhone, we put a script in /usr/bin on the terminal server called kphone (Listing 5). This script simply opens access to the X server, determines the terminal at which the user is sitting and starts the KPhone process on that terminal.

Listing 5. KPhone Script on the Server

```
#!/bin/bash
xhost + > /dev/null
HOST=`echo $DISPLAY | awk -F: '{ print $1 }'`
export HOST
ssh ${HOST} env DISPLAY=:0.0 /bin/kphone
```

To make things easier for the users, we created an entry in the KMenu for KPhone that they can select or move onto their docks if they wish. This entry is created by adding the file kphone.desktop (Listing 6) to /usr/kde/3.3/share/applications/kde on the terminal server.

The user then can select the KPhone menu item and launch KPhone (Figure 1). The first time the application is run, the user has to select File→Identity to open the Identity dialog (Figure 2) and enter the connection information. The data to enter here must match that information for the SIP accounts on the VoIP server (Asterisk in our case). Because KPhone stores its configuration in the user's home directory, it needs be con-

ASA COMPUTERS

Want your business to be more productive?

The ASA Servers powered by the **Intel® Xeon™ Processor** provides the quality and dependability to keep up with your growing business.

Hardware Systems For The Open Source Community–Since 1989

(Linux, FreeBSD, NetBSD, OpenBSD, Solaris, MS, etc.)

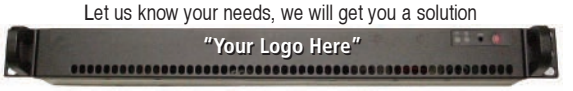
6TB + in 5U—$8,450
Intel 7501, Dual Intel® Xeon™ 2.4GHz
512 MB DDR ECC RAM *Max: 8GB*
6TB + IDE Storage
Dual Gigabit LAN, CD+FD, VGA
Options: SATA Drives, Firewire, DVD+RW, CD+RW, 64 Bit OS Configurations, etc.

1U Dual Itanium IDE—$3,925
Dual Intel® Itanium®2 1.4 Ghz
2 GB ECC DDR
1 of 4 x 40 GB HDD
Dual Gigabit LAN
Based on Supermicro 6113M-i

14" Deep Appliance Server—$865
Intel® Xeon™ 2.4 Ghz Processor
40 GB Hard Drive, One GigE
Options: CD, FD, 2nd HD, Your Logo on Bezel
Call for Low Cost Options.

1U Dual Xeon™ EM64T Superserver—$1,799
SuperMicro 6014H-82 Barebones
1 of 2 Intel® Xeon™ 2.8 GHz 800 FSB
1 GB DDR II-400 RAM *Max: 16GB*
36 GB 10K RPM SCSI *Max: 4 HS HDD*
CD+FD, Dual GigE, VGA, RAILS
Options: RAID, etc.

Your Custom Appliance Solution
Let us know your needs, we will get you a solution



ASA Colocation
$50 per month for 1U Rack - 20 GB/month

ASA Colocation Special
First month of colocation free.*

Storage Solutions
IDE, SCSI, Fiber RAID solutions
TB storage options
3Ware, Promise, Adaptec, JMR, Kingston/Storcase solutions

Clusters
Rackmount and Desktop nodes
HP, Intel, 3Com, Cisco switches
KVM or Cyclades Terminal Server
APC or Generic racks

All systems installed and tested with user's choice of Linux distribution (free). ASA Colocation—$50 per month

2354 Calle Del Mundo,
Santa Clara, CA 95054
www.asacomputers.com
Email: sales@asacomputers.com
P: 1-800-REAL-PCS | FAX: 408-654-2910

intel inside XEON™

Intel®, Intel® Xeon™, Intel Inside®, Intel® Itanium® and the Intel Inside® logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Prices and availability subject to change without notice. Not responsible for typographical errors.

Listing 6. kphone.desktop

```
[Desktop Entry]
Comment=
Exec=kphone
GenericName=Office Telephone
Icon=/usr/kde/3.3/share/icons/Locolor/32x32/apps/kab.p
ng
Name=kphone
Path=
StartupNotify=true
Terminal=0
TerminalOptions=
Type=Application
Categories=Qt;KDE;Office
X-KDE-SubstituteUID=false
X-KDE-Username=
```

Figure 1. The user's desktop environment runs on the LTSP server, but KPhone runs locally.

figured only the first time the user starts KPhone. Because /home is NFS-mounted from the server, the station where users log in is their phone, so the phone effectively follows them if they should change workstations. Once users have registered with the server, they can make calls from the call dialog and DTMF panel (Figure 3).

Initially we had KPhone running, but the response time for any action was horrible. Any time the user would perform an action that caused an SIP message to be sent—dial a number, press a phone button on an active call, answer or hang up the phone—it would take nearly a minute for the action to occur.

We determined that this problem was occurring because of a DNS name resolution issue that was waiting to time-out. The solution was to put entries into /etc/hosts for each of the stations that would be running KPhone, install dnsmasq on the terminal server and have the terminals reference the terminal server as their DNS server, configured

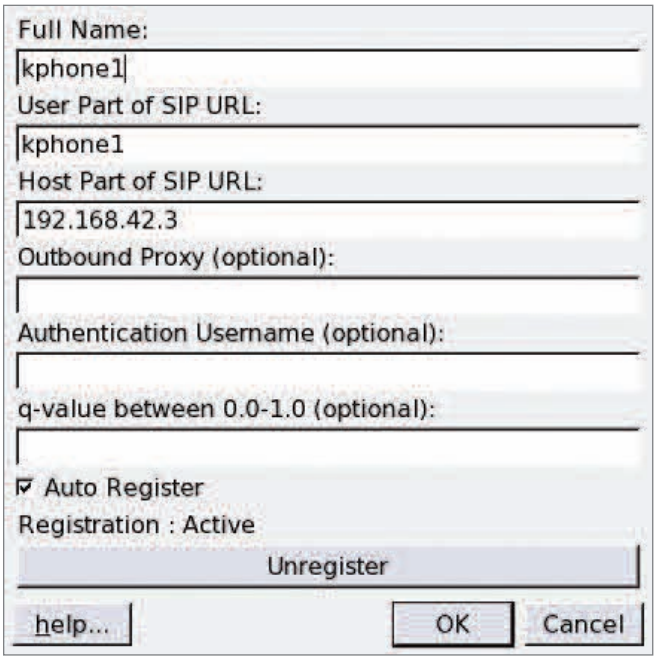


Figure 2. When first running KPhone for a new user, enter the information for the Asterisk server.

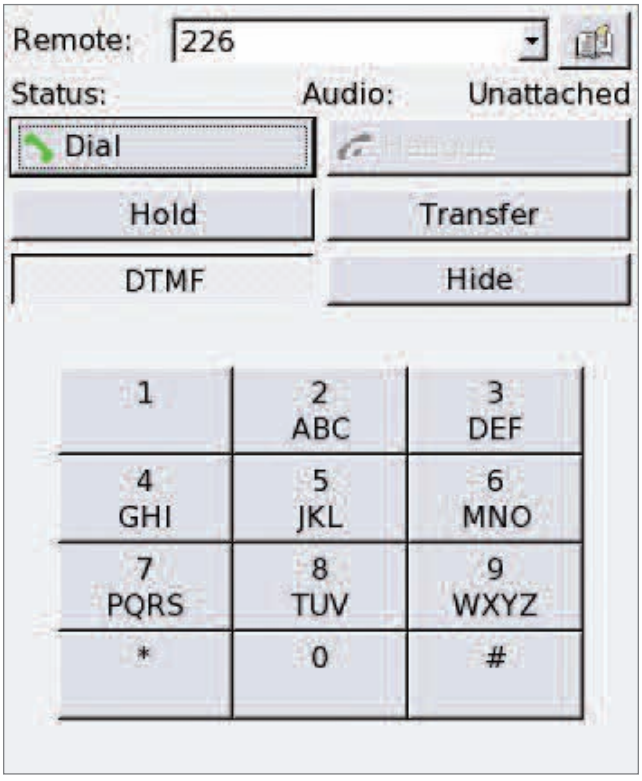


Figure 3. The KPhone call dialog works like a hardware phone.

in dhcp.conf. There are other, perhaps better, ways to solve this issue, but this solution took minimal time to configure and run, and it worked. Finding the source of the problem was the hard part.

Gotchas

There have been a couple drawbacks to this system. Occasionally KPhone closes for no given reason, which can be quite annoying. We have not yet determined the cause of this problem, and we hoped that upgrading KPhone to 4.1 might help.

The KPhone package.def file contains the necessary lines for building KPhone 4.1.1. The change to the Makefile mentioned above for 4.0.5 still applies as of 4.1.1. Our preliminary tests indicate, however, that 4.1.1 has the same problem of closing suddenly for an unknown reason. We have inquired with the maintainers of KPhone to see if they can help, but so far we do not know the cause of the problem.

Another drawback is that when the phone rings, it rings through the headset and gives a visual alert on the screen. If users are not in front of their terminals with their headsets on, they will not know that their phones are ringing. Once the call center is in full operation, operators probably will spend most of their time at the terminals, so this may not be a problem.

Conclusion

We now have KPhone installed and able to be run from any terminal attached to the LTSP server. Adding another user is as simple as creating an account for them on the server, adding an SIP phone entry for them on the phone system and having them configure KPhone. The terminal server is the single point of maintenance for everyone's desktops. Even though KPhone runs locally on each terminal, the LTSP build environment is the single point of maintenance for all of them.

The cost for the system is concentrated in the terminal server and phone system. The incremental cost for each new user is the cost of a low-end terminal and a sound card headset. This expense is much more cost effective than putting a full workstation at each desktop along with a headset-capable hard phone.

Acknowledgements

Thanks to James McQuillan at the Linux Terminal Server Project for his excellent documentation (LTSP and LBS) and everyone on the LTSP IRC channels that helped me get KPhone running locally. Also thanks to Thorsten Kukuk for authoring "The Linux NIS(YP)/NYS/NIS+ HOWTO". Thanks to Mark Spencer, Digium and everyone involved with the Asterisk Project who have made open-source telephony a reality, as well as the author(s) of KPhone. And thanks to Trolltech, the creators of the Qt application framework.

Resources for this article: www.linuxjournal.com/article/8460.

Michael earned his degrees in Computer Science from Michigan Technological University and Purdue University more years ago than he likes to admit. He now lives in rural Michigan with his wife and five children. He has been using Linux since 1994 and now works with Ideal Solution, finding new and creative ways to put open-source software, including LTSP and Asterisk, to work for clients. He can be reached at george@idealso.com.

ASA COMPUTERS

www.asacomputers.com

1-800-REAL-PCS

Hardware Systems For The Open Source Community–Since 1989

(Linux, FreeBSD, NetBSD, OpenBSD, Solaris, MS, etc.)

The AMD Opteron™ processors deliver high-performance, scalable server solutions for the most advanced applications. Run both 32- and 64-bit applications simultaneously

AMD Opteron™ Value Server—$795

- 1U 14.3" Deep
- AMD Opteron™ 240
- 512MB RAM *Max 8GB*
- 40GB IDE HDD
- 2x 10/100/1000 NIC
- Options: CD, FD or 2nd HD, RAID

Front I/O Dual AMD Opteron™ Cluster Node—$1,600

- 1U Dual AMD Opteron™ Capable Font I/O
- Single 240 AMD Opteron™
- 1GB RAM *Max RAM 16GB*
- 80GB HDD
- Dual PCI Expansion Slot

8 Hot Swap Bays in 2U AMD Opteron™—$1,950

- 1 of 2 AMD Opteron™ 240
- 512MB RAM *Max 16GB*
- 3x80GB IDE RAID # 5
- 2xGigE, CD+FD
- Options: SATA/SCSI, Redundant PS

No Frills AMD Opteron™ Storage Server—$8,450

- 6TB+ IDE/SATA Storage in 5U
- Dual AMD Opteron™ 240
- 512MB RAM
- 6TB IDE Storage
- Dual GigE, CD
- Options: SATA HDD, DVD+RW etc.

Your Custom Appliance Solution

Let us know your needs, we will get you a solution

Custom Server, Storage, Cluster, etc. Solutions

Please contact us for all type of SCSI to SCSI, Fibre to SATA, SAN Storage Solutions and other hardware needs.

AMD 64 Opteron

2354 Calle Del Mundo, Santa Clara, CA 95054

www.asacomputers.com

Email: sales@asacomputers.com

P: 1-800-REAL-PCS | FAX: 408-654-2910

Prices and availability subject to change without notice. Not responsible for typographical errors. All brand names and logos are trademark of their respective companies.

Dirt-Cheap 3-D Spatial Audio

With the addition of free audio software, an ordinary inexpensive surround sound card becomes the basis for a 3-D cube for simulation, visualization or gaming.

BY ERIC KLEIN, GREG S. SCHMIDT, ERIK B. TOMLIN AND DENNIS G. BROWN

Many computer systems set up for advanced gaming include Dolby Surround Sound. The typical speaker configurations are 4.1—four speakers and one subwoofer— 5.1 and 7.1. This system is designed for all speakers to be located on a plane centered at the listener, and thus it is not possible to have a sound truly be emitted from above or below the listener, although some systems attempt to simulate that effect. Imagine a game scenario where a monster is climbing down a wall above and behind the player while, at the same time, a mouse is scrambling across the floor behind the listener. In a planar surround system, the sound effects for both the monster and the mouse would come from the rear speakers, making it hard to distinguish the actual locations of the sound sources.

With true 3-D spatial audio, the monster's sound effects could be played from speakers located to the back upper-left and the mouse's sound from speakers located to the back lower-left and back lower-right. In this setup, the player has a much better feel for what is creating the sound and where the sound is coming from. Now the player can arm the rocket launcher and turn toward the back upper-left directly and blast the monster—no need to aim toward the harmless mouse.

Spatial sound has been available for several years and primarily is employed in immersive virtual environments. The systems are not in mass-scale production and often must be installed by professionals, making them costly and out of the reach of most home users. We have devised a low-cost true 3-D spatial audio solution that requires only inexpensive consumer-level hardware and open-source software. This solution allows for the arbitrary placement of speakers, not necessarily co-planar as in other systems. Our 3-D spatial audio solution is the first that we are aware of that provides true 3-D sound at such a low cost.

Background on 3-D Spatial Audio

Preliminary technology for 3-D spatial audio, Fantasound, first was developed in the late 1930s by Disney for the movie industry. Over the years, a great deal of work has been done to advance the field, especially by Dolby Laboratories. In the last few decades, researchers enabled personal computers to emit spatial audio. Today, spatial audio is commonplace in modern computer games. Home systems typically use headphones or a planar array of speakers, usually in a preset configuration, such as Dolby Surround Sound 5.1.

Headphones present a unique opportunity to provide inexpensive 3-D audio. Algorithms that use head-related transfer functions (HRTFs) can create convincing 3-D spatial audio on headphones using a simple stereo sound card. HRTFs use data about how sound is transformed by the user's body, especially the shape of the ears, for mapping sounds with 3-D positional sources. The technique relies heavily on applying different time delays for each ear. Ultimately, we decided not to use headphones, because we needed a system that scaled easily to many users. It was far more practical and cost efficient to use speakers.

A number of high-cost professional-grade hardware packages are available, such as the RME Hammerfall series, M-Audio Delta series and Lake Audio, that provide true 3-D spatial audio. Each package has a cost exceeding $1,000 US, boasts high sound quality and has a large array of features aimed at the professional market. Although the acoustic quality of these packages undoubtedly is higher than that of our low-cost 3-D audio solution in terms of audio clarity and fidelity, both options provide true 3-D spatial audio.

When we started putting together a spatial audio system, no inexpensive hardware and software combination existed to produce true 3-D spatial audio. Although there are software APIs that allow arbitrary, not necessarily co-planar positioning of sound sources, such as Microsoft DirectSound and the Advanced Linux Sound Architecture (ALSA), the low-level drivers officially support only the co-planar 4.1, 5.1 and 7.1 speaker positions mentioned earlier. There is no way to tell the drivers that the speakers have been moved to an alternate configuration, for example, with speakers above or below the listener. So even though software developers could position a sound above or below the user's head, the low-level drivers still assumed the sound was emitted in a circle around the user's head. The bulk of true 3-D spatial audio support comes from customized APIs.

Tommi Ilmonen at the Helsinki University of Technology (HUT) developed a 3-D spatial audio API called Mustajuuri that is built on the ALSA drivers. The Mustajuuri API implements Vector Base Amplitude Panning (VBAP), introduced by Ville Pulkki (see the on-line Resources), as the underlying 3-D spatial audio model. In short, VBAP is the algorithm responsible for moving a sound across a 3-D array of speakers and making the sound appear to come from a specific direction. VBAP selects the three speakers closest to the virtual sound position and calculates the required volumes for each speaker. See Figure 1 for an example of how VBAP works. Mustajuuri also simulates depth for audio by using time delays and distance attenuation. This makes it possible to position a sound anywhere in space relative to the listener. Mustajuuri already has been used to produce 3-D spatial audio using high-end audio cards, but up until now it has not supported low-end audio cards.

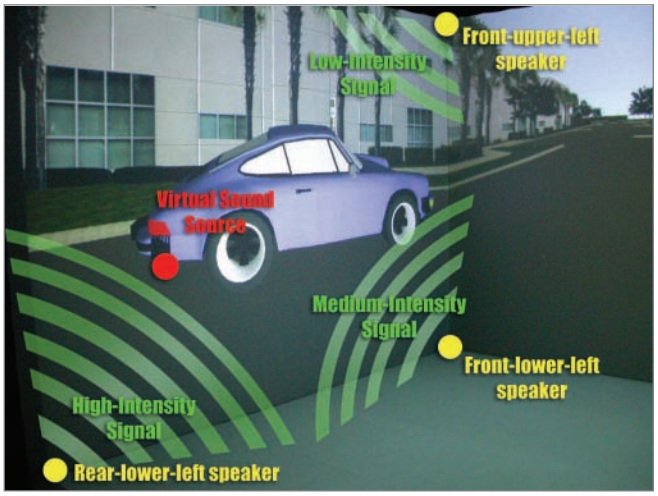


Figure 1. View of 3-D spatial audio test case in the immersive room. Visual depictions show from which speakers the sound is coming for the current view. For each sound, the three speakers closest to the virtual sound source are used to play the sound. Their volumes are varied based on the distance from the speaker and a number of other factors.

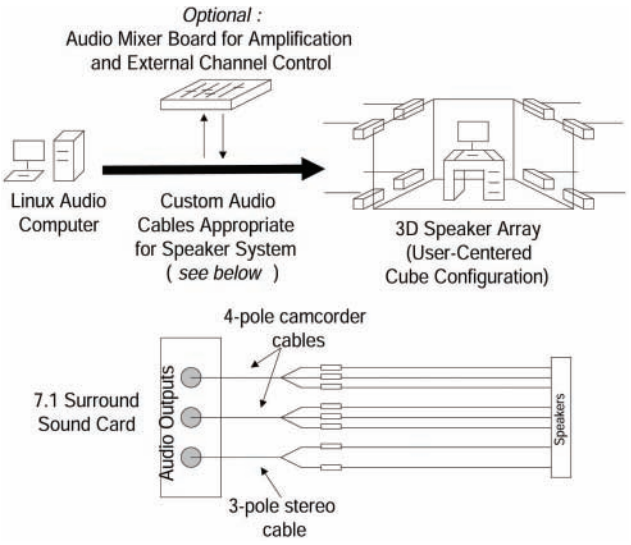


Figure 2. Audio Hardware Setup

Hardware Selection and Setup

The hardware needed to set up a low-cost 3-D spatial audio system includes a commodity sound card with certain features, speakers and audio cables. Here, we describe our choices for hardware components and the steps needed to set up the hardware. Throughout the discussion, refer to Figure 2 for an illustration of the hardware interconnections, speaker placement and wiring.

The first thing to consider is the number of speakers needed to produce 3-D spatial audio for a specific application. A minimally encompassing setup produces sound from all directions around the user—left-right, front-back and up-down. The speakers can be placed in any configuration, but setting up the 3-D audio panning functions is not as simple for irregular configurations. We decided to use eight speakers in a cubic configuration, with each speaker at a vertex of the cube, as shown in Figure 2. There is nothing special about the speakers needed for this task—the choice is a matter of budget and taste. We used eight amplified commercial-grade speakers for the simple reason that we already had them in our lab.

The eight speakers require a sound card that can produce eight channels of audio. Of the low-cost commodity audio cards, the only applicable candidates are the 7.1 cards. We chose the Creative Labs Audigy 2 card, which we found available at the time of this writing for as low as $90 US. Although it is possible to produce eight independent channels of audio using more expensive sound cards, the Audigy 2 card is the only commodity card we are aware of that has drivers in place to support what we are doing.

Ultra Dense, Powerful, Reliable...
Datacenter Management Simplified!
15" Deep, 2-Xeon/Opteron or P4 (w/RAID) options

Customized Solutions for...
Linux, BSD, W2K
High Performance Networking Solutions
• Data Center Management
• Application Clustering
• Network and Storage Engines
Rackmount Server Products
• 1U Starting at $499: C3-1GHz, LAN, 256MB, 20GB IDE
• 2U with 16 Blades, Fast Deployment & more...
iron SYSTEMS™
Iron Systems, Inc.
2330 Kruse Drive, San Jose, CA
www.ironsystems.com
CALL: 1-800-921-IRON

It is important to understand the output from the card. In a typical Dolby Surround Sound 7.1 speaker arrangement, there are two front speakers, left and right; two side speakers; two rear speakers; a center speaker, which sits in front and above the video screen; and a subwoofer channel, the .1 speaker. The Audigy 2 ZS has three analog output jacks, an 1/8-inch mini-phone, labeled 1, 2 and 3, that provides line-level outputs for the eight speakers. Jack 1 is three-pole, meaning it carries three signals—two signals drive the front left and right speakers and the third is ground. Jacks 2 and 3 are four-pole, each carrying four signals. Jack 2 drives the rear speaker pair and a side speaker, while Jack 3 drives the subwoofer, the center speaker and the remaining side speaker. One final consideration is these signals are unamplified line-level, so the speakers need to be the amplified type that accept line-level inputs. Alternatively, a separate amplifier or set of amplifiers should be used between the sound card and the speakers.

The next step is to install the speakers. Many speakers designed for surround use include mounts, but speaker mounts are available commercially for a number of other speaker types. In our application, we already had the cubic infrastructure in place and used custom mounts to attach the speakers to the cube. The simpler the speaker configuration, the simpler the software configuration can be—that process is explained later in this article.

Finally, the speakers must be connected to the audio card. How one connects speakers to the Audigy 2 depends on the type of speaker and amplification used. A trip to a favorite electronics store should yield any necessary connectors. In our case, the speakers each have a two-pole 1/4-inch phone jack, so we needed to split the three combined outputs of the sound card into eight separate signals. For Jack 1, we used a readily available 1/8-inch-stereo-to-dual-RCA adapter. For Jacks 2 and 3, we found similar adapters with four poles and three RCA connectors. These adapters are used most commonly with camcorders, when the three signals are used for composite video and stereo audio. These adapters gave us eight separate RCA connectors, and after obtaining eight long RCA-to-1/4-inch-mono cables, we were set.

In our final configuration, we used an Alesis Studio 32 mixer. This device fits in-line between the audio card's outputs and speakers' inputs and allows fine-tuning of the volume levels. Although the mixer made it a little easier to test and tune the audio, it wasn't truly necessary, as the same adjustments can be made in software.

Software Selection and Setup

The software solution for low-cost 3-D spatial audio is best described by the layered hierarchy shown in Figure 3. The software layers required to interface with the sound cards include low-level audio drivers and a 3-D spatial audio API. We focused our primary development efforts on Linux because of easy access to the source code for low-level audio drivers and the overall support community that exists for developers working on projects such as ours.

For the driver layer, we chose ALSA, which was mentioned previously. ALSA provides audio and musical instrument digital interface (MIDI) functionality to the Linux operating system. It supports many types of audio hardware, ranging from consumer sound cards to professional multi-channel audio interfaces.

We selected ALSA because it appeared to require the least effort to generate the eight channels we needed for 3-D spatial audio. Until we modified the ALSA driver to access all eight channels, it supported only six channels (5.1) on the Audigy 2. These changes have been incorporated into ALSA, but they may or may not be in a release version at publication time. In that case, one can get the latest source and build it—be sure to include the emu10k1 sound card argument when using the

Figure 3. Audio Driver Layers

./configure script so that the ALSA driver recognizes the Audigy card.

After the driver is set up, the 3-D spatial sound API can be installed. It distributes sound effects from a given 3-D position to the appropriate audio channels. Although there are quite a few APIs to choose from, we chose Mustajuuri, as mentioned previously. The Mustajuuri software works with ALSA and provides 3-D panning over an arbitrary array of speakers using the VBAP algorithm, also described previously. The Mustajuuri API provides all of the features needed for a basic 3-D positional sound system and is fairly easy to extend. Over the course of this project, we made several minor source code modifications, and they are included in the October 2004 release.

Mustajuuri does its magic by way of a module called the Mixer, which mixes multiple sound sources—sound files, microphone inputs or other sources—into individual audio streams. These streams then are piped into a panning module, which is responsible for routing each input signal to the appropriate speakers, setting the correct gain and time delay at each speaker and mixing multiple streams meant for the same speaker into a single stream to be sent to that speaker. It does the routing and gain calculations based on VBAP, and some additional gain and delay calculations are based on distance. The result is each incoming sound source to the panning module leaves from a set of three speakers, and the resulting sound appears to come from a specific 3-D position in space. Doppler shifting also is simulated.

Once Mustajuuri is compiled and installed, several tasks must be performed to configure the software to work with the given 3-D speaker array.

Configuring ALSA

ALSA needs to know how to communicate with all eight channels of the audio card. This normally would be achieved sim-

ply by using the device named surround71, but it is not fully compatible with the spatial sound API Mustajuuri. Mustajuuri requires support for input channels. The device surround71 supports eight output channels but no input channels. Therefore, it is necessary to define a new device that has eight output channels and some input channels.

In order to meet this requirement, an asymmetric device is defined. The device is called asymmetric because the number of input and output channels are not necessarily the same. Notice that the number of input channels is not stated explicitly. ALSA determines the number of input channels automatically and assigns the maximum; the Audigy card we used has two.

To configure ALSA, add the following text to the file /etc/asound.conf or create the file if necessary. This file holds information about user-defined devices, so we use the following text to add an asymmetric device called eightout:

```
ctl.eightout {
    type hw
    card 0
}

pcm.eightout {
    type asym
    playback.pcm {
        type route
        slave.pcm surround71
        ttable.0.0 1
        ttable.1.1 1
        ttable.2.2 1
        ttable.3.3 1
        ttable.4.4 1
        ttable.5.5 1
        ttable.6.6 1
        ttable.7.7 1
    }
    capture.pcm {
        type hw
        card 0
    }
}
```

Next, an environment variable must be set to allow Mustajuuri to talk to the audio card through ALSA. Set the following environment variable:

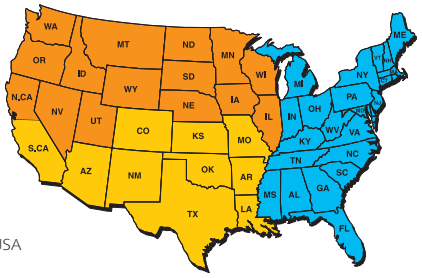
```
export MJ_AUDIO_CONF= \
"Input=2=hw:0,0 | Output=8=eightout"
```

Once this is done, Mustajuuri should be able to output audio through all eight channels of the audio card.

Configuring the Mustajuuri Mixer Panel

Mustajuuri uses a mixer-board-style GUI for sending input audio streams to a speaker array, combining them or just passing them through intact. The input streams can come either from sound files or from live sources, such as a microphone. The GUI lays out several strips of channels that can be assigned different functions applied in a sequential process.

LINUX JOURNAL



PO Box 55549
Seattle, WA 98155-0549 USA
www.linuxjournal.com

ADVERTISING SERVICES

VP OF SALES AND MARKETING
Carlie Fairchild, carlie@ssc.com
+1 206-782-7733 x110,
+1 206-782-7191 FAX

FOR GENERAL AD INQUIRIES
e-mail ads@ssc.com
or see www.linuxjournal.com/advertising

Please direct international advertising inquiries to VP of Sales and Marketing, Carlie Fairchild.

REGIONAL ADVERTISING SALES

NORTHERN USA
Joseph Krack, joseph@ssc.com
866-423-7722 (toll-free),
866-423-7722 FAX

SOUTHERN USA
Laura Whiteman, laura@ssc.com
206-782-7733 x 119

EASTERN USA
Martin Seto, mseto@ssc.com
+1 905-947-8846,
+1 905-947-8849 FAX

INTERNATIONAL
Annie Tiemann, annie@ssc.com
866-965-6646 (toll-free)

Advertiser	Page #
Aberdeen, LLC www.aberdeeninc.com	9
APPRO HPC Solutions appro.com	19
ASA Computers www.asacomputers.com	75, 77
Avocent www.avocent.com	53
Cari.net www.complexdrive.com	67
Ciara Technology www.ciara-tech.com	16, 17
Coraid, Inc. www.coraid.com	21
Coyote Point www.coyotepoint.com	61
Cyclades Corporation www.cyclades.com	C2, 1, 11
Emac Inc www.emacinc.com	44
EmperorLinux www.emperorlinux.com	15
Etnus www.etnus.com	85
FairCom Corporation www.faircom.com	39
Free Software Foundation www.gnupress.org	93
Google www.google.com/lj	35
Hurricane Electric www.he.net	55
Imagestream Internet Solutions www.imagestream.com	37
Integrated IT Solutions DBA SAG Electronics www.sagelectronics.com	87
Intel intel.com/go/xeon	29
Intel Premier Provider www.insistthebest.com	28
Iron Systems www.ironsystems.com	79
Iwill Usa Corp www.iwillusa.com	2
JTL Networks www.jtl.net/lj	92
Layer 42 Networks www.layer42.net	43
Levanta www.levanta.com	13
Linux Journal www.linuxjournal.com	91
Linux World Expo UK www.turretgroup.cm	33
LPI www.lpi.org	65
MBX www.mbx.com	7
Microway, Inc. www.microway.com	C4, 83
Novell www.novell.com	47, 49, 51
Open Source Professional Institute www.ospinstitute.com	69
Penguin Computing www.penguincomputing.com	23
PFU Systems www.pfusystems.com	57
R Cubed Technologies www.rcubedtech.com	25
Rackspace Managed Hosting www.rackspace.com	C3
RoboNexus www.robonexus.com	63
SBE, Inc. www.sbei.com	5
SC|05 sc05.supercomputing.org	89
Servers Direct www.serversdirect.com	41
Storix Software www.storix.com	59
SugarCRM, Inc. www.sugarcrm.com	73
Technologic Systems www.embeddedx86.com	45
TUX Magazine www.tuxmagazine.com	90
The Portland Group www.pgroup.com	30, 31
ZT Group International www.ztgroup.com	27

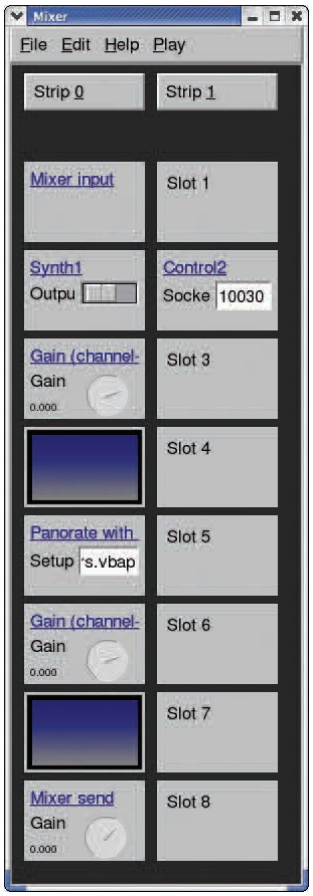


Figure 4. Screenshot of the Mustajuuri Mixer GUI (Tommi Ilmonen, Mustajuuri-2004)

Some example functions are input, send (to speaker), amplitude gain, panning and synthesizer. The gain and panning modify how the audio is distributed to individual output audio channels.

The Mixer Panel configuration we use is shown in Figure 4, which uses two mixer strips. The first has two interesting channels: a synthesizer channel, which manages the sound files, and a panning module, which handles the VBAP-based panning across speakers. The second strip is used to manage remote connections from external applications and does not accept an audio stream as input. It sends commands to the synthesizer and the VBAP module.

To create a similar configuration, launch Mustajuuri and create a new mixer from the File menu. This mixer has several strips already, and all of these strips essentially are blank. The number of strips and the number of modules per strip can be changed using the Edit menu, if needed. Modules can be assigned by clicking with the mouse on a particular slot. To adjust the module's properties, simply click on the blue link defining the module's type, such as Synth1 or Mixer Input. The Strip X button at the top of a strip can be used to modify and remove the modules in any slot in that strip. All mixer configuration changes are saved by using the save options from the File menu. The resulting configuration file, for example, SpatialAudio.mj, is specified on the command line when Mustajuuri is called.

Specifying Speaker Placement

In order to use VBAP, it is necessary for Mustajuuri to know the locations of the speakers in the 3-D array. Mustajuuri does this through a configuration file that is specified as part of the VBAP module setup; this module was created as part of configuring the Mustajuuri Mixer Panel. This file specifies the azimuth and elevation angles (in degrees) for each speaker relative to the listener. Because our system uses eight speakers arranged in a cube configuration, our configuration file is specified as follows:

```
3         # dimensionality
# Azimuth, followed by elevation.
# 0 0 would be straight ahead.
-45 45    # Front upper left
45 45     # Front upper right
-135 45   # Back  upper left
135 45    # Back  upper right
-45 -45   # Front lower left
45 -45    # Front lower right
-135 -45  # Back  lower left
135 -45   # Back  lower right
```

This configuration file is used by the main configuration file for the Mustajuuri Mixer. This file assumes that all speakers are equidistant from the listener. If this is not the case, adjust the gain and delay for each speaker manually by using the Mustajuuri Mixer. In our system, such adjustments were not necessary, and it would involve significant work if it were necessary. The easiest solution is to try to place all speakers equidistant from the ideal listening position.

Configuring the Sound File Loader

In order to use a sound file from Mustajuuri, that sound file must be known to Mustajuuri at load time. The mechanism used to do that is a configuration file that specifies all of the sound files that possibly might be used by Mustajuuri. This configuration file is used by the synthesizer mixer module, which was created when we configured the Mustajuuri Mixer Panel above. A sample configuration file that loads three sound files follows. Once a file is created, any name can be assigned; make sure the synthesizer module points to that file:

```
unusevoices *-stk
polyphony 48

sample audioeffect1.wav
sample audioeffect2.wav
sample sudioeffect3.wav
```

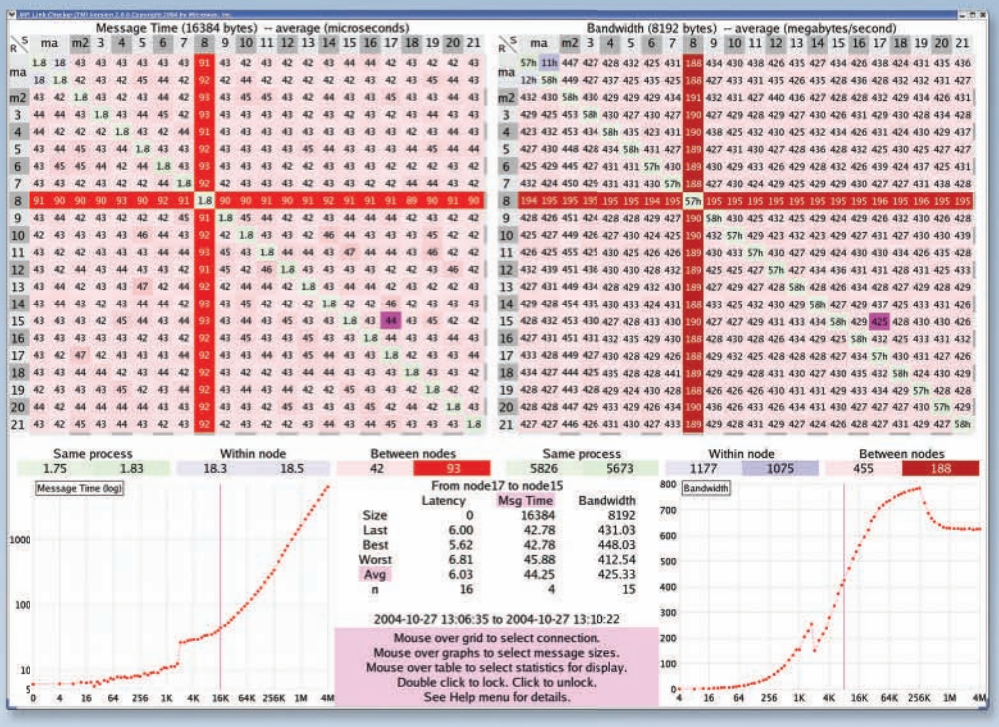
The polyphony line specifies the maximum number of audio files that should be loaded by Mustajuuri, so it should be at least as large as the number of audio files listed in this file. The last three lines specify three sample audio WAV files. Any audio files specified here must be placed in a directory specified separately as part of the synthesizer module configuration. The unusevoices line is a somewhat more-advanced setting, but one that should help improve efficiency somewhat.

Configuring Mustajuuri for Remote Control

Mustajuuri is designed to act as a standalone program to manipulate audio, not as a library to be linked against by another application. To control Mustajuuri from another application, as is the case with our project, two steps are required. The first involves setting up Mustajuuri to listen for control commands over the network. The second step consists of writing a simple API in the main application to talk to Mustajuuri.

To get Mustajuuri to accept commands from the network, a network module must be loaded. This network module is the only way for an external application to control Mustajuuri, even if both the application and Mustajuuri are running on the same machine. Adding this module is a simple task and simply requires configuring which port Mustajuuri will listen to; the default port is 10030. This module automatically communicates with the synthesizer module and the VBAP modules, if

X Marks the Slow Node!



Microway® Quad Opteron™ Cluster with 36 Opteron 852s, redundant power and 45 hard drives in CoolRak™ cabinet.

MPI Link-Checker™ to the Rescue!

A single slow node or intermittent link can cut the speed of MPI applications by half. Whether you use GigE, Myrinet, Quadrix, InfiniBand or InfiniPath HTX, there is only one choice for monitoring and debugging your cluster of SMP nodes:

Microway's MPI Link-Checker™

Our unique diagnostic tool uses an end-to-end stress test to find problems with cables, processors, BIOS's, PCI buses, NIC's, switches, and even MPI itself! The newest release provides ancillary data on inter-process and intra-CPU latency which can vary by a factor of 10 between MPI versions. MPI Link-Checker is also useful for porting applications to new hardware. It provides instant details on how latency and bandwidth vary with packet size. It is available now for a free 30 day evaluation!

Wondering what's wrong with your cluster, or need help designing your next one? Call our HPC staff at 508-746-7341. Visit microway.com to learn about new low latency interconnects including the PathScale InfiniPath HTX Adapter, which delivers unmatched MPI latency of under 1.5 microseconds.

Microway has been an innovator in HPC since 1982. We have thousands of happy customers. Isn't it time you became one?

Call us first at 508-746-7341 for quotes and benchmarking services. Find technical information, testimonials, and newsletter at www.microway.com.

Excellence in Cluster Technology
64-bit Turnkey Solution
WINNER
ClusterWorld

Microway®
23 Years of Expertise Built In

PathScale™
Accelerating Cluster Performance™

they are in the mixer.

In order to control Mustajuuri from an application, it is necessary to add code included with the Mustajuuri API into the application. We present example code segments here that show how to connect to a remote audio server, play audio specified coming from a given 3-D position and change the position of a sound source and listener position.

The first code segment shows the initial commands to connect the application to the remote Mustajuuri audio server and initialize it. If Mustajuuri is running on another machine, change the address to reflect this. To change the default port, 10030, that Mustajuuri listens to, specify the new port in the address string, for example, mjserver.mydomain.com:12345. The two objects that we created, one instance each of AC_Control and AC_VrControl, are used later to send commands to Mustajuuri:

```
// connecting to a remote server
  #include <ac_vr_control.h>

  AC_Control acControl =
    new AC_Control();
  char* mjServerAddress = "127.0.0.1";
  if(!acControl->init(mjServerAddress))
  {
    // error handling code here
  }

  AC_VrControl acVrControl =
    new AC_VrControl( acControl );
```

The next code segment shows how to specify the position of the source of the audio and play it. One interesting thing to note here is that the variable outputChannel identifies the intended sound source to use. The number of supported sound sources was specified in the synthesizer module from the Configuring the Mustajuuri Mixer Panel section, and outputChannel should be between 0 and the number of sources, minus one. The variable soundFilename should not have a path as part of the filename. The filename should be one of the files listed in the configuration file created as part of configuring the Sound File Loader file. Lastly, the soundLevel is essentially the initial gain level for the new sound. This needs to be experimented with to find an appropriate setting:

```
AC_Vector3 location(
    positionX,
    positionY,
    positionZ);

  int soundId =
    acVrControl->playSample(
      outputChannel,
      soundFilename,
      soundLevel,
      location,
      true,
      0, 0);
```

The last code segment shows how to reposition the sound source location and the orientation and position of the listener. The outputChannel variable refers to the sound source that is desired to be moved and should be the same value used to call playSample from the previous example. The listenerRotation matrix specifies the orientation of the listener relative to the world, and the worldRotation matrix specifies the orientation of the world relative to the speakers:

```
// reposition a source of sound
  AC_Vector3 location(
    positionX,
    positionY,
    positionZ);

  acVrControl->moveSource(outputChannel,
    0.05, location);

// reposition the listener orientation
  AC_Matrix3 listenerRotation(
    ... listener rotation matrix ... );
  AC_Matrix3 worldRotation(
    ... world rotation matrix ... );

  acVrControl->setTransformations(
    location, listenerRotation,
    worldRotation, 0.05);
```

Hardware Testing and Calibration

We tested the hardware design of our 3-D spatial audio system by integrating the hardware with our four-wall immersive virtual reality room at the Virtual Reality Laboratory, part of the Naval Research Laboratory in Washington, DC. We arranged the speakers in a cube array and placed them at the corners of the immersive room, as shown in Figure 5. We designated a 1.2GHz Red Hat Linux machine as an audio server and installed an Audigy 2 ZS card. We connected the speakers using the cabling described before and tried the system both with and without the mixing board mentioned earlier.

While Mustajuuri was running with three audio sources in motion, CPU utilization on this machine generally was less than 20%, and the memory usage was negligible. Further savings could be realized, of course, by using optimized compiler settings rather than the debug settings we used.

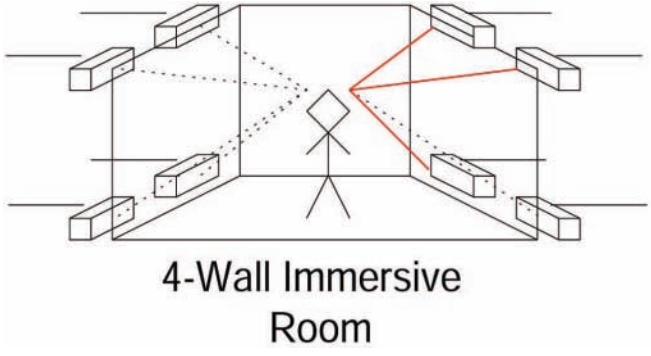


Figure 5. Immersive room depiction showing placement of speakers in a cube array and audio coming from the user's front upper-right direction; the lines are colored red.

We tested the outputs from each speaker to determine the range of intensities that could be played on each channel. We listened to each speaker individually for sound quality, sound balance and percussive resonance. The easiest aspect to listen for is the sound balance between treble and bass. If one of the two obviously is higher than the other, adjust the related frequency filters as needed. For example, if there is too much bass, decrease the bass and/or increase the treble. If there seems to be excessive low-end or high-end noise, adding a low-pass or high-pass filter may be necessary.

Another easy aspect to listen for is speaker distortion. Simply put, if the speakers are so loud that the sound produced is bad, lower the volume of the speaker. If a given set of speakers cannot produce quality sound at an acceptable volume, it may be necessary to acquire more powerful speakers.

One of the hardest aspects to listen for is the resonance of percussive sounds generated by a speaker. This quality basically is how much the sound echoes from where the speaker is located. Adjustments have to be made if it sounds like a speaker is reverberating with percussive sounds. Depending on the quality of the speaker and the quality of the mixer board, this problem may be corrected to some degree by continuing to filter the signal. For excessively bad cases, hard objects such as exposed metal, concrete, hard plastic and even glass should be covered with a sound dampening material such as cloth or foam.

Once each speaker is calibrated, the entire setup has to be balanced. This can be done either by using devices designed to measure acoustic levels or by listening to the speaker from the predetermined center of the 3-D speaker array. Either way, the gain of each speaker should be adjusted until the same audio intensity level is received from each channel. Keep in mind that the outputs for each channel of the audio card were customized by the manufacturer for the intensity requirements for each type of speaker—satellite, center and subwoofer—normally attached in a surround configuration, and the intensity output for each type differs. There are many published methods for dealing with this problem, but we went with the low-tech solution of having someone stand and listen at the center of our speaker array. We set the software control to maximum gain and adjusted the mixer board based on feedback from the listener. Remember that these changes can be made in software with the ALSA drivers and Mustajuuri if a mixer is not available.

Software Testing

We tested the software by integrating sound into an existing in-house simulation platform, BARS-Utopia, that operates on a Linux visualization cluster from ORAD Incorporated, which drives our immersive room. BARS-Utopia supports several virtual world databases, interaction methods and spatial audio. However, no support was available for interacting with the Mustajuuri API in particular, so we implemented a plugin to bridge the BARS-Utopia spatial audio support with Mustajuuri. BARS-Utopia already contains all of the information needed

A NASTY LEAK
LEFT CODY UP THE CREEK

Characters and Images ©2004 Brad Fitzpatrick, ActiveEdge. All Rights Reserved.

Memory leaks can spring up at the worst moments. Finding them is fast and easy with TotalView's built-in, interactive memory debugger. Now you can uncover leaks anytime, even while you debug your code, and without batch-style reports. So, try TotalView today for free, and you won't find yourself up that famous creek without a paddle.

Now with built-in, interactive memory leak detection!

Try TotalView FREE at www.etnus.com

The most advanced debugger on Linux and UNIX

ETNUS
TOTALVIEW

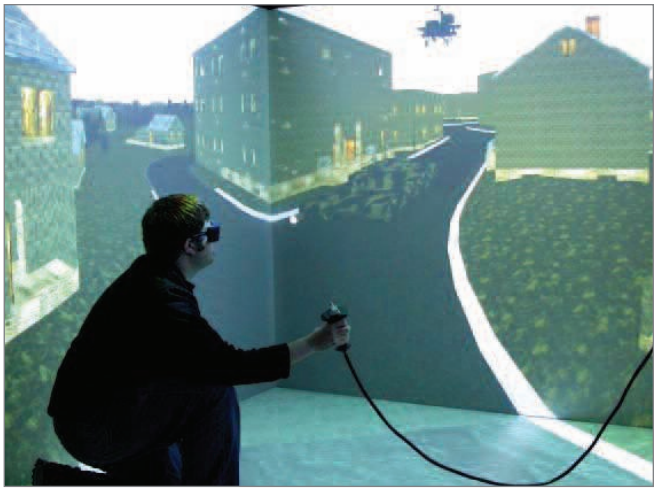


Figure 6. Shows a user interacting with a scene with multiple sound sources—a tank, helicopter and a jet that is not visible. The nearest speakers are determined for each sound source and the output is mixed at each speaker, if overlapping.

by Mustajuuri, such as sound source positions, listener position and orientation and sound source creating/deletion notifications—the plugin simply translates that data into a form that Mustajuuri understands.

When the plugin was completed, we tested and debugged the new system. The primary software adjustments we made were to the attenuation level of the audio channel outputs. Mustajuuri uses a simple attenuation model and requires some manual tweaking for the expected environment, things such as outdoor, indoor, time of year and so on. In the real world, sound attenuation rates are quite complicated and are influenced by factors such as temperature, humidity and the frequency makeup of the sound.

We tested the sound system by implementing several scenarios, each with a different scene dataset and different audio effects attached to an animated object. Before the audio objects were animated, we evaluated several volume levels and several distances away for each object. Figure 1 shows a simple scenario we designed and tested—the sound effects of a car. When we finished testing the volume and distance effects, we generated an animated path for the car to follow.

Figure 6 shows a more complex scenario with three audio sources, tank, jet and helicopter—the jet is off the screen and not shown in view. We performed some simple tests to see how many sound sources interacted together. It was of primary importance that the jet, typically far away, not sound too quiet, while the tank and helicopter, typically closer to the camera, not dominate the aural bandwidth. As a result, some minor tweaking was done on both the far and near objects' attenuation parameters.

System Validation

After all of the testing and calibration was completed, we performed two informal, qualitative user tests that would help us validate our new low-cost spatial audio system. The first test evaluated how the new sound system configuration with eight speakers compared with our previous planar configuration containing four speakers. The prior configuration simply used the four speakers on the top of the cube array. We realize that directly comparing these two configurations is somewhat biased, due to the placement of the four-speaker array being located above the user's head. It would be more fair to compare against a four-speaker array located at the height of the user. However, by using the top four speakers, we were able to switch between the two configurations without dismantling our installation.

We performed the experiment by asking a few test subjects to stand in the middle of the immersive room and listen to sounds played for each configuration. We played different sequences of audio on both speaker configurations and made use of the full range of speakers available. The subjects were not told which configurations were being used, nor in which order the pairs of configurations were presented. Several iterations of the pairs of configurations were tried for each subject. After each pair was presented, the subjects rated the two systems. Admittedly, this was not a scientific test, as is evidenced by several unaddressed biases, but all test subjects clearly preferred the eight-speaker configuration.

The second user test evaluated how well the listener is able to localize the source of the audio using the eight-speaker configuration. Again, the subjects were tested and each was asked to stand in the center of the immersive room. Each subject was presented with several sounds played one at a time and originating from different positions surrounding the subject. The subjects were asked to point in the direction of the sound source, as they heard it. The visual system was not running, so the users did not get visual cues as to the sound source's location. The subjects were able to localize the sounds with a high degree of accuracy, especially with respect to elevation.

The implementation of our 3-D spatial audio system integrated with our immersive room really enhanced the simulation and training demos we have. Our completed system has improved dramatically the sense of immersion when running the demos. A simulation user easily perceives helicopters and jets flying overhead and a tank rumbling down one of the many streets nearby in the virtual world. The perception of depth from the source of audio is conveyed accurately and also includes doppler effects. Our system is a step above a four-speaker solution when we had previously used the Microsoft DirectSound API. It also is a good replacement for the capable but outdated and unsupported eight-speaker solution we had running using another expensive hardware and software platform.

Conclusions and Future Work

We have devised a true 3-D spatial audio solution that is low cost and has comparable quality to expensive high-end commercial systems. The 3-D spatial audio solution allows sound effects to be generated from all directions surrounding a user, not only from planar directions. We accomplished this feat by using only commodity hardware and open-source software. We feel this feature, now available at an affordable price, creates numerous options for game and virtual reality system developers.

We feel our system leads the way for others to devise similar solutions with current and future commodity audio equipment. The developer needs only to purchase a Dolby Surround

Sound 7.1 audio card, four pairs of low-cost speakers and audio cables. We spent less than $150 US on hardware—Audigy 2 audio card and audio cables—as we already had speakers available. From start to finish, including hardware and software debugging, configuring and testing, we spent less than a month developing the low-cost 3-D spatial audio system. We feel that using this document as a guide, it should be possible for others to implement this system in less than a week.

Acknowledgements

We wish to thank Tommi Ilmonen from the Helsinki University of Technology (HUT) for support on modifications made to Mustajuuri. We also wish to thank Bryan Hurley, Simon Julier, Mark Livingston, Yohan Baillot and Jonathan Sabo for contributions to the research. This research was sponsored by the Office of Naval Research under contract #N00014-04-WX-20102.

Resources for this article: www.linuxjournal.com/article/8407.

Eric Klein is a graduate student in the Institute for Data Analysis and Visualization Virtual Reality Laboratory at UC Davis. He is working on a PhD in Computer Science, specializing in virtual reality. Eric received his BS from UC Santa Barbara and spent several years working as an engineer in the industry before returning to graduate school. His primary research interests are immersive audio, data sonification, scientific visualization, collaborative environments and human-computer interaction.

Greg S. Schmidt is a computer scientist in the 3D Virtual and Mixed Environments Laboratory at the Naval Research Laboratory. He has a PhD and MCS in Computer Science from Texas A&M University and a BS in Biomedical Engineering from Marquette University. His research interests include scientific and information visualization, human-computer interaction, augmented reality, modeling and simulation for terrain and medical applications and computer vision.

Erik B. Tomlin is a student at the University of Pittsburgh majoring in computer engineering. He has been working for the 3D Virtual and Mixed Environments Laboratory at the US Naval Research Laboratory on research projects involving virtual and augmented reality, human-computer interaction and scientific visualization.

Dennis G. Brown is a computer scientist at the Naval Research Laboratory. He received his BA in Computer Science from Rice University and his MS in Computer Science from the University of North Carolina at Chapel Hill. He works on the Battlefield Augmented Reality System (BARS) and multi-modal virtual reality projects. His research interests include augmented and virtual reality, specifically, novel user interfaces and data distribution.

YOUR HIGH PERFORMANCE COMPUTING SOLUTION HAS ARRIVED

intel inside XEON

SAG STF Blade server

- up to 14 Xeon™ Processor 800MHz front side bus
- up to 56G ecc reg ddr2 400
- up to 24 36gb or 73gb 2.5" SCSI disk drives
- 1x gigabit ethernet switch chassis
- 1x management module
- 2x blowers
- 1x cd-rom, 1x floppy
- 1x rack mount kit
- 2x 2000 watt power supplies

Please call for detailed configuration requirements and pricing.

The core of any custom built HPC solution built by SAG Electronics is the Intel® Xeon™ Processor based blade server. We have servers , workstations and storage to create a custom solution that meets your demanding HPC specs with a service package to meet your needs. Call today for pricing, based on your configuration requirements.

3 YEAR NO WORRY WARRANTY

Call Now! 1-800-488-4724

SAG ELECTRONICS www.sagelec.com

GSA Schedule GSA# 35F-0313M

Intel® Xeon™ is a trademark of Intel Corporation

Taming the TODO

Buried under a mass of sticky notes? If you worry about forgetting important tasks or you want to schedule things efficiently, here are some ways to get organized. BY SACHA CHUA

In this article, I offer some ways to manage your tasks. From simple text files to full-blown personal information managers (PIMs), there's bound to be one method that fits your way of working. I also share some tips on managing your tasks and tell you about how I fit a task manager to my way of working.

You Want the Works

If you're accustomed to the advanced task management features of Microsoft Outlook and other proprietary PIMs, then Ximian Evolution and the KDE PIM suite are great fits for you. Ximian Evolution was developed for the GNOME user environment, and the KDE PIM suite is part of KDE, but each is usable with other desktop environments.

Offering a polished interface for creating and managing tasks, attaching files and even synchronizing with personal digital assistants (PDAs), these full-fledged personal information managers can help you tame your to-do lists (TODOs) in style.

You Want to Keep Things Simple

Sometimes the simplest method is the best. Keep tasks in a plain-text file, and you're already well on your way to taming your TODOs. Text files win in terms of flexibility. You can keep your lists in any format you want and edit them using your favorite editor. You also can share them with others through e-mail or the World Wide Web. You even can keep them backed up and synchronized with other computers using tools such as rsync and CVS.

Memorize keyboard shortcuts for copy and paste. Incremental search is a great way to jump to tasks if you remember a small part of the description. Your text editor then can display matches as you type in characters. Check out your text editor's features for more help.

Beyond the basics, a little bit of programming makes TODOs easier to keep. Write a small program or shell script to add items from the command line or a keyboard shortcut. The less effort it takes to write down a task, the more you'll remember, so automate as much as you can. You can sort tasks manually by copying and pasting lines in your TODO list or even writing programs to put everything together.

For more software support, check out Freshmeat.net for hundreds of simple TODO managers. If you know how to program, pick a TODO manager in a language you know or would like to learn. Extending a manager's capabilities not only helps you grow as a programmer but also lets you tailor it to your particular quirks.

You Get Most of Your Tasks through E-mail

E-mail is a popular way to keep track of tasks. If you practically live in your e-mail client, why not use it to keep track of the things you need to do? You can forward messages or write yourself reminders. Use meaningful subjects to make it easier to get a bird's-eye view of your messages.

Watch out for information overload, however. You may need to find that urgent TODO in an archive of thousands of messages. Check out your mail client's features for options on how to tag messages. Use folders or labels to flag messages for follow-up action. Tag or file messages as TODO, and remove the label or change it to "done" after you finish the task. Keeping track of tasks is easier with full-fledged PIMs, such as Evolution and KDE PIM, which allow you to mark a message for follow-up or convert it to a task.

What about small tasks? It might seem silly to e-mail yourself a reminder to buy milk, but unless all of these TODOs are written down somewhere, you're going to spend mental energy thinking about them. You therefore may need to supplement your inbox with a way to keep track of smaller tasks.

If most of your tasks can be accomplished quickly and you can keep your inbox manageable, e-mail is a convenient way to keep track of your tasks.

You Work with a Lot of People on Tasks

Many software projects use request trackers to make sure that bug reports and feature requests don't slip through the cracks. You can use one to keep track of your personal TODOs too. Although a request tracker requires a lot of setup time and effort, you reap the benefits of a solid project management system.

Request trackers, also known as bug-tracking systems (BTSes) or issue trackers, archive all of the messages related to a TODO, making them great for tasks occurring over long periods of time and tasks when you need to collaborate with other people. You can send the e-mail address or URL for a task to other people so they can confirm your work or add comments.

Request trackers can produce task-related graphs. For example, you can track the increase or decrease in open, resolved and closed tasks over time to get a rough estimate of when you're most productive or overloaded.

If most of your tasks require input from others, check out programs such as RequestTracker and Bugzilla. With a good bug-tracking system in place, you easily can keep track of what you're waiting for and from whom.

You Practically Live in Your Web Browser

Web-based TODO lists are a fun and easy way to create task lists you can share with other people. If you always have a Web browser open or you need to keep non-techies updated, a Web-based TODO list might be a handy way to keep track of your tasks. New services such as Ta-da and Backpackit use

SC|05 Gateway to Discovery

Washington State Convention and Trade Center, Seattle WA
November 12-18

SC|05 Highlights:

- **Keynote Address by Microsoft Chairman and Chief Software Architect Bill Gates**

SC|05 initiatives HPC Analytics and StorCloud will push the boundaries of high performance storage, analytics and networking applications

Technical, Education and SC Global Programs offer attendees a gateway into innovative ideas and expertise in HPC, networking and storage

A dynamic exhibition featuring more than 200 of the world's leading HPC industry and research organizations

SUPERCOMPUTING

SC|05, the premier international conference on high performance computing, networking and storage, will convene in November 2005 in Seattle. Under the theme "Gateway to Discovery" SC|05 will showcase how high performance computing, networking, storage and analysis lead to advances in research, education and commerce.

Registration Date: August 2005
Register: http://sc05.supercomputing.org/
See you in Seattle!

Sponsors:
ACM SIGARCH
IEEE Computer Society

JavaScript and XHTML tricks to provide a great user interface.

Look for a bookmarklet or extension that lets you easily create TODOs. Make your task overview the default page in your browser so that you're sure to review them daily.

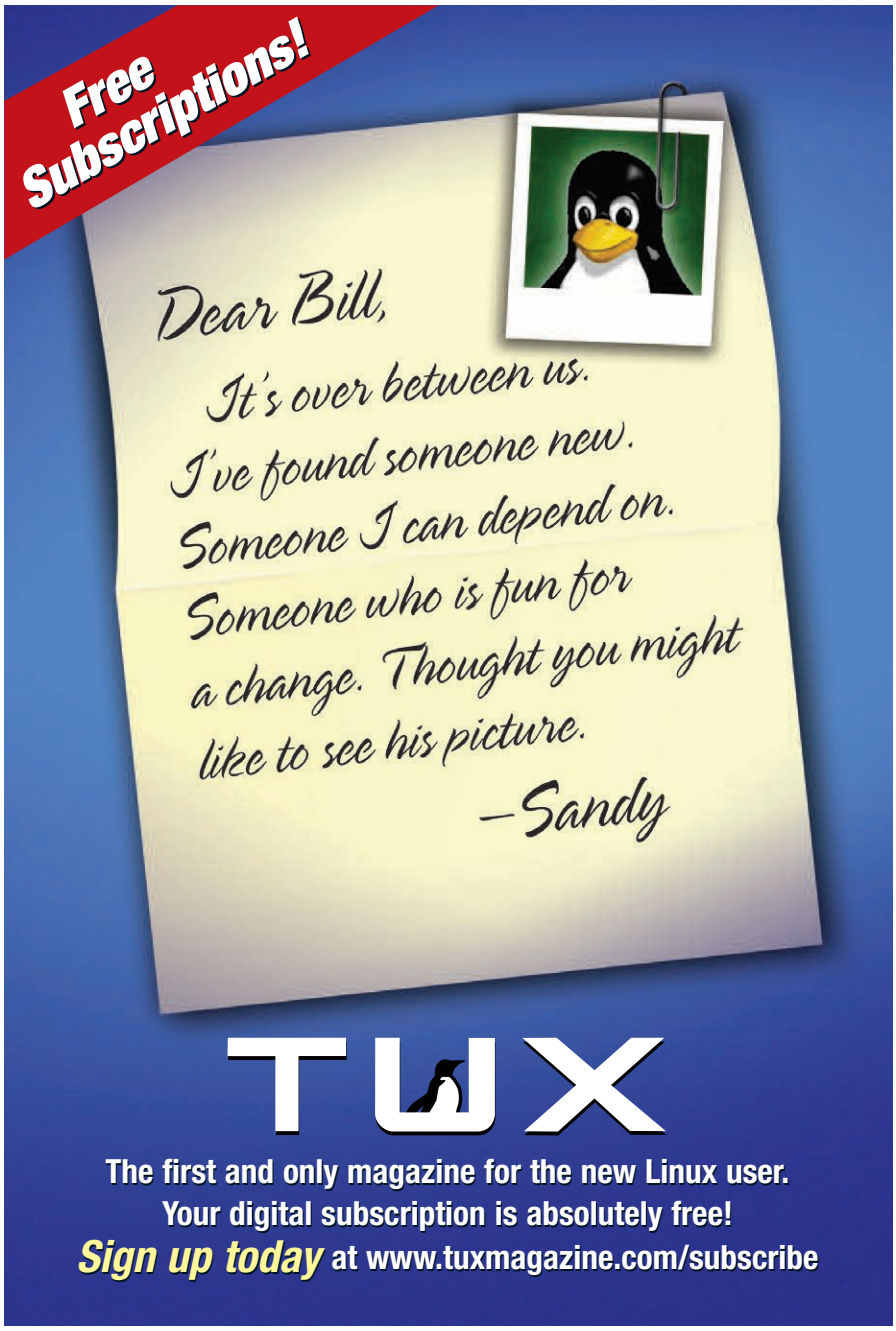
You're Always on the Move

If you spend a lot of time on the move, you've probably thought about getting yourself a PDA. Both Evolution and KDE PIM can synchronize your tasks with Palm-based PDAs, making them ideal for the mobile warrior. Libraries such as coldsync can help you support synchronization for your custom hacks.

My productivity tool of choice is a pack of 3"×5" index cards held together with a fold-back clip or rubber band. Affectionately called the "Hipster PDA" by productivity geeks, this surprisingly effective low-tech tool is a great way to keep track of tasks.

Write down your tasks, one per index card. You can write down subtasks and notes as well. Shuffle through your tasks while waiting or sort them by the context you can perform the tasks in. Rip the card up after completing your TODO for an extremely satisfying end.

Print useful data onto cards. Around 50 names and contact numbers can fit on an index card if you use a really small font. Month and year calendars also are handy. No hardware worries, no productivity-sapping games and no hassles make the Hipster PDA great for people on the go.



Tips for Taming your TODOs

Got an idea about what to use to manage your tasks? Well, now here are some tips for keeping on top of everything.

Make It as Easy as Possible

"Hmm, that looks interesting", you think. "Let's try it out." You switch to your task manager to write down that TODO. Oops, you still need to open the application. Now you have to arrange your windows so you can see the article. Wait, you need to copy the URL. By the time you have it all set up, you might've forgotten what you wanted to write down in the first place!

If a task manager is too cumbersome to use, you won't bother with it. Make it as easy as possible to get a task out of your head and into the system. Make your task manager a keystroke or click away, and you'll find yourself using it more often.

Don't Get Overwhelmed

Keep your TODO list short so that you don't get overwhelmed by all the things you need to do. Ruthlessly prune TODO items you no longer have to do or are no longer interested in doing. Delete or archive completed tasks so that they don't clutter your main task list.

TODO items can be intimidating. "Write a novel" is an example of a task that can be difficult to start. Make sure your TODO items are small enough to work on. I usually break my tasks down into subtasks I can do in one sitting. Breaking these tasks down also makes it easier to stop procrastinating, because there's always something small to work on.

Fill in the Cracks

Make a system you can trust. Ensure that none of the tasks fall through the cracks. Make your reminder system the first thing that shows up after you log on or start your browser. Set aside time to review all of your tasks regularly.

If your task manager is easy to use, you'll trust it with more tasks. Writing down all of your tasks in a reliable system means you don't have to worry about forgetting anything—as long as you don't forget to check!

Hack Your System

The way you keep track of tasks probably will change as you come up with new ideas or read about other people's experiences. Don't be afraid to improve your system. Instead of making a giant step to a brand-new methodology, however, break changes down into incremental improvements. That way, you give yourself time to make it a habit.

Don't spend too much time tweaking your system, though! One way to manage this impulse is to find a community of like-minded people. That way, you can use their hacks and customizations without having to spend a lot of time coming up with your own. The trick is to find a personal information manager that fits the way you work and can be extended as you experiment with new ways of working.

A Truly Personalized Personal Information Manager

I went through the whole spectrum of personal information managers before I found something that works for me. I'm absolutely crazy about Planner.el, a personal information manager that's extremely customizable. I'd like to share some of the things I love about it with you so that you can see how personal work style affects how you plan.

I spend most of my time working with text files in the Emacs text editing environment. Because Emacs is so extensible, it has accumulated a lot of useful modules along the way, including several e-mail clients, Web browsers, Internet relay chat (IRC) clients and even instant messengers. I can program, surf, chat and check mail within Emacs. Emacs itself runs on GNU/Linux, Microsoft Windows and Mac OS X and is surprisingly easy to learn.

Planner.el is built into my main working environment, making it only a keystroke away. Because most of my tasks are based on what I'm looking at, I really appreciate how Planner.el stays out of my way. When I create a task, a small text prompt shows up at the bottom of my screen (Figure 1). I don't get distracted by pop-ups or switching to another application. I simply type the task description in, tag it with a project or two and get back to work.

Not only that, but it also intelligently picks up information from whatever I'm looking at, automatically creating a hyperlink back to the file, e-mail, Web page or even IRC session (Figure 2). Even newbies can add support for new tools, thanks to extensive examples. Planner.el's ability to hyperlink to my mail messages is the only way I can impose order on the thousands of messages in my mail archive!

I like reviewing my week to see what I have accomplished. Because it's easy to view completed tasks, I can write accom-

Figure 1. You can create a task using a small text prompt in your regular editing window.

Figure 2. Hyperlinked tasks give you an easy way to refer back to information related to a task, whether in the form of a file, mail message, Web page or IRC session.

Figure 3. Publishing to the Net lets you check your tasks from any platform, anywhere.

plishment reports without struggling to remember what I did the other day. Seeing a lot of crossed-out tasks for today also is a great morale booster. As a nifty bonus, I can keep detailed logs of how much time I spend on each task or project—great for billing time, improving my time estimates or simply finding out how (un)productive I am each day.

Manageable, not Overwhelming

I like keeping my task list short. I typically have fewer than ten tasks on my task list for any given day. I like scheduling tasks for particular days and organizing them according to projects, keeping my daily task list small and manageable. When I feel particularly productive, it's easy to reschedule more tasks onto today's page.

I break tasks down into bite-size bits to simplify keeping track of my progress and to motivate me to work. When tasks are of a manageable size, they're much easier to work on. Instead of goofing off, I find myself picking the next small task from my list and working on it.

Trustworthy

I need a system that can keep track of small tasks as well as large projects. Because Planner.el is only a keystroke away and I use it for all of my tasks, I trust that it holds all the things I need to remember. I made Planner.el the first thing that shows up when I turn on my computer, and I check it at least once a day. Knowing that all of my reminders are safe and can be checked easily from one place definitely takes a load off my mind.

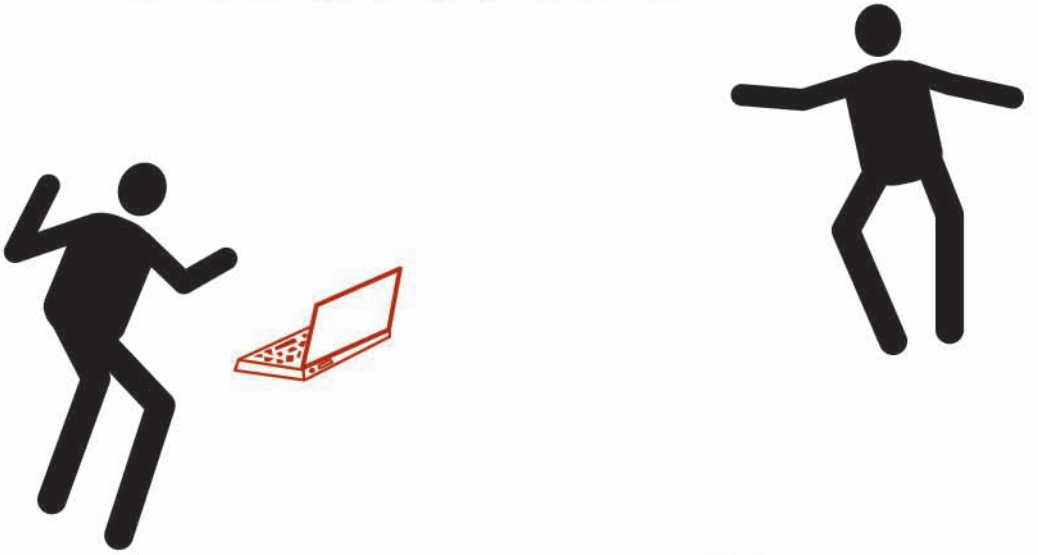
It's also easy for me to back up my files. Because Planner.el uses plain-text files, I don't have to worry about corrupted data. If some experimental code makes Planner.el unusable for me, I still can use any text editor to manage my plans. In addition, it's easy to publish my task list and notes as HTML (Figure 3), so if something happens to my laptop, I can check my TODOs using any computer with Net access.

Extremely Customizable

My method of planning has really changed over the years. I went from micromanaging my schedule by assigning specific times to tasks to keeping an unsorted list on my day page. I tried both keeping one big list of tasks and using projects to group together related tasks. Sometimes I think up weird things, too, such as having my computer automatically display a fortune cookie whenever I finish a task.

This is where Planner.el really shines. Because it's built on top of Emacs, I can change anything I want through a simple, easy-to-learn programming language. I've tweaked it to fit not only my planning style but also my little quirks. Although my

FREE SOFTWARE
FOUNDATION



be FREE

Join us at member.fsf.org

Although a request tracker requires a lot of setup time and effort, you reap the benefits of a solid project management system.

planning style has changed much in the past three years, being able to replace bits of Planner.el and add new features has made it possible for Planner.el to grow along with me.

Things to Remember

There are many ways to manage your tasks, so spend some time finding one that fits you. Here are a few things to remember:

- Make it as easy as possible. Use keyboard shortcuts and scripts to simplify task creation and review.
- Don't get overwhelmed. Keep your task list short and simple. Don't drown in hundreds of TODOs or choke on intimidating tasks.
- Fill in the cracks. Put all of your important tasks in there. If you can, put minor tasks in as well. Check your list regularly.
- Hack your system. Keep an eye out for ways to improve your way of planning. Don't spend too much time hacking your system and not enough time actually accomplishing your TODOs, however.

Have fun!

Resources for this article: www.linuxjournal.com/article/8461.

Sandra Jean Chua—or Sacha, as she commonly is known—maintains Planner.el and is absolutely crazy about it. Her blog and TODO list is at **sacha.free.net.ph**. Write her at sacha@free.net.ph with your productivity tips and way of working!

LETTERS CONTINUED FROM PAGE 8

sed Tip

I read with great interest the article by Larry Richardson entitled "Text Manipulation with sed" [July 2005]. We're in the early stages of porting a MS Access application to Linux using MySQL and Qt to build the GUI. I've recently been working with sed, so the article hit the spot, so to speak.

One thing I noted missing from the article is the fact that when writing commands that substitute one string with another, for example, using slashes to separate the from string and the to string is only one option. Actually, any unique character could be used as a separator. In the example on page 81 (first column, second example), which reads:

```
cat customer.txt \
sed -e 's/Sam Jones/Samuel Jones/' >
customer.txt
```

The slashes could be replaced by the plus (+) sign (or most any other unique character) as in:

```
cat customer.txt \
sed -e 's+Sam Jones+Samuel Jones+' >
customer.txt
```

I've found this useful when I'm editing a file containing paths, such as /etc/exports. In this case, the command:

```
cat /etc/exports |
sed -e 's/\/windows/\/winNT/'
>/etc/exports
```

reduces to:

```
cat /etc/exports |
sed -e 's+/windows+/winNT+'
>/etc/exports
```

--
Bill Lugg

Patent Searching

I've liked open software since the time it had no name yet, when you were nearly a hacker because you had a 1,200-baud modem and a dial-up connection to some friend's BBS.

I'm writing from the old continent, where a battle is being played between software companies who want the ratification by the EU of the right for software patents and developers who don't want to have their mind and ideas limited.

And now the question: I suppose that any person should be granted the same rights as anyone else, but in the fight between open software and non-open software there is a disparity. Any company developing non-open software can check any open software to look for similar, copyright infringing, code. The same right is not granted to the counterpart. How can we be sure that Microsoft, Oracle, SAP or anyone else, is not importing code and concepts from open-source software? Maybe this issue has been discussed many times, but I couldn't find a definitive answer to it.

--
Andrea Rui

GPL violators do sometimes get caught. See gpl-violations.org *for examples.—Ed.*

LDAP Question

I just found your OpenLDAP article on www.linuxjournal.com [July 2005]. I notice you include the nis.schema, but you don't use NIS at all, correct? Thanks for any clarification.

--
Jiann-Ming Su

Craig Swanson replies: *thank you for your interest in "OpenLDAP Everywhere Revisited". You are correct that we are not using Network Information System (NIS). The nis.schema provides several essential attribute types that are used in our LDAP directory. For example, nis.schema defines the LDAP entries that take the place of fields in /etc/passwd. For background information on the history of NIS and LDAP, see the IETF RFC2307, "An Approach for Using LDAP as a Network Information Service".*

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.

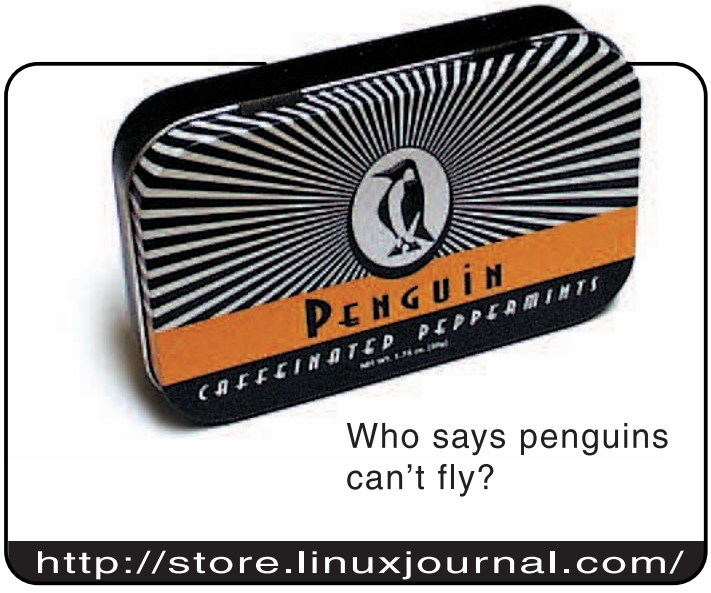
Linux Support Technician

Aberdeen LLC, a growing Los Angeles area server manufacturer is looking for a Linux expert.

Duties include software and hardware pre-sale and post-sale technical support along with system assembly, configuration and testing. Some technical writing. Linux certification a plus. Full time position with excellent salary and benefits.

Fax or email resume:
(562) 695-5570
hr@aberdeeninc.com

ABERDEEN



The Universal Internet Time Source

If your computer's clock is wrong, scheduled tasks, such as backup and virus scans, might run during the day, or your log files could become worthless. The pool.ntp.org project has an answer. **BY ADRIAN VON BIDDER**

In theory, setting a computer's clock over the network is easy: simply send a query to the time server and get the current time in return. For low-precision usage on a trusted network, this process does indeed work fine, as demonstrated by the old UNIX "time" protocol. For today's Internet, however, and for millisecond (ms) or even sub-ms precision, problems such as authentication, reliability of the time servers and network delays need to be considered. This is where the Network Time Protocol (NTP), with its reference implementation, steps in. The specification and the reference implementation are being written by Professor David Mills of the University of Delaware, his graduate students and many other volunteers.

To allow everybody to use NTP to synchronise computers' clocks over the public Internet, Prof. Mills has long maintained a list of public time servers. Most of these servers are operated by universities or national standardisation organisations. Today, this list is maintained by the NTP Public Services Project, under the umbrella of the Internet Systems Consortium. However, the growth of the Internet and the prevalence of small, cheap appliances, such as cable or DSL routers, with built-in NTP clients, lead to a rapidly growing load on these public time servers. One of the most famous cases involved a severe firmware problem in a range of such devices, resulting in more than 150Mbps of NTP traffic to the University of Wisconsin's NTP server.

After reading the discussion of one time server operator's request to be taken off the public time servers list, I wondered if there was a better approach to this whole problem—instead of having tens of thousands of clients targeting one single time server, the load should be distributed on many different time servers all over the network. So I went ahead and created the original time.fortytwo.ch DNS round-robin in January 2003. The project quickly acquired many interested volunteers and was well received by Prof. Mills and his team. It soon became the pool.ntp.org project with a somewhat more official status.

The Road Ahead

During the next two years, the project continued to grow, thanks to all the people who mentioned it in various Web forums, HOWTO documents and the like. Today, the project consists of more than 300 servers, offering service to tens of thousands of clients, in a very rough estimate. Also, pool.ntp.org is now the default time server in several operating system distributions, including Debian GNU/Linux, NetBSD and Gentoo Linux.

So far, the growth in servers could more or less match the growth of the user base of the project. However, the future remains challenging, and discussions on the project's discussion mailing list have shown that the project needs to deal with an inherent conflict between providing easy service for as many clients as possible and assuring good quality of the time servers participating in the project. That aside, the big challenges for the near and medium future are:

- IPv6 integration.
- More automation—currently, I process server additions and removals mostly manually.
- Better, more novice-friendly documentation on the Web.
- Of course, we always need more servers too.
- And above all, we need to deal with abusive clients. In one example, the six worst clients were responsible for 25% of the traffic on one time server.

Although the first three items are not technically difficult and the "getting more servers" plan should see a big leap ahead with the publication of this article, we don't currently have a good plan to educate the hundreds of users with sub-optimally configured clients. Due to their number, they are a serious problem for the project. At the same time, the bandwidth per client is small enough that the big ISPs' abuse departments are not prepared to help in any way.

In the medium to long term, we will need to face the issue that DNS round-robin, as currently implemented, is not a good solution for load balancing on the scale of several hundred servers with a hundred thousand or more clients. Wide deployment of IP multicast together with the existing multicast support in ntpd would be a good solution to this problem, but obviously not one the NTP and pool.ntp.org crew can deploy on their own. Another possible solution is to make the ntpd dæmon aware of the pool.ntp.org project and, in some generic way, similar such databases, and have the dæmon configure itself to use such a resource.

Finally, on a personal note, I honestly can say that it was fun to get this project started and see it grow, but I now see the need for somebody new, with fresh ideas, to take over from here. Indeed, as I write this, I am talking with several people about the project's future, and I am certain that the involvement of a new "father figure" will do the project much good as new ideas are looked at and implemented by a new crew.

Resources for this article: www.linuxjournal.com/article/8454.

Adrian von Bidder graduated with a degree in computer science from the Federal Institute of Technology in Zurich, Switzerland, in 2004. He is running the pool.ntp.org project in his spare time. His day job is developing the SEPP e-mail encryption gateway at Onaras AG in Wettingen, Switzerland. He can be contacted at avbidder@fortytwo.ch.

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**

The World's Fastest InfiniBand Switch

From a Company You've Trusted for 23 Years

▲ 72 Port FasTree™ Configuration

Microway's **FasTree**™ DDR InfiniBand switches run at 5GHz, twice as fast as the competition's SDR models. FasTree's non-blocking, flow-through architecture makes it possible to create 24 to 72 port modular fabrics which have lower latency than monolithic switches. They aggregate data modulo 24 instead of 12, improving nearest neighbor latency in fine grain problems and doubling the size of the largest three hop fat tree that can be built, from 288 to 576 ports. Larger fabrics can be created linking 576 port domains together.

Working with PathScale's InfiniPath HTX Adapters, the number of hops required to move MPI messages between nodes is reduced, improving latency. The modular design makes them useful for SDR, DDR and future QDR InfiniBand fabrics, greatly extending their useful life. Please send email to fastree@microway.com to request our white paper entitled *Low Latency Modular Switches for InfiniBand.*

Microway's **QuadPuter**® includes four AMD single or dual core Opteron™ processors, 1350 Watt redundant power supply, and up to 5 redundant, hot swap hard drives-all in 4U. One of the most powerful processing platforms in the HPC industry, QuadPuter can serve as a cluster node or a standalone supercomputer. Constructed with stainless steel, its RuggedRack™ architecture is designed to keep the processors and memory running cool and efficiently. The power supply exhaust does not mix with air in the motherboard chamber. Hard drives are cooled with external air and are front-mounted along with the power supply for easy access and removal. The RuggedRack™ is available with an 8-way motherboard, dual-core Opterons and up to 128 GB of memory for power- and memory-hungry SMP applications.

◀ QuadPuter® Navion™ with Hot Swap, Redundant Power & Hard Drives

Call us first at 508-746-7341 for quotes and benchmarking services. Find technical information, testimonials, and newsletter at microway.com.

Visit us at SC2005 Seattle

AMD 64 Opteron

PathScale™ Accelerating Cluster Performance™

Microway®
Technology you can count on℠

508.746.7341 microway.com

Cyclades AlterPath™ System makes
AlterPath™ System
out-of-band manager
blade manager
console
kvm over IP
Intellisent power
IPMI
A+
For a FREE copy of Enterprise Management Associates' white paper
Increased IT Operations Effectiveness With Cyclades AlterPath™ System for the Out-of-Band Infrastructure,
please visit us at
www.cyclades.com/ema