

eyeOS | Bitlbee | KOffice 2.0 | Ksplice | Rootkits | squidGuard

LINUX JOURNAL

Since 1994, The Original Magazine of the Linux Community

AUGUST 2009 | ISSUE 184 | www.linuxjournal.com

KERNEL CAPERS

Avoid **setuid**
Root Exploits

Ksplice: No More
Reboots!

Real-Time
Kernel Scheduler

Completely Fair
Scheduler

Defend against
/dev/mem Attacks

Using **Fixtures**
and **Factories** in
Rails Applications

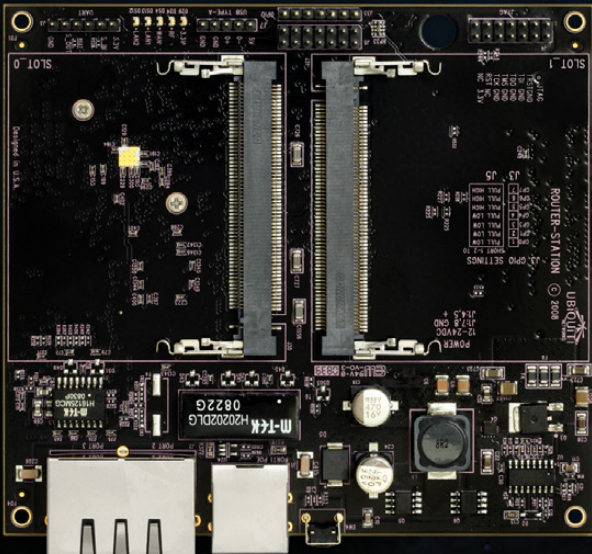
PLUS

REVIEWED:
KOffice 2.0

Point/Counterpoint
Kyle and Bill Debate
the Merits of Twitter



Embedded Wireless Dream Machines.

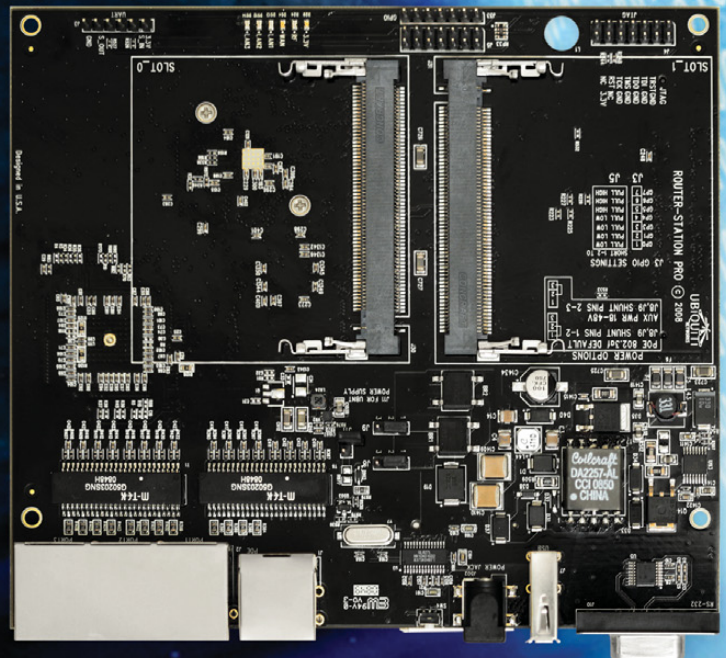


RouterStation

Featuring a fast 680MHz MIPS 24K CPU, 64MB RAM, and 16MB Flash; RouterStation provides an excellent horsepower for a variety of processor intensive multi-radio system applications.

Up to 3 mini-PCI radios, 3 10/100 ethernet interfaces, a 5A power supply for multiple hi-power card support, USB 2.0, and enhanced temperature operating performance and ethernet ESD protection for carrier applications.

MSRP \$59



RouterStation Pro

In response to the outstanding demand for our initial RouterStation OEM platform, Ubiquiti Networks announces the RouterStation Pro. Breakthrough Price/Performance with a \$79 USD MSRP.

Pro Version Enhancements:

- 48V 802.3af Power Over Ethernet
- 4-Port Gigabit Ethernet Switch
- 256MB RAM
- On Board SDIO Support
- On Board, USB 2.0, RS232/dB9, and DC power jacks

MSRP \$79

1&1 Summer Specials:



.us Domain Names

\$2.99 ~~\$8.99~~
first year per year

No Setup Fee!

WEB HOSTING

Everything you need for a professional website.



1&1® BUSINESS PACKAGE

~~\$9.99~~
per month

3 months FREE!*

SERVERS

Powerful hardware designed for high performance needs.



1&1® DUAL CORE XL

~~\$199.99~~
per month

3 months FREE!*

E-COMMERCE

Set up your online store and start selling!



1&1® ADVANCED ESHOP

~~\$49.99~~
per month

3 months FREE!*

More special offers are available online.

For details, visit www.1and1.com



Now accepting

PayPal™

*Offers valid as of July 1, 2009. 12 month minimum contract term required. Setup fee and other terms and conditions may apply. Visit www.1and1.com for full promotional offer details. Private domain registration not available with .us domains. Server prices based on Linux servers. Program and pricing specifications and availability subject to change without notice. 1&1 and the 1&1 logo are trademarks of 1&1 Internet AG, all other trademarks are the property of their respective owners. © 2009 1&1 Internet, Inc. All rights reserved.



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

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



CONTENTS

AUGUST 2009
Issue 184

FEATURES

50 SAY GOODBYE TO REBOOTS WITH KSPLICE

It's not a dream!
Waseem Daher

54 REAL-TIME LINUX KERNEL SCHEDULER

Do real time
with Linux and
the -rt patchset.
Ankita Garg

62 MAKING ROOT UNPRIVILEGED

Change the way
you think.
Serge Hallyn

68 COMPLETELY FAIR SCHEDULER

Linux's latest
scheduler makeover.
Chandandeep
Singh Pabla

ON THE COVER

- Defend against /dev/mem Attacks, p. 72
- Using Fixtures and Factories in Rails Applications, p. 18
- Avoid setuid Root Exploits, p. 62
- Ksplice: No More Reboots!, p. 50
- Real-Time Kernel Scheduler, p. 54
- Completely Fair Scheduler, p. 68
- Reviewed: KOffice 2.0, p. 46
- Point/Counterpoint: Kyle and Bill Debate the Merits of Twitter, p. 76

HOW MUCH STORAGE DO YOU NEED?



CAPACITY

Performance tuned storage.
Up to 50TB in a single storage server.

EFFICIENCY

Reduce operating costs.
Best TB/\$ ratio.

SCALABILITY

Easily expand storage to well beyond
400TB via XDAS and JBOD units.

ABERDEEN STIRLING SCALABLE STORAGE SERVERS



- 2x Quad-Core Intel® Xeon® Processor 5500 Series featuring Intel® Microarchitecture, codenamed Nehalem
- Up to 96GB 1333MHz DDR3 Memory
- Supports both SAS & SATA Storage Drives
- RAID 0, 1, 5, 6, 10, 50, 60 Capable
- Redundant Power Supply
- SAS & iSCSI Expansion Ports
- Windows & Linux NAS Available
- 5-Year Warranty

3U 8TB Starting at	\$4,495
4U 16TB Starting at	\$7,595
5U 24TB Starting at	\$9,995
6U 32TB Starting at	\$13,495
8U 50TB Starting at	\$18,595

EXPAND CAPACITY TO OVER 400TB



- Daisy-Chain DAS Units and JBOD Expansion Boxes
- 2U, 3U, 4U Enclosures Available
- RAID 0, 1, 5, 6, 10, 50, 60 Capable
- SATA & SAS Drive Support
- 5-Year Warranty

16TB JBOD Expansion	\$5,995
16TB DAS	\$8,995
24TB DAS	\$12,495



CONTENTS AUGUST 2009

Issue 184

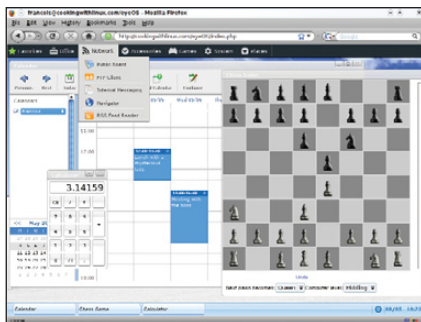
COLUMNS

18 REUVEN M. LERNER'S AT THE FORGE

Fixtures and Factories

24 MARCEL GAGNÉ'S COOKING WITH LINUX

The Case of the Missing OS



30 DAVE TAYLOR'S WORK THE SHELL

Looking More Closely at Letter and Word Usage

32 MICK BAUER'S PARANOID PENGUIN

Building a Secure Squid Web Proxy, Part IV

38 KYLE RANKIN'S HACK AND /

What Really IRCs Me: Instant Messaging

76 KYLE RANKIN AND BILL CHILDERS' POINT/COUNTERPOINT

Twitter

80 DOC SEARLS' EOF

The Mania of Owning Things

INDEPTH

72 ANTHONY LINEBERRY ON /DEV/MEM ROOTKITS

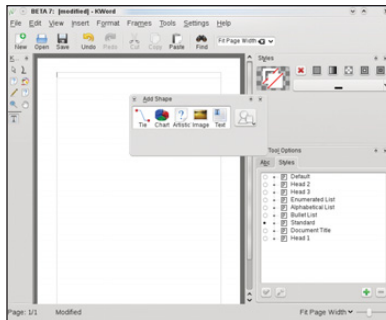
The epic battle between good and evil continues.

Mick Bauer

REVIEW

46 KOFFICE 2.0

Bruce Byfield



IN EVERY ISSUE

8 CURRENT_ISSUE.TAR.GZ

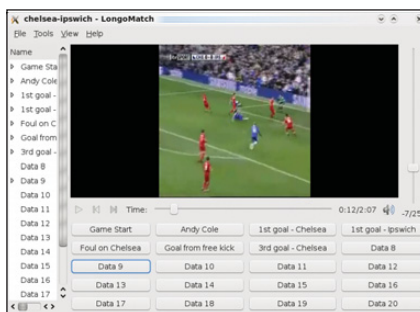
10 LETTERS

14 UPFRONT

40 NEW PRODUCTS

42 NEW PROJECTS

65 ADVERTISERS INDEX



42 LONGOMATCH



42 KANATEST

Next Month

CROSS-PLATFORM DEVELOPMENT

Someday all computers will run Linux, but until then, there's cross-platform work to be done. And, some of the best tools—actually, most of the tools—for cross-platform development are open source.

Next month, we'll be looking at some of those tools, including Qt, the cross-platform application development framework that powers KDE. We'll be looking at Lazarus, the open-source Delphi-like RAD development environment as well. Also on tap is an article on Titanium, a cross-platform technology that uses HTML and JavaScript (or Python, or Ruby, ...) for developing rich desktop applications.

Plus, we have an interview with the lead developer of Google Chrome, the "secret" new browser from Google. All this and more coming next month in *Linux Journal*.

Polywell Linux Solutions

More Choices, Excellent Service, Great Value!
Serving the Industry for More Than 20 Years

Quiet Performance NAS Storage



8TB \$1,999

12TB \$2,999

24TB \$5,999

- Dual Gigabit LAN
- RAID-5, 6, 0, 1, 10
- Hot Swap, Hot Spare
- Linux, Windows, Mac
- E-mail Notification
- Tower or Rackmount



LD-001



Silent Eco Green PC

NVIDIA® nForce Chipset and GeForce Graphics
Energy efficient, Quiet and Low Voltage Platform. starts at **\$199**

The Best Terminal PC

Fanless Silent Mini-ITX PC

1G DDR2, Solid State Drive starts at **\$299**

Nvidia Ion Based PC is now Available: ITX-9400

ITX-30A with PCI Riser



ITX-10A



ITX-20A with Slim DVD



ITX-1000 with WiFi Option

4U-24Bay 48TB Storage Server

Hardware RAID-6, NAS/iSCSI/SAN Storage
Mix SAS and SATA, 4 x GigaLAN or 10Gbit LAN



4024SS

Mini-1U Server for Data Center ISP

Intel Dual-Core or Quad-Core Processor, Dual GigaLAN
4GB to 8GB RAM, 2 x 500GB RAID HD
Linux Server Starts at **\$499**



1002SC-1U

Polywell OEM Services, Your Virtual Manufacturer

Prototype Development with Linux/FreeBSD Support
Small Scale to Mass Production Manufacturing
Fulfillment, Shipping and RMA Repairs

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

888.765.9686

linuxsales@polywell.com

www.polywell.com/us/Lx



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

NVIDIA, nForce, GeForce and combinations thereof are trademarks of NVIDIA Corporation. Other names are for informational purposes only and may be trademarks of their respective owners.

LINUX JOURNAL™

Since 1994: The Original Magazine of the Linux Community

Digital Edition Now Available!

Read it first

Get the latest issue before it
hits the newsstand

Keyword searchable

Find a topic or name
in seconds

Paperless archives

Download to your computer for
convenient offline reading

Same great magazine

Read each issue in
high-quality PDF

Try a Sample Issue!

www.linuxjournal.com/digital



LINUX JOURNAL

Executive Editor	Jill Franklin jill@linuxjournal.com
Senior Editor	Doc Searls doc@linuxjournal.com
Associate Editor	Shawn Powers shawn@linuxjournal.com
Associate Editor	Mitch Frazier mitch@linuxjournal.com
Art Director	Garrick Antikajian garrick@linuxjournal.com
Products Editor	James Gray newproducts@linuxjournal.com
Editor Emeritus	Don Marti dmarti@linuxjournal.com
Technical Editor	Michael Baxter mab@cruzio.com
Senior Columnist	Reuven Lerner reuven@lerner.co.il
Chef Français	Marcel Gagné maggagne@salmar.com
Security Editor	Mick Bauer mick@visi.com
Hack Editor	Kyle Rankin lj@greenfly.net
Virtual Editor	Bill Childers bill.childers@linuxjournal.com

Contributing Editors

David A. Bandel • Ibrahim Haddad • Robert Love • Zack Brown • Dave Phillips • Marco Fioretti
Ludovic Marcotte • Paul Barry • Paul McKenney • Dave Taylor • Dirk Elmendorf

Proofreader Geri Gale

Publisher Carlie Fairchild
publisher@linuxjournal.com

General Manager Rebecca Cassidy
rebecca@linuxjournal.com

Sales Manager Joseph Krack
joseph@linuxjournal.com

Associate Publisher Mark Irgang
mark@linuxjournal.com

Webmistress Katherine Druckman
webmistress@linuxjournal.com

Accountant Candy Beauchamp
acct@linuxjournal.com

Linux Journal is published by, and is a registered trade name of, Belltown Media, Inc.
PO Box 980985, Houston, TX 77098 USA

Reader Advisory Panel

Brad Abram Baillo • Nick Baronian • Hari Boukis • Caleb S. Cullen • Steve Case
Kalyana Krishna Chadalavada • Keir Davis • Adam M. Dutko • Michael Eager • Nick Faltys • Ken Firestone
Dennis Franklin Frey • Victor Gregorio • Kristian Erik • Hermansen • Philip Jacob • Jay Kruiuzenga
David A. Lane • Steve Marquez • Dave McAllister • Craig Oda • Rob Orsini • Jeffrey D. Parent
Wayne D. Powell • Shawn Powers • Mike Roberts • Draciron Smith • Chris D. Stark • Patrick Swartz

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
Bruce Perens
Bdale Garbee, Linux CTO, HP
Danese Cooper, Open Source Diva, Intel Corporation

Advertising

E-MAIL: ads@linuxjournal.com
URL: www.linuxjournal.com/advertising
PHONE: +1 713-344-1956 ext. 2

Subscriptions

E-MAIL: subs@linuxjournal.com
URL: www.linuxjournal.com/subscribe
PHONE: +1 818-487-2089
FAX: +1 818-487-4550
TOLL-FREE: 1-888-66-LINUX
MAIL: PO Box 16476, North Hollywood, CA 91615-9911 USA
Please allow 4-6 weeks for processing address changes and orders
PRINTED IN USA

LINUX is a registered trademark of Linus Torvalds.



EtherDrive®

The AFFORDABLE Network Storage

Now
Supporting
VMware®
ESX 3.5



Fibre Channel speeds at Ethernet prices!

Is your budget shrinking while your network storage needs are growing? Are you suffering from “sticker shock” induced by expensive Fibre Channel and iSCSI storage area network solutions? EtherDrive® SAN solutions offer Fibre Channel speeds at Ethernet prices! Starting at just \$1,995 for a 4TB system, EtherDrive® is the **affordable** storage area network solution. With sustained access speeds from 200MBytes/sec to over 600MBytes/sec, EtherDrive® SAN solutions are **fast**. From a 4TB single storage appliance to multi-PetaByte system by simply adding more storage appliances, EtherDrive® SAN solutions are **scalable**. From a single storage appliance to a network of sophisticated virtualized storage LUNs, EtherDrive® SAN solutions embrace **virtualization**.

Coupling Ethernet technology with SATA hard disk drives, EtherDrive® SAN solutions exploit commodity components to deliver **affordable, fast** storage area network solutions that keep more green in your wallet! Whether you use your own SATA compliant disk drives or our certified enterprise class disk drives, you are in control! EtherDrive® SAN solutions accept standard SATA hard disk drives. Ethernet and SATA disk drives - two proven technologies in one **affordable, fast** storage area network solution - EtherDrive®.

EtherDrive® SAN solutions use the open ATA-over-Ethernet (AoE) lightweight network storage protocol. Simple. Easy to understand. Easy to use. AoE uses Ethernet to transport ATA disk commands without the burden of TCP/IP overhead, thereby enabling disk drives to become AoE devices connected directly to an Ethernet network. An AoE device can be a single physical disk or a logical device made up of multiple disks. An EtherDrive® SAN appliance is an AoE target device.

Finally, an **affordable, fast** storage area network solution for your VMware® ESX 3.5 installation. The EtherDrive® VMware ESX Host Bus Adapter empowers ESX with AoE technology to deliver EtherDrive® SAN solutions for your VMware ESX 3.5 installation.

Shipping EtherDrive® RAID solutions since 2004, Coraid boasts thousands of satisfied customers spanning a broad spectrum of the market including enterprise, government, educational institutions, and hosting service providers. **Call today** to order your EtherDrive® solution, and join the ranks of our thousands of satisfied customers!

Call 1.877.548.7200
or visit our website at
www.coraid.com
International: +1.706.548.7200



vmware® | technology alliance
PARTNER

ESX 3.5 compatible EtherDrive® HBA



SHAWN POWERS

Kentucky Fried Linux

For most people, the word kernel inspires visions of corn, wheat or possibly fried chicken. Here in the Linux world, although we still might appreciate The Colonel's 11 herbs and spices, kernel means something much more profound. The kernel *is* Linux. Sure we add lots of fancy programs, interfaces and command-line tools, but in the end, Linux is the kernel. This month, we focus on it.

Do you ever smugly brag about the uptime of your Linux machines to your Windows friends? I don't know about you, but every time I do, either the power goes out or I have to reboot due to a kernel upgrade. Thankfully, Waseem Daher shows us a bit about Ksplice. Using Ksplice, software updates can be applied without rebooting the Linux machine. Add to that a battery backup, and we can all brag to our friends about uptimes. Unless, that is, we're running an old kernel and they come over 498 days after we first started. (Uptime wraparound hasn't been a problem for a while, but most of us still remember it.)

One of the jobs the kernel has is to schedule CPU time for different processes. We have a few different looks at kernel schedulers: a real-time scheduler that Ankita Garg explains and a "Completely Fair" scheduler that Chandandeep Singh Pabla tells us about. One of the great strengths of the Linux kernel is its flexibility, and that should be fairly evident after reading this month's issue.

All Linux admins worth their salt know that a properly maintained Linux machine is a fairly secure beast. "Fairly secure" usually isn't satisfactory, however, and that's where people like Mick Bauer come into play. This month, he continues his series on setting up a secure proxy, but he also interviews Anthony Lineberry about /dev/mem rootkits. The best security specialist is a paranoid security specialist, and Mick does his best to worry us all a bit.

If all this kernel talk is beginning to worry you that this issue has nothing for you, fear

not! I'd be lying if I claimed to do any work with the kernel anymore. In fact, not since the days of compiling Debian kernels for my PowerPC hardware have I even used anything but the stock kernel that comes with my distro. We realize you might fit into that boat as well, so we've stuffed a ton of other stuff between these covers just for you. (And maybe for me.)

Kyle Rankin shows us how to join the instant-messaging bandwagon without ever leaving the comfort of our IRC windows. With Bitlbee, you can pretend everyone on the planet uses IRC, even if they're using the dreaded MSN Messenger. To follow that up, Kyle and Bill Childers are back to their spating. This time, they're arguing over the usefulness of Twitter. As a Twitter user myself, I think I lean toward Bill's side this month, but feel free to choose for yourself.

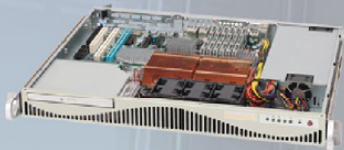
Dave Taylor takes us back to the command line as we dissect the English language a bit. Of course, we have the computer do the dirty work, but in the end, you'll learn a bit about scripting language and the English language. Reuven M. Lerner teaches us about Fixtures and Factories in your Rails projects. If you program in Rails and work with databases, you won't want to miss Reuven's column.

We didn't stop there. Marcel Gagné demonstrates eyeOS—an entire operating system you can control from a Web browser. You get to set up a little cloud computing system of your very own! Add to that our normal list of reviews, product announcements and regular columns, and this issue is bound to please even the most obscure Linux user. So go get your can of corn kernels, a bag of wheat kernels or even a bucket of chicken, and sit back to enjoy this issue of *Linux Journal*. ■

Shawn Powers is the Associate Editor for *Linux Journal*. He's also the Gadget Guy for LinuxJournal.com, and he has an interesting collection of vintage Garfield coffee mugs. Don't let his silly hairdo fool you, he's a pretty ordinary guy and can be reached via e-mail at shawn@linuxjournal.com. Or, swing by the #linuxjournal IRC channel on Freenode.net.

YOUR HIGH PERFORMANCE COMPUTING HAS ARRIVED.

The ServersDirect® Systems with the Intel® Xeon® Processor helps you simplify computing operations, accelerate performance and accomplish more in less time



STARTING AT **\$899**

ENTRY LEVEL INTELLIGENT SERVER

SDR-S1341-T00 is among our most cost-effective 1U Xeon Servers, and it is ideal for large high-performance computing deployments



STARTING AT **\$959**

APPLICATION SERVER

Refresh your servers with new **SDR-S1337-T02** powered by Intel® Xeon® processor 5500 series, based on intelligent performance, automated energy efficiency and flexible virtualization.



SDR-S1343-T04
STARTING AT **\$1,099**

1U INTEL® XEON® PROCESSORS 5500 SERIES SERVER W/ 4X 3.5" HOT-SWAP SATA DRIVE BAYS

- Supermicro 1U Rackmount Server with 560W Power Supply
- Supermicro Server Board w/Intel® 5520 Chipset
- Support up to Dual Intel® 5500 series Xeon® Quad/Dual-Core, with QPI up to 6.4 GT/s
- Support up to 96GB DDR3 1333/ 1066/ 800MHz ECC Reg.DIMM
- 4x 3.5" Hot-swap SATA Drive Bays
- Intel® 82576 Dual-Port Gigabit Ethernet Controller



SDR-S2311-T08
STARTING AT **\$1,159**

2U INTEL® XEON® PROCESSORS 5500 SERIES SERVER W/ 8X 3.5" HOT-SWAP SAS/SATA BAYS

- Supermicro 2U Rackmount Server with 560W Power Supply
- Supermicro Server Board w/Intel® 5500 Chipset
- Support up to Dual Intel® 5500 series Xeon® Quad/Dual-Core, with QPI up to 6.4 GT/s
- Support up to 24GB DDR3 1333/ 1066/ 800MHz ECC Reg.DIMM
- 8x 3.5" Hot-swap SATA Drive Bays
- Dual Intel® 82574L Gigabit Ethernet Controller



SDP-IP308-T10
STARTING AT **\$1,599**

PEDESTAL INTEL® XEON® PROCESSORS 5500 SERIES SERVER W/ 10X HOT-SWAP (OPT.) SATA BAYS

- Intel Pedestal Chassis w/ 750W (1+1) Power Supply
- Supermicro Server Board w/Intel® 5520 Chipsets
- Support up to Dual Intel® 5500 series Xeon® Quad/Dual-Core, with QPI up to 6.4 GT/s
- Support up to 96GB DDR3 1333/ 1066/ 800MHz ECC Reg./unbuffered DIMM
- Option 10x 3.5" Hot-swap SATA Bays
- Intel® 8257EB Dual-port Gigabit Ethernet Controller



SDR-S4313-T24
STARTING AT **\$1,899**

4U INTEL® XEON® PROCESSORS 5500 SERIES SERVER W/ 24X 3.5" HOT-SWAP SAS/SATA BAYS

- Supermicro 4U Rackmount 900W (1+1) Red. Power Supply
- Supermicro Server Board w/ Dual Intel® 5520 Chipsets
- Support up to Dual Intel® 5500 series Xeon® Quad/Dual-Core, with QPI up to 6.4 GT/s
- Support up to 144GB DDR3 1333/ 1066/ 800MHz ECC Reg. DIMM
- 24x 3.5" Hot-swap SATA Drive Bay
- Intel® 82576 Dual-port Gigabit Ethernet Controller



SDR-S3305-T16
STARTING AT **\$1,979**

3U INTEL® XEON® PROCESSORS 5500 SERIES SERVER W/ 16X 3.5" HOT-SWAP SAS/SATA BAYS

- 3U Rackmount Server with 1+1 900W Red. Power Supply
- Supermicro Server Board w/ Dual Intel® 5520 Chipsets
- Support up to Dual Intel® 5500 series Xeon® Quad/Dual-Core, with QPI up to 6.4 GT/s
- Support up to 96GB DDR3 1333/ 1066/ 800MHz ECC Reg.DIMM
- 16x Hot-swap SAS/SATA Drive Bays
- Intel® Dual 82576 Dual-Port Gigabit Ethernet (4 ports)



SDR-C9303-T50
STARTING AT **\$4,339**

9U INTEL® XEON® PROCESSORS 5500 NEHALEM SERIES SERVER W/ 50X HOT-SWAP SATA II / SAS BAYS

- 9U Chassis with 1620W Redundant Power Supply
- Supermicro Server Board w/ Dual Intel® 5520 Chipsets
- Support up to Dual Intel® 5500 series Xeon® Quad/Dual-Core, with QPI up to 6.4 GT/s
- Support up to 144GB DDR3 1333/ 1066/ 800MHz ECC Reg. DIMM
- 50 x 3.5" Internal SATA Drives Trays
- Intel® 82576 Dual-port Gigabit Ethernet Controller



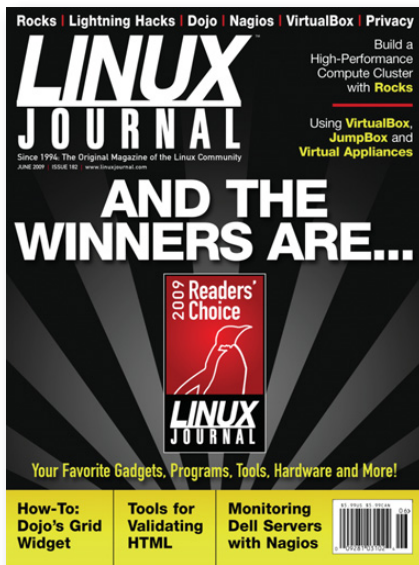
SERVERS DIRECT CAN HELP YOU CONFIGURE YOUR NEXT HIGH PERFORMANCE SERVER SYSTEM - CALL US TODAY!

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

1.877.727.7886 / www.ServersDirect.com

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





State of Linux Audio

Hey guys, great mag, keep up the good work. I'm an audio recording engineer and an avid Linux user. I'm currently running Windows, OS X and Ubuntu 9.04 on my MacBook Pro. I use Linux for pretty much everything I do, but it's really nice in my industry to have an OS X machine around. My primary recording platform is Steinberg's Cubase. I love the software. I go to great length to avoid using ProTools professionally. Also, I run Cubase in Windows XP. I refuse to pay \$2,500 for a computer that could be built for around \$1,000, so OS X is out (plus, I hate the operating system). So my question is this. With the Vista debacle, I don't have much hope for Windows 7 being a viable solution as a multimedia production platform. Not wanting to go to OS X, why haven't manufactures of professional audio hardware and software compiled for Linux? What are they waiting for? Why, oh why, MOTU, can I not get drivers for my 424 PCI in Linux? Why, oh why, Steinberg, can I not run Cubase and Nuendo on Ubuntu or Red Hat? There are plenty of home recording users and audio professionals like myself that need an alternative to Apple. I love my MOTU hardware, but I've always heard that MOTU is very Linux-unfriendly. What are they scared of? I'm not saying we need to open-source anything, but at least give me the choice. I pose this question to MOTU, Steinberg and all the

other companies not tied to Apple (as in Logic Audio). There's no reason why Digidesign couldn't get in the game as well. Ardour is becoming a great piece of software, and Audacity is great as well. But, as long as both of them lack good VST support, OMF transfer and other tools that are essential to what I do, I'm forced to stick to a closed-source solution like Steinberg's Cubase (not that I mind, I love the software). What are we going to do if Windows 7 is a multimedia flop?

--
Michael Russo

Dave Phillips replies: First, thank you for your interest in the future of Linux audio and multimedia development. Indeed, Linux could be the alternative to hardware lock-in, but some significant factors keep it from happening.

Hardware manufacturers have been slow to adapt their products to Linux. This situation is perhaps the most significant factor in the non-acceptance of Linux in the wider professional audio worlds. After all, we can (and do) have the most amazing audio infrastructure, especially with the JACK server, but what good is it if industry-standard hardware won't work with it? A few intrepid manufacturers, such as RME and M-Audio, have entered the arena and have done well with sales to Linux users. Those companies wisely donated the driver spec sheets to the community development teams, and voilà, Linux users have drivers for some pro-audio gear. But, it's not enough.

The major music software houses are another story. We've seen pro-Linux movement from Renoise, Garritan, Reaper and a few other high-profile development houses, but I don't look for the big guns (Cubase, Logic, Pro Tools and so on) to enter the ring any time soon. Steinberg might lead the way if enough pressure is brought to bear upon them, but they need to see that a market exists before they expend the resources to create a Linux version of their products. Incidentally, those manufacturers also might consider Linux to be a support nightmare, although the crew at Renoise seems to be doing

things the right way.

I don't expect the closed-source makers to embrace open-source practices or philosophies. I and many others would be happy to see working Linux versions of the major music packages for Windows, and I suspect that sales could be brisk. However, there is no denying that the Linux audio world is still very small, and any manufacturer who gets into the game at this time must be considered a pioneer.

So what can we do? We can continue to lobby the majors for Linux versions of their software. We can continue to ask hardware-makers for driver specs, and we can more actively support their entry into the Linux audio ecology. Also, the group at linuxaudio.org can act as an intermediary for companies or individuals who want to design an effective strategy for marketing their products to Linux users. We can continue to support those who already support Linux, and we can be vocal about it. We can advise builders that we would spend our money on their products if they supported Linux, and we should tell them what we will be buying instead.

Beyond these methods, I'm open to suggestions. Being harsh and rude won't be very convincing, so we must remain civil even to the most uncivil manufacturers. After all, we can't force anyone to support Linux.

Above all, the manufacturers can't resist market factors. If enough users populate the Linux audio world, the majors eventually will have to admit that their attitudes are costing them real money. Unfortunately, we can't easily draw more users into the fold when hardware choices are so limited. So, we return to the well-worn scenario of the chicken and the egg. From my point of view, it really is mostly about money. Perhaps MOTU has some ingrained anti-FOSS philosophy, but certainly even they would want to sell more of their goods in a larger market.

It's been suggested that what Linux audio really needs is a #1 hit—a song

that sells strongly and has been made with Linux software. The way the music industry works, if that happened, there'd be a boatload of willing converts wanting to get on board this year's hobby horse. And, that would be fine with me, because some percentage of those converts will want to stay on board, thereby permanently enlarging the user base.

By the way, Ardour3 will include support for MIDI edition and for VST/VSTi plugins. The plugin support comes from recent open-source work that has effectively replaced the proprietary code from Steinberg, which means that a VST-enabled Ardour will be freely distributable.

Again, thank you for your interest and remarks, and I welcome further commentary.

Linux Power

I've been reading *LJ* cover to cover since 1999. I absolutely love it, and it's a highlight of my month when it arrives. I've built many machines and currently have about ten Linux boxes (singles, duals, quads) running 24/7. Imagine my electricity bill. I used to buy PSUs on price/performance ratings. This year, my electric bill topped \$300 in the winter (in Texas). Now I look for PSU efficiency ratings, which are sometimes difficult to find. I just found the coolest site ever for determining power supply efficiency: www.80plus.org/manu/psu/psu_join.aspx. I'm sure you already knew about this, but please share it with your readers. It lists efficiencies for all the major manufacturers' power supplies, graphing efficiency vs. load, and it makes selecting a new PSU a no-brainer.

--
Jim Peterman

I must admit, I've never considered buying PSUs for their efficiency either. As my electric bill recently has been more than \$200/month, perhaps it's time I visit that site and begin to shop a little more wisely! Thanks for the tip, and thanks for the kudos.—Ed.

ssh-copy Tip

I have not written to your publication before but felt compelled to comment on Kyle Rankin's, "Lightning Hacks

Strike Twice" [June 2009]. First, kudos to Kyle for showing us new and useful tricks. I appreciate the little time savers like `cd -` that he wrote about.

However, one item he talked about seemed needlessly complex. While discussing the "SSH Key One-Liner", Kyle used `ssh` and a redirected `cat` to append an SSH key to a remote server. I have had to manage dozens of remote servers, and I've found the `ssh-copy` utility to be much more effective. Kyle's example could be simplified as:

```
$ ssh-copy -i ~/.ssh/id_rsa.pub user@server.example.net
```

The nice thing about `ssh-copy` is that it verifies that the key being added to the remote server doesn't already exist, which, of course, Kyle's solution does not.

Keep up the great magazine, and keep giving us great tips. I, for one, love them.

--
Mark K. Zanfardino

Kyle Rankin replies: *Thanks Mark! On my system, that tool appears to be called `ssh-copy-id`. I'm always game for learning an even simpler solution, and it looks like `ssh-copy` (or `ssh-copy-id`, in my case) certainly beats a long one-liner.*

Re: "Free to a Good Home: Junk"

Regarding Shawn Powers' "Free to a Good Home: Junk" [in the May 2009 UpFront section]: it's good concept, and I have some responses. First, I understand that you want to support sister print media (newspapers), but realistically, it makes more sense to "get with the times", and offer the equipment in the free and computer sections of Craigslist if you are in one of its covered markets. Although some papers do not charge for advertising free stuff, still, more tech-oriented people look to Craigslist first.

Second, if you're giving away PCs (desktop or notebook), when in the process of wiping your info from the hard drive (good idea!), install one of the more-common Linux mini-distros. I have found that most of the current mainstream distros are almost as bad as Vista as far as hardware demands,

and they will not install (or run well if they can be installed) on many of the older PCs that folks would want to give away. Also, many of the distros, large and small, are challenged with supporting the huge variety of devices in x86 PCs, especially notebooks, so it may not always be possible to advance the OSS cause this way.

Third, a mixed success story for re-use (PCs to needy kids, but all Windows) is happening in the Raleigh-Durham-Chapel Hill "Research Triangle" of North Carolina, a huge techie area. The Kramden Institute (www.kramden.org) is a nonprofit that has collected and refurbished more than 3,000 PCs that it has given to needy middle- and high-school students in the area. I think it has gone as far afield as military families at nearby Fort Bragg (yes, many of them are needy, unfortunately). Aside from its Windows-centric bias, it is having an impact with PC re-use. It seems that M\$ offers the Kramden Institute sweetheart bulk-OS licensing deals that have gone from Win 2000 to XP, last I knew (I do not know if this has "progressed" to Vista, but I see now the Kramden folks want at least a 700MHz CPU for donated systems vs. the original 300 or so). Despite the inclusion of OpenOffice.org in the PC build, my efforts a few times to interest them in using something like Puppy Linux for lower-end PCs, otherwise inadequate for Windows, has not gotten anywhere.

--
RO

You make a good point with Craigslist—I just mentioned the newspaper, because in the area where I live, most folks who would be looking for giveaway computers still don't have computers at all, much less Internet access. As to what distro to install, there are arguments for common distros vs. more efficient smaller distros—ultimately, the right answer will vary from instance to instance.

As far as Windows installation on giveaway computers, it's true Microsoft is making some incredible deals on bulk nonprofit purchases. My suspicion is Microsoft is concerned that if people's

[LETTERS]

first experience with computers is with Linux, they'll have little motivation later to switch to Windows—a "get 'em hooked early" type of thing. We just need to keep doing what we can and not become discouraged. Thanks for your letter.—Ed.

Squid Clarification

Regarding Mick Bauer's Squid series in the April, May and July 2009 issues, I set up my system (Fedora 8) to use Squid. I set up my other computer to use Firefox and set the proxy the same as Mick specified in the article. I issued the tail command and waited to see the display...nothing. After some fooling around, I discovered that the firewall on both my Windows box (ZoneAlarm) and on the Fedora box was not allowing the port to work. After setting both firewalls to allow port 3128, it worked great. I don't know if Mick was going to say anything about firewalls in the next installment, but he needs to, because it won't work without the firewall set correctly.

I also should mention that when I started Squid (Fedora 8), it complained about not having visible_hostname set in the squid.conf file. After I set it, Squid would start.

--
John Bruce

Mick Bauer replies: *You're right. I completely forgot to mention personal/local firewalls, which, as you correctly point out, need to be set up to allow access to/from TCP 3128 (or whatever port Squid is using) on the Squid server and all client systems. Regarding the visible_hostname setting, on both my Ubuntu 9.04 and 8.04 systems, this option is not set at all, yet I've had no problems. Either Squid is figuring out its hostname on its own via the local DNS resolver, or Ubuntu's version of Squid (2.7) behaves differently from Fedora's (Squid 3.0, since it's Fedora 9). Either way, I'm sorry for the omission. I try to make my tutorials as comprehensive as possible!*

PHOTO OF THE MONTH

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



I am sending this picture of myself wearing my GNU/Linux T-shirt as a way to say a big thank you for the free *Linux Journal* digital subscription I won a couple of months ago. I have eleven years' experience in ICT, and, as an experienced Linux user, I wanted to tell you that your magazine is, in a word, magnificent. This picture was taken on April 13, 2009, in Valbrevenna, Genoa, Italy. GPS coordinates are 44.5516 N and 9.0793 E. It is a beautiful place not far from Genoa, where I live.—N. Gianluca Falco.

LINUX JOURNAL

At Your Service

MAGAZINE

PRINT SUBSCRIPTIONS: Renewing your subscription, changing your address, paying your invoice, viewing your account details or other subscription inquiries can instantly be done on-line, www.linuxjournal.com/subs. Alternatively, within the U.S. and Canada, you may call us toll-free 1-888-66-LINUX (54689), or internationally +1-818-487-2089. E-mail us at subs@linuxjournal.com or reach us via postal mail, Linux Journal, PO Box 16476, North Hollywood, CA 91615-9911 USA. Please remember to include your complete name and address when contacting us.

DIGITAL SUBSCRIPTIONS: Digital subscriptions of *Linux Journal* are now available and delivered as PDFs anywhere in the world for one low cost. Visit www.linuxjournal.com/digital for more information or use the contact information above for any digital magazine customer service inquiries.

LETTERS TO THE EDITOR: We welcome your letters and encourage you to submit them at www.linuxjournal.com/contact or mail them to Linux Journal, PO Box 980985, Houston, TX 77098 USA. Letters may be edited for space and clarity.

WRITING FOR US: We always are looking for contributed articles, tutorials and real-world stories for the magazine. An author's guide, a list of topics and due dates can be found on-line, www.linuxjournal.com/author.

ADVERTISING: *Linux Journal* is a great resource for readers and advertisers alike. Request a media kit, view our current editorial calendar and advertising due dates, or learn more about other advertising and marketing opportunities by visiting us on-line, www.linuxjournal.com/advertising. Contact us directly for further information, ads@linuxjournal.com or +1 713-344-1956 ext. 2.

ON-LINE

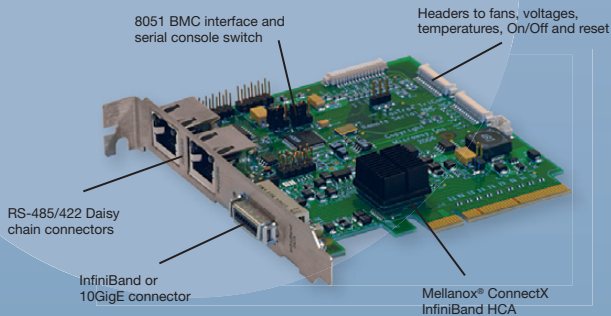
WEB SITE: Read exclusive on-line-only content on *Linux Journal's* Web site, www.linuxjournal.com. Also, select articles from the print magazine are available on-line. Magazine subscribers, digital or print, receive full access to issue archives; please contact Customer Service for further information, subs@linuxjournal.com.

FREE e-NEWSLETTERS: Each week, *Linux Journal* editors will tell you what's hot in the world of Linux. Receive late-breaking news, technical tips and tricks, and links to in-depth stories featured on www.linuxjournal.com. Subscribe for free today, www.linuxjournal.com/enewsletters.

Your Applications Will Run Faster With Next Generation Microway Solutions!

TriCom™ X

- QDR/DDR InfiniBand HCA
- ConnectX™ Technology
- 1µsec Latency
- Switchless Serial Console
- NodeWatch™ Remote Management



Teraflop GPU Computing

For Workstations and HPC Clusters

- NVIDIA® Tesla™ GPU with 240 Cores on One Chip
 - CUDA™ SDK
- NVIDIA® Quadro® Professional Graphics
- AMD® FireStream™ GPU
 - Stream SDK with Brook+



FasTree™ X

- Mellanox® InfiniScale™ IV Technology
- QDR/DDR InfiniBand Switches
- Modular Design
- 4 GB/sec Bandwidth per Port
- QSFP Interconnects
- InfiniScope™ Real Time Diagnostics



NumberSmasher®

Large Memory Scalable SMP Server

- Scales to 1 TB of Virtual Shared Memory
- Up to 128 CPU Cores
- 8U System Includes 32 Quad Core CPUs
- QDR 1 µsec Backplane

Call the HPC Experts at Microway to Design Your Next
High-Reliability Linux Cluster or InfiniBand Fabric.

508-746-7341

Sign up for Microway's
Newsletter at
www.microway.com

 **Microway**
Technology you can count onSM

diff -u

WHAT'S NEW IN KERNEL DEVELOPMENT

Some interesting **intellectual property** issues surfaced recently. With the settlement between **Microsoft** and **TomTom**, **Martin Steigerwald** thought it would be a good idea to replace **VFAT** as removable media's standard cross-platform filesystem under Linux. No sense risking a lawsuit from Microsoft, he felt. But, although there was some support for the idea, there doesn't seem to be an easy way to address VFAT's ubiquity. Most filesystem projects would like to gain as many users as possible, so starting a filesystem project whose goal is to migrate all of VFAT's users to it would be nothing new. Still, Martin's idea may find some adherents. **Mark Williamson** suggested as an alternative the possibility of a filesystem over **USB**, which would let each device use whatever on-disk filesystem it chose, while providing seamless communication between them, at least in theory.

It's not clear what, if anything, will come out of that debate; the modern patent system is truly broken. But Microsoft apparently has not bothered about VFAT in Linux yet, so maybe the kernel folks will decide to wait until something more threatening happens.

Meanwhile, **Steven Rostedt** has applied for a software patent, specifically so as to include his code in the kernel without fear that some other company might try to patent the algorithm and use it against the Open Source community. As it turns out, **Andi Kleen** thought at first that Steven had the standard evil motive and gave him a pretty good dressing down, saying that the kernel really should avoid using patented algorithms anywhere in the kernel. But, Steven explained the situation, and **Alan Cox** also encouraged people to patent their algorithms and release them for

free to the Free Software world.

After staying with **quilt** for quite a while, **Bartlomiej Zolnierkiewicz** finally moved the IDE subsystem to a **git** repository. Apparently, the various IDE developers had been making noise about preferring git, and he finally gave in. git has been making steady inroads, although it's hard to make a real estimate of how many people in the world really use it. Commercial version control software has a number of features that git doesn't yet have, which seem to be high on the list of priorities for companies considering switching to git. One feature is the ability to check out only a portion of the repository. It's one of git's great powers—letting developers check out the entire tree and use version control on their local copy, but for some very large projects, that might not be economical or secure. Regardless, git works quite well for kernel development, and at least in the near term, it's unlikely to develop many complex features that don't directly relate to kernel development.

Greg Banks from **SGI** announced that some of its filesystem software would be coming out under the **GPL version 2**. SGI folks apparently had just taken a batch of code that seemed to have some useful parts and decided to quit their own work on it and just give it away to open-source people who could benefit from it. Some of the code was obviously useful, such as tools designed to put heavy loads on a given filesystem and detect certain types of corruption, but other code relies too much on internal SGI infrastructure and would have to be purged of all that before it really could be useful. In any case, Greg made it clear that SGI would not be supporting any of the code.

—ZACK BROWN

Kindle DX—So Big You'll Want to Fold It

The Kindle 2 was Amazon's answer to concerns over the shortcomings of its immensely popular ebook reader. Version 2 improved on the form factor, usability and many other issues that annoyed users—everything except the screen size.

That's where the new model, the DX, comes into play. This Kindle boasts an 8.5 x 11" screen, which is conveniently the same size as standard US letter paper. The screen rotates, is rumored to have better contrast and, not to beat a dead horse, it's huge! Due to the size of the Kindle DX, sideways scrolling for things like PDF documents will be a thing of the past.

On the downside, the Kindle DX costs almost \$500 and was released right on the heels of the Kindle 2. Those folks who just shelled out \$359 for their shiny (almost) new book reader will be hard-pressed to spend anything, much less \$489, on a device that adds only a larger screen that rotates and displays PDFs. But, if you were holding out because you thought the screen was too small, the Kindle DX might be just the ticket for you. Besides, don't we all need another Linux device in our lives?

—SHAWN POWERS



NON-LINUX FOSS

FreeRTOS.org uIP WEB server demo - Microsoft Internet Explorer

Address: http://172.25.218.9/index.shtml

RTOS Stats | TCP Stats | Connections | FreeRTOS.org Homepage | IO

Task statistics

Page will refresh every 2 seconds.

Task	State	Priority	Stack	#
uIP	R	2	150	0
QConSB6	R	0	84	6
PolSEM1	R	0	02	12
PolSEM2	R	0	82	13
IntMath	R	0	95	18
IDLE	R	0	98	22
QProdB2	R	0	84	7
QProdB3	R	0	03	3
QProdB5	R	0	83	5
LEDx	B	2	92	9
QProdNB	B	2	91	17
RTest2	B	2	87	8
QConSB	B	2	90	16
LEDx	B	2	92	10
LEDx	D	2	92	11
CREATOR	B	3	115	21
BlkSEM1	B	1	82	14
BlkSEM2	B	1	82	15
QConSB1	H	2	84	1
QConSB4	D	2	04	4
Check	B	3	81	19
BTest1	B	3	83	7
LCD	S	2	82	20

Refresh count = 149

FreeRTOS Task Stats (from www.freertos.org)

FreeRTOS is an open-source mini-real-time kernel with preemptive multitasking and coroutines. It provides queues for intertask communication, and binary semaphores, counting semaphores, recursive semaphores, mutexes and mutexes with priority inheritance for task synchronization. It can run with as little as 1K bytes of RAM and 4K bytes of ROM.

FreeRTOS has been ported to 19+ architectures that include 8-, 16- and 32-bit processors. The core code is mostly written in C and is compatible with most C compilers. The distribution comes with numerous samples for a number of different development boards. Many of the samples include an embedded Web server.

—MITCH FRAZIER

Forecast: Cloudy

In the November 2008 UpFront section, we published a little piece describing what “Cloud Computing” really meant. I’m a big fan of cool buzzwords, but for some reason, “The Cloud” is a term that always has seemed unnerving. Don’t get me wrong; the concept is great. It’s just that so many companies seem to be touting their new “cloud solution” to entice people with their hipness.

As much as I’d love to see the term change, I think it’s here to stay. And, thanks to Linux, it’s going to be quite a contender for replacing traditional server farms. Companies like Amazon with its EC2 or Google with its massive numbers of abstracted computer clusters are proving that Linux is the perfect way to throw serious computing muscle into large-scale clouds.

Once more, Linux is taking a huge market share and getting little praise for it. Linux is the premier choice for cloud computing providers, because it’s affordable, scalable and, quite frankly, cheap. Now if only we can get a different name to stick—maybe S.H.A.W.N. (Superior Horsepower Abstracted Windows-lessly in a Network) computing? Yeah, probably not.

—SHAWN POWERS

LJ Index August 2009

1. Millions of Netbooks shipped in first quarter 2009: **4.5**
2. Percent increase in Netbook shipments from first quarter 2008: **700**
3. Netbook shipments as a percent of all PC shipments in first quarter 2009: **8**
4. Billions of PC systems shipped to date: **2.9**
5. Billions of ARM processors shipped to date: **10**
6. Median hourly wage (US/all occupations): **\$15.57**
7. Median hourly wage (US/computer and mathematical science occupations): **\$34.26**
8. Median hourly wage (US/health-care practitioner and technical occupations): **\$27.20**
9. Median hourly wage (US/farming, fishing and forestry occupations): **\$9.34**
10. Median hourly wage (US/food preparation and serving-related occupations): **\$8.59**
11. Number of language front ends supported by GCC: **15**
12. Number of back ends (processor architectures) supported by GCC: **53**
13. Worldwide number of official government open-source policy initiatives: **275**
14. Millions of active .com domain registrations: **80.5**
15. Millions of active .org domain registrations: **12.2**
16. Millions of active .net domain registrations: **7.6**
17. Millions of active .info domain registrations: **5.1**
18. Millions of inactive (deleted) .com domain registrations: **301.2**
19. US National Debt as of 05/03/09, 7:36:12pm MST: **\$11,250,870,541,216.70**
20. Change in the debt since last month's column: **\$115,410,006,992.80**

Sources: 1–3: IDC / 4: Metrics 2.0 / 5: ARM Holdings
6–10: BLS (Bureau of Labor Statistics) / 11, 12: Wikipedia
13–18: Domain Tools / 19: www.brillig.com/debt_clock
20: Math

at, batch and cron—the ABCs of Doing Work When Nobody’s Home

People always have been interested in doing more work with less effort. This drive reaches its peak when work is being done, even though you aren’t actually doing anything. With Linux, you effectively can do this with the trio of programs `at`, `batch` and `cron`. So, now your computer can be busy getting productive work done, long after you’ve gone home. Most people have heard of `cron`. Fewer people have heard of `at`, and even fewer have heard of `batch`. Here, you’ll find out what they can do for you, and the most common options for getting the most out of them.

`at` is actually a collection of utilities. The basic idea is that you can create queues of jobs to run on your machine at specified times. The time `at` runs your job is specified on the command line, and almost every time format known to man is accepted. The usual formats, like `HH:MM` or `MM/DD/YY`, are supported. The standard POSIX time format of `[[CC]YY]MMDDhhmm[.SS]` is also supported. You even can use words for special times, such as `now`, `noon`, `midnight`, `teatime`, `today` or `tomorrow`, among others. You also can do relative dates and times. For example, you could tell `at` to run your job at 7pm three days from now by using `7PM + 3 days`.

`at` listens to the standard input for the commands to run, which you finish off with a `Ctrl-D`. You also can place all of the commands to run in a text file and tell `at` where to find it by using the command-line option `-f filename`. `at` uses the current directory at the point of invocation as the working directory.

By default, `at` dumps all of your jobs into one queue named `a`. But, you don’t need to stay in that one little bucket. You can group your jobs into a number of queues quite easily. All you need to do is add the option `-q x` to the `at` command, where `x` is a letter. This means you can group your jobs into 52 queues (`a–z` and `A–Z`). This lets you use some organization in managing all your after-hours work. Queues with higher letters will run with a higher niceness. The special queue, `=`, is reserved for jobs currently running.

So, once you’ve submitted a bunch of jobs, how do you manage them? The command `atq` prints the list of your upcoming jobs. The output of the list is job ID, date, hour, queue and user name. If you’ve broken up your jobs into multiple queues, you can get the list of each queue individually by using the option `-q x` again. If you change your mind, you can delete a job from the queue by using the command `atrm x`, where `x` is the job ID.

Now, what happens if you don’t want to overload your box? Using `at`, your scheduled job will run at the assigned time, regardless of what else may be happening. Ideally, you would want your scheduled jobs to run only when they won’t interfere with other work. This is where the `batch` command comes in. `batch` behaves the same way `at` does, but it will run the job only once the system load drops below a certain value (usually 1.5). You can change this value when `atd` starts up. By using the command-line option `-l xx`, you can tell `batch` not

to run unless the load is below the value `xx`. Also, `batch` defaults to putting your jobs into the queue `b`.

These tools are great for single runs of jobs, but what happens if you have a recurring job that needs to run on some sort of schedule? This is where our last command, `cron`, comes in. As a user, you actually don’t run `cron`. You instead run the command `crontab`, which lets you edit the list of jobs that `cron` will run for you. Your `crontab` entries are lines containing a time specification and a command to execute. For example, you might have a backup program running at 1am each day:

```
0 1 * * * backup_prog
```

`cron` accepts a wide variety of time specifications. The fields available for your `crontab` entries include:

- minute: 0–59
- hour: 0–23
- day of month: 1–31
- month: 1–12
- day of week: 0–7

You can use these fields and values directly, use groups of values separated by commas, use ranges of values or use an asterisk to represent any value. You also can use special values:

- `@reboot`: run once, at startup
- `@yearly`: run once a year (0 0 1 1 *)
- `@annually`: same as `@yearly`
- `@monthly`: run once a month (0 0 1 * *)
- `@weekly`: run once a week (0 0 * * 0)
- `@daily`: run once a day (0 0 * * *)
- `@midnight`: same as `@daily`
- `@hourly`: run once an hour (0 * * * *)

Now that you have these three utilities under your belt, you can schedule those backups to run automatically, or start a long compile after you’ve gone home, or make your machine use up any idle cycles. So, go out and get lots of work done, even when nobody is home.

—JOEY BERNARD

Most Popular Articles on LinuxJournal.com

Here are the all-time most-popular articles on LinuxJournal.com. Have you read them yet?

1. "Why Python?" by Eric Raymond: www.linuxjournal.com/article/3882
2. "Boot with GRUB" by Wayne Marshall: www.linuxjournal.com/article/4622
3. "Building a Call Center with LTSP and Soft Phones" by Michael George: www.linuxjournal.com/article/8165
4. "GNU/Linux DVD Player Review" by Jon Kent: www.linuxjournal.com/article/5644
5. "Python Programming for Beginners" by Jacek Artymiak: www.linuxjournal.com/article/3946
6. "Chapter 16: Ubuntu and Your iPod" by Rickford Grant (excerpt from *Ubuntu Linux for Non-Geeks: A Pain-Free, Project-Based, Get-Things-Done Guidebook*): www.linuxjournal.com/article/9266
7. "Monitoring Hard Disks with SMART" by Bruce Allen: www.linuxjournal.com/article/6983
8. The Ultimate Distro by Glyn Moody: www.linuxjournal.com/node/1000150
9. "Streaming MPEG-4 with Linux" by Donald Szeto: www.linuxjournal.com/article/6720
10. "The Ultimate Linux/Windows System" by Kevin Farnham: www.linuxjournal.com/article/8761

—KATHERINE DRUCKMAN

They Said It

Computer Science is no more about computers than astronomy is about telescopes.

—E. W. Dijkstra

Java is the single most important software asset we have ever acquired.

—Larry Ellison

I think there is a world market for maybe five computers.

—IBM Chairman Thomas Watson, 1943

The world potential market for copying machines is 5,000 at most.

—IBM to the founders of Xerox, 1959

Two years from now, spam will be solved.

—Bill Gates, World Economic Forum 2004

The nice thing about standards is that there are so many to choose from.

—Andrew S. Tannenbaum

Why is it drug addicts and computer aficionados are both called users?

—Clifford Stoll

The Rise of the Unconvention

This past May, I had the privilege of attending Penguicon 7.0 in Detroit, Michigan, as a "Nifty Guest". It's not a big convention, with a little more than 1,000 attendees, and it's not even solely devoted to Linux. But, that seems to be part of the magic. Although at times it's a bit unnerving to see people walking around dressed in furry suits (Penguicon is also a science-fiction convention you see), the mix of people with varying levels of technical expertise is actually quite refreshing.

It's not terribly often a panel on Linux adoption on the desktop can be attended by both those familiar with using Linux and those just curious about it. The ability to bring those folks together so each can understand the other's viewpoint is truly invaluable. Add to that some of the geekiest guests, craziest entertainment and all-you-can-drink Monster energy drinks—Penguicon really sets itself apart as a conference that will bring you back year after year. As the entry fee is less than \$50, making that yearly trek isn't even terribly painful on the pocketbook.

Are the smaller Linux conventions going to take over and grow while the larger ones start dwindling? I won't say it's going to happen, but I certainly wouldn't be surprised. As for myself, I'm going to try getting to the Ohio LinuxFest. If you look around, you'll probably find one close to you too. And, if not, consider starting one yourself!

—SHAWN POWERS



Associate Editor Shawn Powers at Penguicon



REUVEN M. LERNER

Fixtures and Factories

Use factories and fixtures in your Rails applications to help simplify your database-related testing.

One of the points of pride in the Ruby community is the degree to which developers are focused on testing. As I wrote last month, tests in a dynamic language have more potential to correct more errors and keep your code trim and functional than even the best compilers. Rails developers are used to working with three different types of tests: unit (for database models), functional (for controller classes) and integration (for testing things from a user's perspective). Combined with coverage and analysis tools, such as the `metric_fu` gem I described last month, these tests can help ensure that your code is as solid as possible before it is seen by the general public.

Testing your code requires that you provide it with inputs and that you then match those inputs with expected outputs. When it comes to a Web application, those inputs most likely will come from either a relational database or from a user's form

Fixtures are nice, but as a number of Rails developers have written over the years, they can be hard to write, hard to keep track of and generally brittle.

submission. Testing form submissions is not particularly difficult, especially in a framework such as Rails, which has extensive testing support built in. Testing data that comes from a database, however, can be a bit more challenging, because it means that you must somehow store the data in the database so that the tests can access it.

One possible solution, of course, is to pre-populate the database tables with test data directly. But, as simple and obvious as that solution might appear at first glance, it assumes that you have a source from which you can pre-populate the database. You could do it by hand, but then you'll find that any modifications your program makes to the database—creating, updating and deleting rows—either will stay in effect for the next test or will need to be reloaded from scratch from another source.

In other words, you need a way to put the test database into a known state before you begin your tests. If you know this beginning state, you can write

tests that check subsequent states.

The question is, how do you create that initial state? From the time that Rails was first released, the answer was fixtures—text files containing YAML-formatted hand-crafted data. Fixtures are nice, but as a number of Rails developers have written over the years, they can be hard to write, hard to keep track of and generally brittle.

This month, I take a look at the current state of loading data into a test database. I start by examining fixtures, exploring some ways you still might be able to make them useful inside your tests. Then, I cover a newer approach to test data, known as factories, looking at the `Factory Girl` gem and then taking a quick peek at the `Machinist` gem, both of which are in widespread use among Rails developers and might be a better fit than plain-old fixtures for your project.

Creating Your Application

Fixtures, as I mentioned above, are YAML files containing data that can be loaded into a database. Rails actually allows you to put your fixture data in formats other than YAML, such as CSV. However, my guess is that CSV is mostly unused, and that YAML is the format used by almost everyone working with fixtures.

I created a simple Rails application (using SQLite) on my computer with:

```
rails --database=sqlite3 appointments
```

Then, I generated a RESTful resource for people:

```
./script/generate scaffold person \
  first_name:string last_name:string email:string
```

This not only created a model for working with people, but also a controller for handling the basic RESTful functions, views for all of those controller actions, a database migration that uses Ruby to describe my model and even some rudimentary tests. I can import the database migrations with:

```
rake db:migrate
```

And, voilà! I now have a working application

that allows me to add, delete, modify and list a bunch of people. You might have noticed that I named my Rails application appointments. My plan is to create a very simple appointment calendar, so that I can keep track of with whom I'll be meeting. So, I create another resource, named meetings:

```
./script/generate scaffold meeting \  
  starting_at:timestamp ending_at:timestamp location:text
```

(It should go without saying that if I were creating this for real, I would not store the location as a text field, but rather as an ID pointing to another table of locations. Keeping data in such normalized form, so that the text appears in a single place and is referred to from elsewhere in the database using foreign keys, makes the application more robust, as well as more efficient.)

Finally, I create a third table, meeting_person, which allows one or more people to have a meeting. If I were willing to restrict appointments to a single participant (or two participants, if I include the person using this software), I simply could have a person_id field in the meeting table. To get this, I create a new model:

```
./script/generate model meeting_person \  
  person_id:integer meeting_id:integer
```

Now that the three models are in place, I can add associations—those declarations in the model classes that link them to one another. While I'm editing the model, I also will add some validations, which ensure that the data fits my standards. The final version of the models is shown in Listing 1. Perhaps the only particularly interesting part of the models is the custom validation that I placed in the Meeting model:

```
def validate  
  if starting_at > ending_at  
    errors.add_to_base("Starting time is later than ending time!")  
  end  
end
```

I also created a convenience function that returns an array of names with whom the appointment will be:

```
def people_as_sentence  
  return self.people.map {|p| p.fullname}.to_sentence  
end
```

This validation, which is run whenever I try to save an instance of Meeting, checks to make sure that the starting time is earlier than the ending time. If this is not the case, the validation fails, and

Listing 1. Model Files, with Associations and Validations

```
class Person < ActiveRecord::Base  
  has_many :meeting_people  
  has_many :meetings, :through => :meeting_people  
  
  validates_presence_of :first_name, :last_name, :email  
  validates_uniqueness_of :email  
  
  def fullname  
    "#{first_name} #{last_name}"  
  end  
end  
  
class Meeting < ActiveRecord::Base  
  has_many :meeting_people  
  has_many :people, :through => :meeting_people  
  
  validates_presence_of :starting_at, :ending_at, :location  
  
  def validate  
    if starting_at > ending_at  
      errors.add_to_base("Starting time is later than ending time!")  
    end  
  
    if self.people.empty?  
      errors.add_to_base("You must meet with at least one person!")  
    end  
  end  
  
  def people_as_sentence  
    return self.people.map { |p| p.fullname}.to_sentence  
  end  
end  
  
class MeetingPerson < ActiveRecord::Base  
  belongs_to :person  
  belongs_to :meeting  
  
end
```

the data is not stored. (The fact that I can treat times as full-fledged objects, with access to the > and < operators, is one of my favorite parts of both Ruby and SQL.)

Finally, I'm going to enhance this application by modifying the existing scaffolded controller actions to be more useful. First, I modify the new and create actions, such that they will allow someone to create an appointment, simultaneously indicating the person or people with whom the appointment will take place. Then, I modify the index action,

Listing 2. views/meetings/new.html.erb, Modified from the Default Scaffold to Allow the User to Enter One or More People

```
<h1>New meeting</h1>

<% form_for(@meeting) do |f| %>
  <%= f.error_messages %>

  <p>
    <%= f.label :starting_at %><br />
    <%= f.datetime_select :starting_at %>
  </p>
  <p>
    <%= f.label :ending_at %><br />
    <%= f.datetime_select :ending_at %>
  </p>
  <p>
    <%= f.label :location %><br />
    <%= f.text_area :location %>
  </p>

  <p>With:
    <%= select("person",
              "person_id",
              Person.all.collect { |p| [p.fullname, p.id] },
              {}),
              {:multiple => true}) %>
  </p>
  <p>
    <%= f.submit 'Create' %>
  </p>

<% end %>

<%= link_to 'Back', meetings_path %>
```

so that the user will get a list of all upcoming appointments.

Fixtures

Now that I've created a simple application, the time has come to test it. As I wrote above, testing the application requires that I have some sample data with which to test it. By default, the generators for Rails models create basic fixtures, which have long been the standard way to import data into Rails tests. By basic, I mean that they contain some very, very basic data—too basic, actually, for any real testing I might want to do. For example, here is the automatically generated fixture for people:

```
one:
  first_name: MyString
  last_name: MyString
  email: MyString
```

```
two:
  first_name: MyString
  last_name: MyString
  email: MyString
```

Even if you are new to reading YAML, let alone fixture files, the format should be easy enough to understand. YAML consists of name-value pairs within a hierarchy, and indentation indicates where in the hierarchy a particular name-value pair exists. (You also can associate a list of values with the key, by separating values with commas.) Thus, there are two people defined in the fixture, one and two, and each has three name-value pairs.

However, these name-value pairs are close to useless. They might contain valid data, or they might contain data that fails to adhere to the standards laid out in my model validations. If I had defined a validator for the email field, ensuring that the field always would contain a valid e-mail address, the tests would fail right away, before they even ran. Rails would load the fixtures into ActiveRecord, the database would reject them as being invalid and I'd be left scratching my head.

Things get even hairier when you start to make fixtures that depend on associations. I obviously want my meeting_people fixtures to point to valid people and meetings, but using the numeric IDs can get confusing very quickly. Fortunately, recent versions of Rails allow me to name the fixture to which an object is associated, rather than its numeric ID. Thus, although the default fixtures for meeting_people is this:

```
one:
  person_id: 1
  meeting_id: 1

two:
  person_id: 1
  meeting_id: 1
```

instead, I can say this:

```
one:
  person: one
  meeting: one

two:
  person: two
  meeting: two
```

Obviously, you would want to choose more descriptive names for your fixtures. But, I now have indicated that meeting #1 is with person #1, and meeting #2 is with person #2. This is obviously

more descriptive than the simple numbers would be.

You can even do one better than this, because fixtures understand the `has_many :through` associations that I defined in the models. Just as in the Ruby code, I can add a person to a meeting with:

```
meeting.people << a_person
```

I can put the same sorts of information in the fixture file. For example:

```
one:
  starting_at: 2009-05-10 00:48:12
  ending_at: 2009-05-10 01:48:12
  location: MyText
  people: one, two
two:
  starting_at: 2009-05-10 00:48:12
  ending_at: 2009-05-10 01:48:12
  location: MyText
  people: two
```

If you do things this way, you don't want to define things in both the `meeting_people` fixture and in the `meetings` fixture. Otherwise, you might be in for some very strange errors. Note that fixture files are ERb (embedded Ruby) files, so you can have dynamically generated entries, such as:

```
one:
  starting_at: <%= 5.minutes.ago %>
  ending_at: <%= Time.now %>
  location: MyText
  people: one, two
```

Now, how do you use these fixtures in your tests? It's actually pretty straightforward. You need to load the fixtures you want with the `fixtures` method:

```
fixtures :meetings
```

By default, all fixtures are imported, thanks to:

```
fixtures :all
```

in `test/test_helper.rb`, which is imported automatically into all tests. Then, in your test, you can say something like this:

```
get :edit, :id => people(:one).id
```

This example (of a functional test) will load the person object identified as `one` in `people.yml`, invoking the `edit` method and passing it the ID of the appropriate fixture.

Factory Girl

For a small site, or when you can keep everything in your head, fixtures are just fine. I've certainly used them over the years, and I've found them to be an invaluable part of my testing strategy. But, factories are an alternative to fixtures that have become increasingly popular, both because they're written in Ruby code, and they allow you to do all sorts of things that are difficult or impossible with YAML fixtures.

Factory Girl is one of the best known factories, written and distributed by the Thoughtbot company, and it is available as a Ruby gem. After installing Factory Girl on your system and bringing it into your application's environment with:

```
config.gem "thoughtbot-factory_girl",
  :lib => "factory_girl",
  :source => "http://gems.github.com"
```

in `config/environment.rb`, you will be able to use it. Basically, Factory Girl allows you to create objects in Ruby, rather than load them from fixture files. No defaults are created for you by the generator, but that's not a big deal, given how easy it is to use Factory Girl to create test objects.

Above, I showed how in a test environment using fixtures, you can grab the person object with a name of one by using the `people` method, and then passing a symbol:

```
get :edit, :id => people(:one).id
```

`people(:one)` is a full-fledged ActiveRecord object, with everything you might expect from such an object. Factory Girl works in a different way. First, you need to create a `test/factories.rb` file, in which your factories are defined. (You also may create a `test/factories/` directory, the contents of which will be Ruby files defining factories.)

To create a factory for people (that is, in place of `people.yml`), insert `people.rb` inside `test/factories`:

```
Factory.define :person do |p|
  p.first_name 'Reuven'
  p.last_name 'Lerner'
  p.email 'reuven@lerner.co.il'
end
```

Now, inside the tests, you can say:

```
get :edit, :id => Factory.build(:person).id
```

or:

```
person = Factory.build(:person)
get :edit, :id => person.id
```

At first glance, this doesn't seem too exciting. After all, you could have done roughly the same thing with your fixture, right? But factories allow you to override the defaults:

```
person = Factory.build(:person, :first_name => 'Foobar')
get :edit, :id => person.id
```

But wait, there's more. You can set associations as follows:

```
Factory.define :person do |p|
  p.first_name 'Reuven'
  p.last_name 'Lerner'
  p.email 'reuven@lerner.co.il'
  p.meetings {|meetings| meetings.association(:meeting)}
end
```

In other words, if you have created a meeting factory, you can incorporate it into your person factory, taking advantage of the association, using a fairly natural syntax.

An even more interesting idea is that of sequences. If your application needs to create a large number of test people, you might want each of those people to have a unique e-mail address. (Never mind that the e-mail never will be sent.) You can do this with a sequence:

```
Factory.define :person do |p|
  p.first_name 'Reuven'
  p.last_name 'Lerner'
  p.sequence(:email) {|n| "person#{n}@example.com" }
end
```

The first person created with this factory will have an e-mail address of person1@example.com; the second will be person2@example.com and so forth.

As you can see, Factory Girl is as easy to use as YAML fixtures, but it offers a great many capabilities that come in handy when testing Rails applications.

Factory Girl is a terrific library for factories, and it has become quite popular since it was first released. But, not everyone liked its basic syntax, and one of those people was Pete Yandell, who decided that although the basic idea behind factories was sound, he wanted to use a different (and more compact) syntax for his factories. Thus was born Machinist, which uses a Sham object to describe fields in an object, which are then assembled into blueprints for specific objects. For example:

```
require 'faker'

# Define the fields that we will need
Sham.first_name { Faker::Name.first_name }
```

```
Sham.last_name { Faker::Name.last_name }
Sham.email { Faker::Internet.email }
```

```
# Now use these field definitions to create a blueprint
Person.blueprint do
  first_name
  last_name
  email
end
```

Now you can use these blueprints to create test objects. For example:

```
person = Person.make()
```

As with Factory Girl, you also can override the defaults:

```
person = Person.make(:email => 'foo@example.com')
```

Conclusion

Fixtures have been a part of Rails testing practices since the beginning, and they still can be quite useful. But, if you're finding yourself frustrated by YAML files, or if you want to experiment with something that offers more flexibility and features, you might well want to try looking into factories. This month, I looked at two different libraries for creating Rails factories, both of which are in popular use and might be a good fit for your project. ■

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

Resources

The home page for Ruby on Rails is **www.rubyonrails.com**. Information about testing, including the use of fixtures, is in one of the excellent, community-written Rails guides at **guides.rubyonrails.org/testing.html**.

If you are interested in learning more about factories, a good starting point (as is often the case) is the Railscast site, with weekly screen-casts by Ryan Bates. The Railscast that talks about fixtures is at **railscasts.com/episodes/158-factories-not-fixtures**.

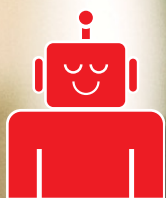
Finally, the home page for Factory Girl is at **dev.thoughtbot.com/factory_girl**, and the home page for Machinist is at **github.com/notahat/machinist/tree/master**.

Rob Purdie
Project Manager
economist.com
amnesty.org



Lullabot- Powered

The most super powered sites in the world
are created in Drupal, by you and Lullabot.



Lullabot™ New Lullabot Learning Series training DVDs at Lullabot.com



MARCEL GAGNÉ

The Case of the Missing OS

Looking around, I was confused by the scene. The operating system (and my desktop), was nowhere to be seen—and yet, there it was, everywhere I turned. It was as though the kidnappers had created countless copies of my desktop OS and left it everywhere for me to find—a mystery indeed.

What's wrong, François? You look as though your Linux desktop has been stolen. It has? *Mon ami*, I was just kidding. I can see notebooks at every table in the restaurant. It's rather strange for a thief to take things yet leave them where they are, don't you think?

Ah, I see. My apologies, *mon ami*. That's my fault. I modified the workstations so that they would boot directly into Firefox without loading a desktop. Why? Because it's the perfect way to demonstrate tonight's feature, and now that I see your reaction, I suppose it does add an air of mystery. You'll see what I am up to shortly, but for now, we must get ready. I can see our guests arriving as we speak. Smile, François. It's all good.

Good evening, *mes amis*, and welcome to *Chez Marcel*, the home of fine wine and superb Linux and open-source software. Your tables are ready, and my faithful waiter was just getting ready to fetch our featured wine. Quickly, François, head to the cellar and bring back the 2005 La Vigna Vecchia Barbera d'Asti from Italy. Check the North wing for that one.

While we wait for François and the wine, allow me to explain the nonexistent desktops before you. Each workstation has a browser running as its sole desktop application to demonstrate a very cool cloud-based open-source desktop that runs from a Linux server.

Richard M. Stallman called cloud computing a trap, saying "It's just as bad as using a proprietary program. Do your own computing on your own computer with your copy of a freedom-respecting program. If you use a proprietary program or somebody else's Web server, you're defenseless. You're putty in the hands of whoever developed that software."

Well, if you find yourself in agreement with old RMS, rest assured that there's another way to get the benefits of cloud computing without being tied to Google, Amazon or whatever other Web behemoth might be trying to get your cloud OS business. The answer is your own private cloud, courtesy of eyeOS (Figure 1). This is an impressive little desktop OS you can run from your Linux server. It comes bundled with small applications, fine-tuned for the Web, to

play music files, watch videos, surf the Web, create documents, chat, play games and a whole lot more. eyeOS is a free and open package with a commercial company behind the product as well as an active community of users and developers. It's a great way to deploy desktops in libraries, schools, for user groups and in business. It's a thin-client network without the thin-client hardware. And, it's pretty cool.

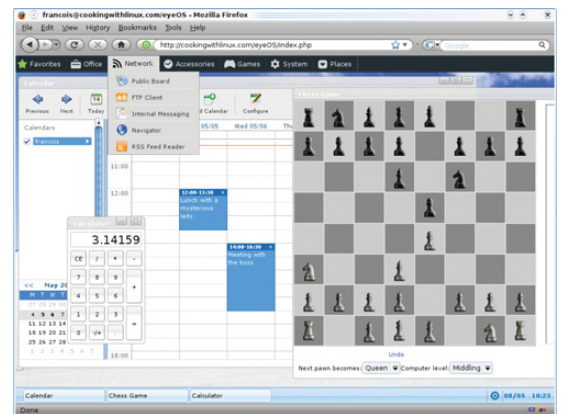


Figure 1. How would you like a screaming-fast desktop, in your own private cloud, accessible from any Web browser? Meet eyeOS.

Ah, François, good to have you back. Please pour for our guests. You'll love this one, *mes amis*. It's deep and complex with a plum and cherry flavor and a little hint of peppery spiciness. Enjoy.

Getting eyeOS served up on your, ahem, server, is easy—frightfully easy in fact. Your Linux server should be up to date and running PHP5 in order to use the latest eyeOS, but there isn't much more to it than that. Download the latest copy of eyeOS, and find a place for it inside your Web server's document root. Now, using your Apache user, extract it with this command:

```
tar -xzf eyeOS_1.8.5.0-3.tar.gz
```

The package extracts into a directory called eyeOS, but you can, of course, change that if you like. If you extract it using the root user, you need to do a global group and ownership change on that directory so Apache can write to it. You can make eyeOS its own Web site or simply place it in an existing Web site tree. Assuming you put eyeOS in the root of an existing Web site, you might connect using your browser like this: <http://yoursite.dom/eyeOS/index.html>. Because this is your first time accessing eyeOS, you'll be redirected to the installer (Figure 2).

Notice that the installer is asking you for the root password. That's the eyeOS root password—the one that will serve as the master account for this installation of eyeOS, not the one for your server. Make sure you select a password for eyeOS specifically. Before you click the Install eyeOS button, look at the check box directly above. It's labeled Allow users to create accounts. If you check this, anyone can freely create an account on the system, without root approval (eyeOS root, that is).

Once you click the Install button, everything happens very quickly, and you find yourself at the login screen. Congratulations. That's really all there

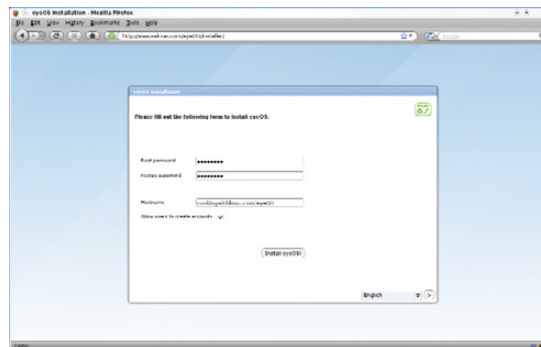


Figure 2. Installing eyeOS is a very simple process and doesn't require much typing.

is to installing eyeOS. You could log in as root here, but let me show you what happens if you allowed registration and chose to create a user. Directly below the user name and password information on the login screen is a New User button (Figure 3). Click the New User button, and the login screen displays an extended login form where you define a user name and password of your choosing.

That's all there is to it. You're ready to log in using

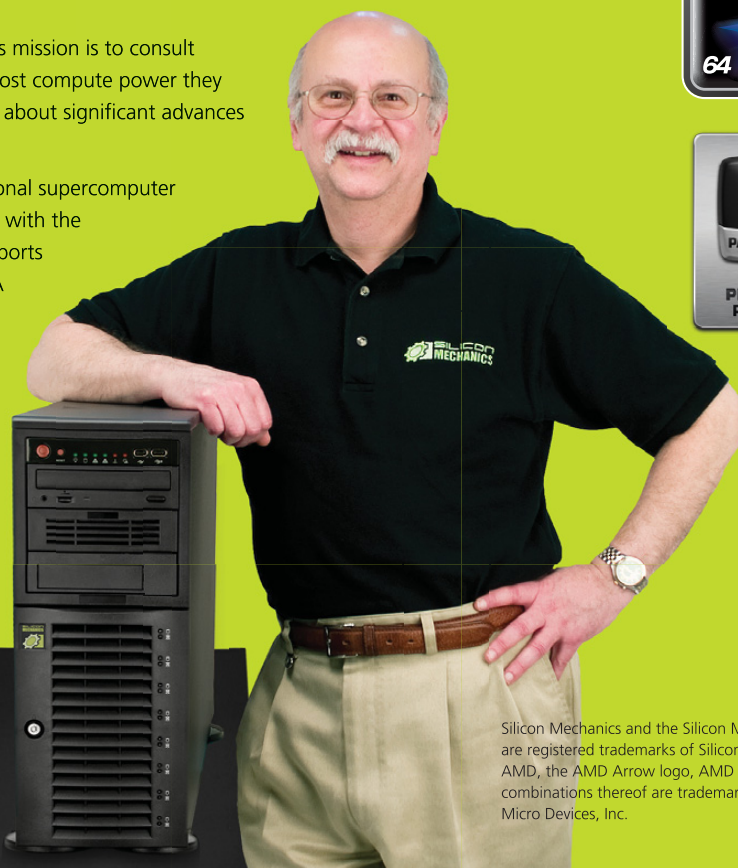
Expert included.

Art is the Silicon Mechanics education and research expert. His mission is to consult with academic and research institutions and offer them the most compute power they can get for their money. Recently he's been talking with them about significant advances in personal supercomputing.

The Hyperform HPCg A2401 from Silicon Mechanics is a personal supercomputer with NVIDIA® Tesla™ GPU technology. This workstation starts with the AMD Phenom™ X4 processor, 8GB of DDR2 RAM, and it supports up to 8 hot-swap hard drives. With the addition of the NVIDIA Tesla C1060 GPU (or two, or three), the A2401 can outperform a small cluster—and it can do it without a cluster's noise, complexity, or cooling requirements. Best of all, it can do it without a cluster's price tag: the A2401 starts at a very user-friendly \$3139.

When you partner with Silicon Mechanics, you get more than high-end compute power at astonishingly affordable prices—you get an expert like Art.

For more information about the Hyperform HPCg A2401 visit www.siliconmechanics.com/TeslaPSC.



Silicon Mechanics and the Silicon Mechanics logo are registered trademarks of Silicon Mechanics, Inc. AMD, the AMD Arrow logo, AMD Phenom, and combinations thereof are trademarks of Advanced Micro Devices, Inc.

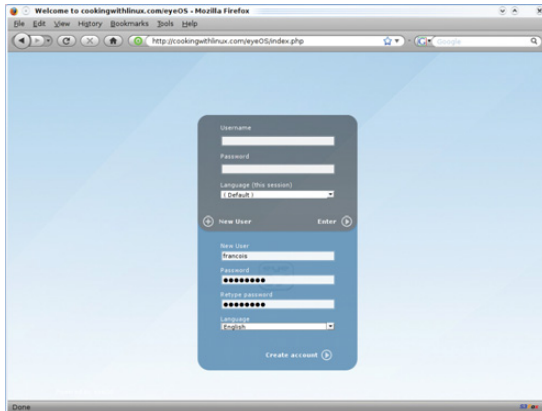


Figure 3. The login screen serves a dual purpose, allowing new users to register an account easily—if you allow it, of course.

the user name and password you've just created. Once logged in, you'll find yourself at the eyeOS desktop (Figure 4). Let's take a quick look at how this desktop is organized. Along the top is a panel referred to as the application dock. This is your gateway to an impressive collection of built-in applications (we'll explore some of those applications shortly). The desktop itself has a handful of icons to access common tools quickly, such as your calendar, the home folder and so on. To the right, you'll see a small menu floating on the desktop. Those are mini actions—functions that, although they may open an application, aren't applications per se.

Along the bottom is another panel, or bar, that shows your running applications, the date and time (along with a pop-up calendar), and a small icon that launches a system menu (Figure 5). From here, you can change your session preferences (the root user gets an enhanced Preferences dialog—more on this later), get a list of all installed applications, find out about eyeOS, launch a program (similar to the Alt-F2 quick launch in GNOME and KDE) or log out.

Before I give you a quick tour of applications,

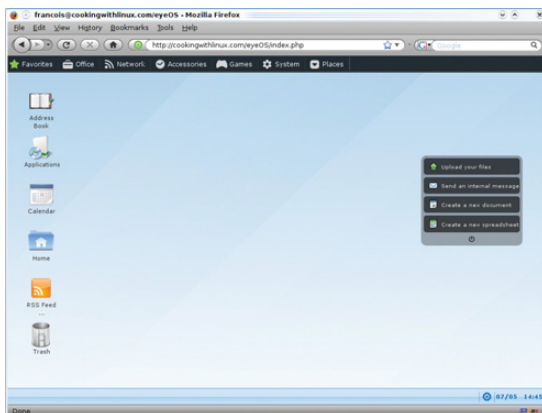


Figure 4. Your shiny new eyeOS desktop, ready to satisfy.

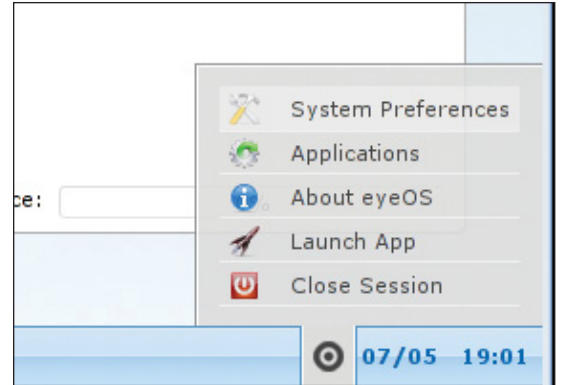


Figure 5. The system menu, the gateway to customizing your session, is right next to the system clock.

let me show you a little of the Preferences dialog (Figure 6). To change your personal information and password, this is stop one. It's also the place to change the look and feel of eyeOS. To change your wallpaper, click on Desktop, or to use a completely different theme that changes your window decorations, icons, application dock and more, click on Theme. Under System, you can change the behavior of the eyeOS board, a kind of built-in instant-messaging program that lets you communicate with other users logged in to your eyeOS cloud. Autorun commands are those that you want to run automatically when you log in. The application dock and the mini-actions are just two of the programs already in the Autorun queue. Security is interesting in that you can secure your personal session by IP address. If you want to make sure that you (or your user name) can log in only from your personal system, look here.

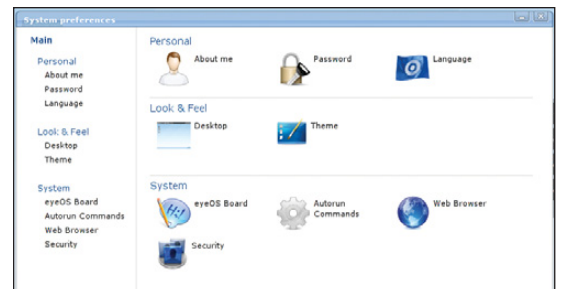


Figure 6. The System Preferences Dialog for Non-Admin Users

Let's say you do want to change that wallpaper. You need some background images first. That brings up the question on everyone's mind, which is, "How to I get files up to this thing?" There are a couple ways to do it, one being via the mini-tools menu and the other being the file manager. To upload via the mini-tools, click on Upload your files to bring up the Upload files dialog (Figure 7). Click the Add files button, then select files using your system's file selection dialog (Firefox



Linux - FreeBSD - x86 Solaris - MS etc.



uses a GTK file dialog, for instance). When you've chosen the file, or files, you want to upload, click the Upload now! button. One by one, the files will be uploaded to your eyeOS desktop.

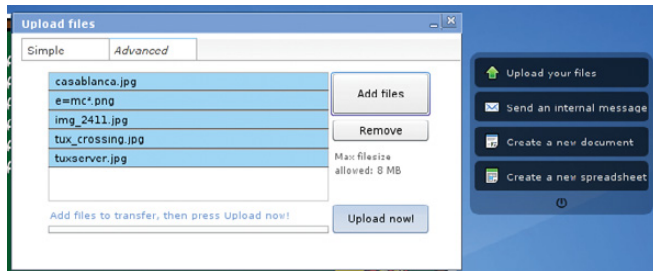


Figure 7. Uploading files is done through the file manager or the desktop mini-tools (shown here).

By default, the files will be uploaded to your desktop folder. As we all know, files and folders aren't always where we want them. Besides, from time to time, you need to do a little cleanup in your virtual home. The same holds true for the folders in your cloud-based desktop. Click the Home icon on the desktop to open the file manager (Figure 8).

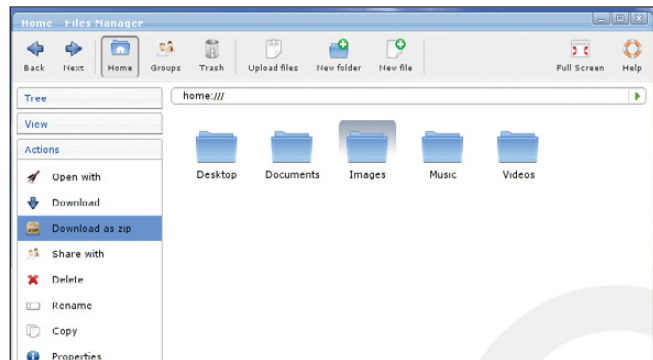


Figure 8. The eyeOS file manager lets you manipulate your data on your virtual desktop as well as compress and download to your physical desktop.

I mentioned a couple applications already, one being the eyeOS board that you can use to chat with other logged-in users. There also are, of course, the classic can't-do-without applications, namely office tools, such as a word processor and spreadsheet (Figure 9). Click the Office link on the eyeOS application dock, and select Word Processor. Although not as full-featured as OpenOffice.org (which you can install, by the way), this word processor does read and write Microsoft Word format (aka .doc) files. There's also a spreadsheet application, an address book and contact manager, a chat client and more.

Although you may not find everything you need right off the bat, an eyeOS software site lists applications by category, complete with descriptions, ratings and installation information. This brings me back around to the whole issue of administration, namely the root user.

If you log in as root, the initial experience is much the same as that for any other user, but there are important differences as

Proven technology. Proven reliability.

When you can't afford to take chances with your business data or productivity, rely on a GS-1245 Server powered by the Intel® Xeon® Processors.

Quad Core Woodcrest



2 Nodes & Up to 16 Cores - in 1U

Ideal for high density clustering in standard 1U form factor. Upto 16 Cores for high CPU needs. Easy to configure failover nodes. Features:

- 1U rack-optimized chassis (1.75in.)
- Up to 2 Quad Core Intel® Xeon® Woodcrest per Node with 1600 MHz system bus
- Up to 16 Woodcrest Cores Per 1U rackspace
- Up to 64GB DDR2.667 & 533 SDRAM Fully Buffered DIMM (FB-DIMM) Per Node
- Dual-port Gigabit Ethernet Per Node
- 2 SATA Removable HDD Per Node
- 1 (x8) PCI_Express Per Node



Servers :: Storage :: Appliances

Genstor Systems, Inc.

780 Montague Express. # 604
San Jose, CA 95131

Www.genstor.com

Email: sales@genstor.com

Phone: 1-877-25 SERVER or 1-408-383-0120



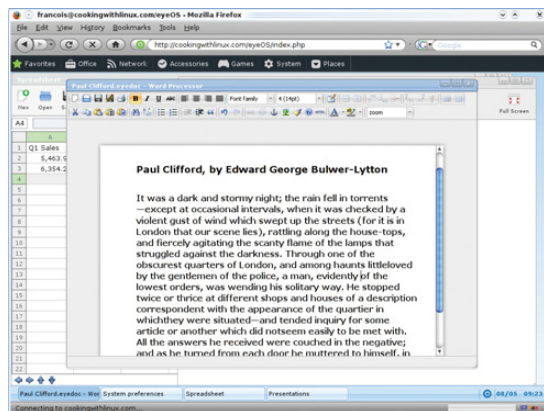


Figure 9. eyeOS even has its own word processor, capable of reading and writing Microsoft Word format files.

well. Most of these you will see when you visit the System Preferences tool. For instance, letting everyone create an account this easily probably is fine if you are truly running from a private cloud, from inside an office or on a private network. But, what if you don't want every person who logs in to have instant access? Under the System Preferences menu, there's a whole submenu for Administration (Figure 10). Under System (Permissions tab), you can turn on and off public registration. If you do so, you then will create users manually.

You also can set a quota for user storage space (the default is set at 1GB), set up repositories for installing software or configure your server to send mail from the eyeOS accounts. Perhaps the most important administrative function you may desire to implement is OpenOffice.org support. This involves a little server-side Linux magic to get things going, not the least of which is to install xfb and OpenOffice.org. The eyeOS wiki has simple instructions that cover several Linux distributions, so I invite you to read the note relevant for your system at the following, friendly URL: wiki.eyeos.org/Setting_Up_Office_Linux.

Speaking of extra software, pay a visit to the eyeOS community apps repository at www.eyeos-apps.org for a huge list of additional packages built for eyeOS. And finally, there's an interesting package (currently in Alpha) that you probably will want to try out. Named eyeSync, this package allows for automatic, transparent synchronization of your eyeOS files with your personal computer. It works with Linux, Windows and Mac OS X Leopard: eyeos.org/en/downloads/eyesync.

There you have it, *mes amis*, a cloud-based OS that is completely open, works well with your existing Linux server and addresses the issues with closed vendor-controlled clouds. It's easy to let everyone share your cloud-based desktop, but if you really want people to stay off of your cloud (unless you want them to, of

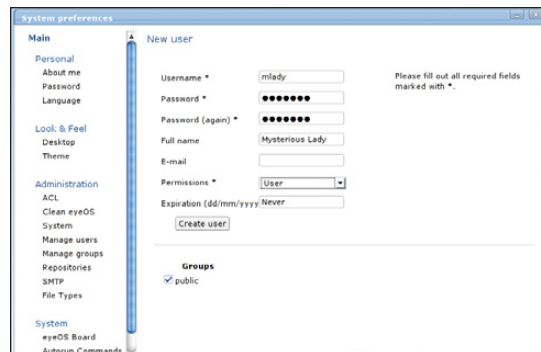


Figure 10. The root user has additional system control functions, such as controlling the addition of users, groups, e-mail and so on.

course), eyeOS provides the tools to do exactly that. You can have hundreds, even thousands, of desktops, invisible and yet always there. A final bit of cool is that if you close down your browser without logging out, you always can come back to your running eyeOS session to the keystroke where you left off.

Yes, *mes amis*, it's that time yet again, when we must say goodbye and return to our respective homes. After tonight, however, your desktop home can follow you wherever you want—always with you but not really there either. Life in the open-source cloud has finally arrived. On that note, I think it's time to ask that most excellent of waiters, François, to refill everyone's glass a final time before we say our goodbyes to one another. Until next time, please, *mes amis*, raise your glasses and let us all drink to one another's health. *A votre santé! Bon appétit!* ■

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

Resources

eyeOS: eyeOS.org

eyeOS for Business: <https://eyeOS.com>

eyeOS Community Apps: www.eyeos-apps.org

Marcel's Web Site: marcelgagne.com

Cooking with Linux: cookingwithlinux.com

WFTL Bytes!: wftlbytes.com

GEMINI SERVERS

The Power of Twin Servers

The iXsystems Gemini Server Class

iXsystems offers an array of server configurations as part of our "Open Source Hardware Design" process. Open Source Hardware Design refers to our carefully crafted server product line, designed and manufactured to ensure operating system and cross-platform compatibility, while providing best in class performance and reliability for any business requirement.

Our Gemini class of servers have superior processing power density and performance/Watt, making them ideal solutions for high-performance computing (HPC) cluster nodes, web servers, or rendering node clusters. They also offer the highest number of CPU cores per rack.

The versatility of these two systems, coupled with these energy saving features, makes them an excellent choice for HPCs, server farms, and other datacenters where space, cost, energy-efficiency, and density are high priorities.

For customers whose needs fall outside of our product line, we offer our custom server solution design process. This means the creation of a custom, open source hardware solution that addresses a company's technical and budgetary needs within their specific network architecture.

For more information about our Gemini class of servers contact iXsystems at (408)943-4100 or visit our website at <http://www.ixsystems.com/gemini> and fill out the inquiry form. One of our expert sales professionals will provide you with a customized quote that best meets your open source hardware solution needs.

800-820-BSDI

<http://www.ixsystems.com>

Enterprise Servers for Open Source

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

© Sergii Tsololo-Dreamstime.com



iX-Gemini Series iX-N12X2

- Two systems/nodes in a 1U form factor
- Each node supports two Intel® Xeon® Processor 5500 series CPU's
- Total of four hot-swap SATA drives, 16 cores, and 96GB DDR3 ECC registered memory (Each node supports two drives and 48GB of memory)
- 1333/1066/800MHz DDR3 ECC registered memory supported
- Each node has its own Dual-port Intel® 82576 GigE network controller
- A 1200W Gold-level High-efficiency (90%+) power supply shared by both nodes to optimize the utilization level and increase energy savings
- Each node has its own remote and lights out management w/IP-KVM SOL via IPMI 2.0 and dedicated LAN
- One (x16) PCI-E low-profile expansion slot per node



iX-Gemini Series iX-N12X8

- Two systems/nodes in a 1U form factor
- Each node supports two Intel® Xeon® Processor 5500 series CPU's
- Total of eight hot-swap SATA drives, 16 cores, and 96GB DDR3 ECC registered memory (Each node supports four drives and 48GB of memory)
- 1333/1066/800MHz DDR3 ECC registered memory supported
- Each node has its own Dual-port Intel® 82576 GigE network controller
- A 1200W Gold-level High-efficiency (90%+) power supply shared by both nodes to optimize the utilization level and increase energy savings
- Each node has its own remote and lights out management w/IP-KVM SOL via IPMI 2.0 and dedicated LAN
- One (x16) PCI-E low-profile expansion slot per node





DAVE TAYLOR

Looking More Closely at Letter and Word Usage

More examples of using the shell to find the frequency of letters used in the English language.

Time and again I have entreated you, dear readers, with my plea for “A letter! My column, nay, my kingdom for a reader letter!” And lo, the miracle occurred, the heavens parted, the angels sang and a letter arrived:

In addition to the letter and word frequency, how about looking at how frequently a letter appears as the first letter of a word? Just to make things more interesting, what is the frequency of two-letter combinations? For instance, if the first letter of a two-letter combination is a t, what is the most frequent second letter? Thanks for the article in *Linux Journal*. It was a good read and nice scripts.—Mike Short

Quando omni flunkus moritati.

First off, before I even read the letter, I was intrigued by the closing quote. Latin? Isn't that, like, a dead language? Turns out the quote's a good one though, especially for IT admins in big companies. It roughly translates to “when all else fails, play dead”, and it comes from the *Red Green Show*, a Canadian comedy. (Thanks Google.)

Now, on to the heart of the letter. Mike's referring to an earlier column where we looked at how to use shell scripts to ascertain letter and word usage, using three books as source material: *Dracula*, *A History of the United States* and *Pride and Prejudice*, all downloaded from Project Gutenberg.

In that series of columns, we ascertained that the ten most common letters in the English language are e, t, a, o, n, i, s, r, h and d. Are they the same if we constrain it to just the first letter of words? Let's find out.

Extracting Just the First Letter of Words

Once we have a corpus of writing and the ability to break it down by words, so that the input stream to the counting script:

```
is
like
this
```

it's done like so:

```
$ cat dracula.txt | tr ' ' '\n' | grep -v '^[[:alpha:]]' | grep -v "^$"

```

That'll turn *Dracula* into the world's narrowest book, with one word per line.

Now we simply can add to it to axe all but the first letter by appending `cut -c1`. The result looks like one of those streams of letters in *The Matrix*, but that's another story.

So, all that's left is to translate uppercase into lowercase, sort, and then use our friend `uniq -c` to tally up the results:

```
tr '[:upper:]' '[:lower:]' | sort | uniq -c | sort -rn | head

```

And, the resultant top ten are:

```
20648 t
15787 a
11110 i
10655 w
9906 h
9030 s
7618 o
5720 m
5411 b
4597 f

```

Quite different! Now, the question is, does it change based on the type of content? Let's do the same command, but this time, let's feed in all three of our books, not just *Dracula* (though with the rabid <cough cough> popularity of *Twilight*, maybe *Linux Journal* would do well to stick with a vampire theme for a few issues?):

```
34359 t
27053 a
18212 w
18119 h
17854 i
15746 s
13614 o
10076 b
9792 m
7712 f

```


It's not exactly the same. Isn't that interesting? I'm not sure what to make of it, but as you can see, a good grasp of shell script commands makes finding out this sort of fairly goofy information interesting.

Calculating Digraphs

But, we're not quite done, because Mike also wondered about two-letter combinations. It's this sort of query that really shows just how helpful becoming savvy on the command line can be. To calculate that requires only one character to be changed in the command invoked above. Do you know what it is?

It's the `cut` command. Above, we're specifying that we want only the very first character of each line of input with `cut -c1`. If we want the first two, we simply can tweak that command flag as appropriate.

But, `-c2` won't work, because that'll give us only the second letter of each word (and the most common second letter in the English language is `o`, followed by `h`, `e`, `a` and `n`).

Instead, we need to use a letter range, which looks like this: `-c1-2`. The result of that invocation is:

```
22100 th
```

```
10168 an
9138 to
7508 he
7100 of
5873 i<space>
5517 in
5332 ha
5157 be
4664 wh
```

There ya go, Mike. The most common two-letter combination in the English language is `th`, which actually makes some sense, followed as a distant second by `an`.

I hope it's trivially obvious how you could use this to calculate the most common three-letter combinations (it should be no surprise at all that *the* is the most common three letter combo, followed by *and*).

I'll wrap up here, but again, I invite you to send me your letters and queries so we can explore various ways to use shell scripts. ■

Dave Taylor has been involved with UNIX since he first logged in to the on-line network in 1980. That means that, yes, he's coming up to the 30-year mark now. You can find him just about everywhere on-line, but start here: www.DaveTaylorOnline.com.

Powerful: Rhino

Rhino M6400/E6500

- Dell Precision M6400/Latitude E6500
- 2.2-3.0 GHz Core 2 Duo or 2.5 GHz Core 2 Quad
- Up to 17" WUXGA LCD w/ X@1920x1200
- NVidia Quadro FX 3700M
- 80-500 GB hard drive
- Up to 16 GB RAM
- DVD±RW or Blu-ray
- 802.11a/g/n
- Starts at \$1330

- High performance NVidia 3-D on a WUXGA widescreen
- High performance Core 2 Quad, 16 GB RAM
- Ultimate configurability — choose your laptop's features
- One year Linux tech support — phone and email
- Three year manufacturer's on-site warranty
- Choice of pre-installed Linux distribution:



Tablet: Raven

Raven X200 Tablet

- ThinkPad X200 tablet by Lenovo
- 12.1" WXGA w/ X@1280x800
- 1.2-1.86 GHz Core 2 Duo
- Up to 8 GB RAM
- 80-320 GB hard drive / 128 GB SSD
- Pen/stylus input to screen
- Dynamic screen rotation
- Starts at \$2200

Rugged: Tarantula

Tarantula CF-30

- Panasonic Toughbook CF-30
- Fully rugged MIL-SPEC-810F tested: drops, dust, moisture & more
- 13.3" XGA TouchScreen
- 1.6 GHz Core 2 Duo
- Up to 8 GB RAM
- 80-320 GB hard drive
- Call for quote

EmperorLinux

...where Linux & laptops converge

www.EmperorLinux.com

1-888-651-6686



Model specifications and availability may vary.



MICK BAUER

Building a Secure Squid Web Proxy, Part IV

Add squidGuard's blacklist functionality to your Squid proxy.

In my previous three columns [April, May and July 2009], I described the concept, benefits and architectural considerations of outbound Web proxies (Part I); discussed basic Squid installation, configuration and operation (Part II); and explained Squid Access Control Lists (ACLs), its ability to run as an unprivileged user and provided some pointers on running Squid in a chroot jail (Part III).

Although by no means exhaustively detailed, those articles nonetheless cover the bulk of Squid's built-in security functionality (ACLs, running nonroot and possibly running chrooted). This month, I conclude this series by covering an important Squid add-on: squidGuard.

squidGuard lets you selectively enforce "blacklists" of Internet domains and URLs you don't want end users to be able to reach. Typically, people use squidGuard with third-party blacklists from various free and commercial sites, so that's the usage scenario I describe in this article.

Introduction to squidGuard

Put simply, squidGuard is a domain and URL filter. It filters domains and URLs mostly by comparing them against lists (flat files), but also, optionally, by comparing them against regular expressions.

squidGuard does *not* filter the actual contents of Web sites. This is the domain of appliance-based

Even if you don't care about conserving bandwidth or enforcing acceptable-use policies, there's still value in configuring squidGuard to block access to "known dangerous" Web sites.

commercial Web proxies such as Blue Coat, and even products like that tend to emphasize URL/domain filtering over actual content parsing due to the high-computing (performance) cost involved.

You may wonder, what have URL and domain filtering got to do with security? Isn't that actually a

form of censorship and bandwidth-use policing? On the one hand, yes, to some extent, it is.

Early in my former career as a firewall engineer and administrator, I rankled at management's expectation that I maintain lists of the most popular URLs and domains visited. I didn't think it was my business what people used their computers for, but rather it should be the job of their immediate supervisors to know what their own employees were doing.

But the fact is, organizations have the right to manage their bandwidth and other computing resources as they see fit (provided they're honest with their members/employees about privacy expectations), and security professionals are frequently in the best "position" to know what's going on. Firewalls and Web proxies typically comprise the most convenient "choke points" for monitoring or filtering Web traffic.

Furthermore, the bigger domain/URL blacklists frequently include categories for malware, phishing and other Web site categories that do, in fact, have direct security ramifications. For example, the free Shalla's Blacklists include more than 27,600 known sources of spyware!

Even if you don't care about conserving bandwidth or enforcing acceptable-use policies, there's still value in configuring squidGuard to block access to "known dangerous" Web sites. That's precisely what I'm going to show you how to do.

Getting and Installing squidGuard

If you run a recent version of Fedora, SUSE, Debian or Ubuntu Linux, squidGuard is available as a binary package from your OS's usual software mirrors (in the case of Ubuntu, it's in the universe repositories). If you run RHEL or CentOS, however, you need to install either Dag Wieers' RPM of squidGuard version 1.2, Excalibur Partners' RPM of squidGuard version 1.4, or you'll have to compile squidGuard from the latest source code, available at the squidGuard home page (see Resources for the appropriate links).

Speaking of squidGuard versions, the latest stable version of squidGuard at the time of this writing is squidGuard 1.4. But, if your Linux distribution of choice provides only squidGuard 1.2, as is the case with Fedora 10 and Ubuntu 9.04, or as with OpenSUSE 11.1, which has squidGuard 1.3, don't worry. Your

distribution almost certainly has back-ported any applicable squidGuard 1.4 security patches, and from a functionality standpoint, the most compelling feature in 1.4 absent in earlier versions is support for MySQL authentication.

Hoping you'll forgive my Ubuntu bias of late, the command to install squidGuard on Ubuntu systems is simply:

```
bash-$ sudo apt-get squidguard
```

As noted above, squidGuard is in the universe repository, so you'll either need to uncomment the universe lines in `/etc/apt/sources.list`, or open Ubuntu's Software Sources applet, and assuming it isn't already checked, check the box next to Community-maintained Open Source software (universe), which will uncomment those lines for you.

Besides using `apt-get` from a command prompt to install squidGuard, you could instead use the Synaptic package manager. Either of these three approaches automatically results in your system's downloading and installing a deb archive of squidGuard.

If you need a more-current version of squidGuard than what your distribution provides and are willing to take it upon yourself to keep it patched for emerging security bugs, the squidGuard home page has complete instructions.

Getting and Installing Blacklists

Once you've obtained and installed squidGuard, you need a set of blacklists. There's a decent list of links to these at squidguard.org/blacklists.html, and of these, I think you could do far worse than Shalla's Blacklists (see Resources), a free-for-non-commercial-use set that includes more than 1.6 million entries organized into 65 categories. It's also free for commercial use; you just have to register and promise to provide feedback and list updates. Shalla's Blacklists are the set I use for the configuration examples through the rest of this article.

Once you've got a blacklist archive, unpack it. It doesn't necessarily matter where, so long as the entire directory hierarchy is owned by the same user and group under which Squid runs (proxy:proxy on Ubuntu systems). A common default location for blacklists is `/var/lib/squidguard/db`.

Expert included.

Meet Victoria (on the right). She is the Silicon Mechanics marketing expert responsible for the events and promotions that keep our customers informed about exciting new products and technologies. She's pictured here with her twin sister Veronica, an industrial designer, to help us make a point about what makes twin servers from Silicon Mechanics so popular. Victoria and Veronica are twins, but they don't look exactly alike and they don't do the same job. Twin servers are two servers in a single 1U chassis: they can be configured differently, and they handle their own individual workloads.

With the introduction of the Rackform iServ R4410 from Silicon Mechanics, twin power has reached a whole new level: the twin². A twin² is a 2U 4-node system. It supports four swappable, full-featured nodes in a 2U chassis with redundant power. In each node you'll find 2 of the new Intel®

Xeon® 5500 Series processors, 12 DDR3 DIMM slots, 3 hot-swap drives, and an integrated dual-port GigE adapter. Integrated InfiniBand is also available with the R4410-IB. Unmatched density and state-of-the-art processors make the R4410 a superior choice for high-performance computing, and Victoria is spreading the word with enthusiasm.

When you partner with Silicon Mechanics, you get more than the latest and greatest in density, performance, and energy efficiency—you get an expert like Victoria.

For more information about the Rackform iServ R4410 visit www.siliconmechanics.com/R4410



Silicon Mechanics and the Silicon Mechanics logo are registered trademarks of Silicon Mechanics, Inc. Intel, the Intel logo, Xeon, and Xeon Inside, are trademarks or registered trademarks of Intel Corporation in the US and other countries.



To extract Shalla's Blacklists to that directory, I move the archive file there:

```
bash-$ cp mv shallalist.tar.gz
/var/lib/squidguard/db
```

Then, I unpack it like this:

```
bash-$ sudo -s
bash-# cd /var/lib/squidguard/db
bash-# tar --strip 1 -xvzf shallalist.tar.gz
bash-# rm shallalist.tar.gz
```

In the above tar command, the `--strip 2` option strips the leading `BL/` from the paths of everything extracted from the shallalist tar archive. Without this option, there would be an additional directory (`BL/`) under `/var/lib/squidguard/db` containing the blacklist categories, for example, `/var/lib/squidguard/db/BL/realestate` rather than `/var/lib/squidguard/db/realestate`. Note that you definitely want to delete the shallalist.tar.gz file as shown above; otherwise, squidGuard will include it in the database into which the contents of `/var/lib/squidguard/db` will later be imported.

Note also that at this point you're still in a root shell; you need to stay there for just a few more commands. To set appropriate ownership and permissions for your blacklists, use these commands:

```
bash-# chown -R proxy:proxy /var/lib/squidguard/db/
bash-# find /var/lib/squidguard/db -type f | xargs chmod 644
bash-# find /var/lib/squidguard/db -type d | xargs chmod 755
bash-# exit
```

And with that, your blacklists are ready for squidGuard to start blocking. After, that is, you *configure* squidGuard to do so.

Configuring squidGuard

On Ubuntu and OpenSUSE systems (and probably others), squidGuard's configuration file `squidGuard.conf` is kept in `/etc/squid/`, and squidGuard automatically looks there when it starts. As root, use the text editor of your choice to open `/etc/squid/squidGuard.conf`. If using a command-line editor like `vi` on Ubuntu systems, don't forget to use `sudo`, as with practically everything else under `/etc/`, you need to have superuser privileges to change `squidGuard.conf`.

squidGuard.conf's basic structure is:

1. Options (mostly paths)
2. Time Rules
3. Rewrite Rules

4. Source Addresses
5. Destination Classes
6. Access Control Lists

In this article, my goal is quite modest: to help you get started with a simple blacklist that applies to all users, regardless of time of day, and without any fancier URL-rewriting than redirecting all blocked transactions to the same page. Accordingly, let's focus on examples of Options, Destination Classes and ACLs. Before you change `squidGuard.conf`, it's a good idea to make a backup copy, like this:

```
bash-$ sudo cp /etc/squid/squidGuard.conf
/etc/squid/squidGuard.conf.def
```

Now you can edit `squidGuard.conf`. First, at the top, leave what are probably the default values of `dbhome` and `logdir`, which specify the paths of squidGuard's databases of blacklists (or whitelists—you also can write ACLs that explicitly *allow* access to certain sites and domains) and its log files, respectively. These are the defaults in Ubuntu:

```
dbhome /var/lib/squidguard/db
logdir /var/log/squid
```

These paths are easy enough to understand, especially considering that you just extracted Shalla's Blacklists to `/var/lib/squidguard/db`. Whatever you do, do not leave a completely blank line at the very top of the file; doing so prevents squidGuard from starting properly.

Next, you need to create a Destination Class. This being a security column, let's focus on blocking access to sites in the spyware and remotecontrol categories. You certainly don't want your users' systems to become infected with spyware, and you probably don't want users to grant outsiders remote control of their systems either.

Destination Classes that describe these two categories from Shalla's Blacklists look like this:

```
dest remotecontrol {
    domainlist    remotecontrol/domains
    urllist       remotecontrol/urls
}

dest spyware {
    domainlist    spyware/domains
    urllist       spyware/urls
}
```

As you can see, the paths in each domainlist

When constructing ACLs, remember that order matters!

and urllist statement are relative to the top-level database path you specified with the dbhome option. Note also the curly bracket `}` placement: left brackets always immediately follow the destination name, on the same line, and right brackets always occupy their own line at the end of the class definition.

Finally, you need an ACL that references these destinations—specifically, one that blocks access to them. The ACL syntax in squidGuard is actually quite flexible, and it's easy to write both "allow all except..." and "block all except..." ACLs. Like most ACL languages, they're parsed left to right, top to bottom. Once a given transaction matches any element in an ACL, it's either blocked or passed as specified, and not matched against subsequent elements or ACLs.

You can have multiple ACL definitions in your squidGuard.conf file, but in this example scenario, it will suffice to edit the default ACL. A simple default ACL that passes all traffic unless destined for sites in the remotecontrol or spyware blacklists would look like this:

```
acl {
  default {
    pass !remotecontrol !spyware all
    redirect http://www.google.com
  }
}
```

In this example, default is the name of the ACL. Your default squidGuard.conf file probably already has an ACL definition named default, so be sure either to edit that one or delete it before entering the above definition; you can't have two different ACLs both named default.

The pass statement says that things matching remotecontrol (as defined in the prior Destination Class of that name) do *not* get passed, nor does spyware, but all (a wild card that matches anything that makes it that far in the pass statement) does. In other words, if a given destination matches anything in the remotecontrol or spyware blacklists (either by domain or URL), it won't be passed, but rather will be redirected per the subsequent redirect statement, which points to the Google home page.

Just to make sure you understand how this works, let me point out that if the wild card all occurred before `!remotecontrol`, as in "pass all !remotecontrol !spyware", squidGuard would not block anything, because matched transactions aren't compared against any elements that follow the element they matched. When constructing ACLs, remember that order matters!

I freely admit I'm being very lazy in specifying that as my redirect page. More professional system administrators would want to put a customized "You've been redirected here because..." message onto a Web server under their control



Want your business to be more productive?

The ASA Servers powered by the Intel Xeon Processor provide 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.)

1U Server - ASA1401i

- 1TB Storage Installed. Max - 3TB.
- Intel Dual core 5030 CPU (Qty-1). Max-2 CPUs
- 1GB 667MGZ FBDIMMs Installed.
- Supports 16GB FBDIMM.
- 4X250GB htswap SATA-II Drives Installed.
- 4 port SATA-II RAID controller.
- 2X10/100/1000 LAN onboard.



2U Server - ASA2121i

- 4 TB Storage Installed. Max - 12 TB.
- Intel Dual core 5050 CPU.
- 1GB 667MGZ FBDIMMs Installed.
- Supports 16GB FBDIMM.
- 16 port SATA-II RAID controller.
- 16X250GB htswap SATA-II Drives Installed.
- 2X10/100/1000 LAN onboard.
- 800w Red PS.



3U Server - ASA3161i

- 4TB Storage Installed. Max - 12TB.
- Intel Dual core 5050 CPU.
- 1GB 667MGZ FBDIMMs Installed.
- Supports 16GB FBDIMM.
- 16 port SATA-II RAID controller.
- 16X250GB htswap SATA-II Drives Installed.
- 2X10/100/1000 LAN onboard.
- 800w Red PS.



5U Server - ASA5241i

- 6TB Storage Installed. Max - 18TB.
- Intel Dual core 5050 CPU.
- 4GB 667MGZ FBDIMMs Installed.
- Supports 16GB FBDIMM.
- 24X250GB htswap SATA-II Drives Installed.
- 24 port SATA-II RAID. CARD/BBU.
- 2X10/100/1000 LAN onboard.
- 930w Red PS.



8U Server - ASA8421i

- 10TB Storage Installed. Max - 30TB.
- Intel Dual core 5050 CPU.
- Quantity 42 Installed.
- 1GB 667MGZ FBDIMMs.
- Supports 32GB FBDIMM.
- 40X250GB htswap SATA-II Drives Installed.
- 2X12 Port SATA-II Multilane RAID controller.
- 1X16 Port SATA-II Multilane RAID controller.
- 2X10/100/1000 LAN onboard.
- 1300 W Red Ps.



All systems installed and tested with user's choice of Linux distribution (free). ASA Collocation—\$75 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®, 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 typographic errors.



Powerful.
Efficient.

and list that URL instead. Alternatively, squidGuard comes with a handy CGI script that displays pertinent transaction data back to the user. On Ubuntu systems, the script's full path is `/usr/share/doc/squidguard/examples/squidGuard.cgi.gz`.

This brings me to my only complaint about squidGuard: if you want to display a custom message to redirected clients, you either need to run Apache on your Squid server and specify an `http://localhost/` URL, or specify a URL pointing to some other Web server you control. This is in contrast to Squid itself, which has no problem displaying its own custom messages to clients without requiring a dedicated HTTP daemon (either local or external).

To review, your complete sample `squidGuard.conf` file (not counting any commented-out lines from the default file) should look like this:

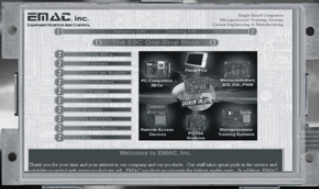
```
dbhome /var/lib/squidguard/db
logdir /var/log/squid


dest remotecontrol {
    domainlist    remotecontrol/domains
    urllist       remotecontrol/urls
}
```

Low Cost Panel PC

PPC-E7

- Cirrus ARM9 200MHz CPU
- 3 Serial Ports & SPI
- Open Frame Design
- 3 USB 2.0 Host Ports
- 10/100 BaseT Ethernet
- SSC-I2S Audio Interface
- SD/MMC Flash Card Interface
- Battery Backed Real Time Clock
- Up to 64 MB Flash & 128 MB RAM
- Linux with Eclipse IDE or WinCE 6.0
- JTAG for Debugging with Real-Time Trace
- WVGA (800 x 480) Resolution with 2D Accelerated Video
- Four 12-Bit A/Ds, Two 16-Bit & One 32-Bit Timer/Counters





2.6 Kernel

Setting up a Panel PC can be a puzzling experience. However, the PPC-E7 Compact Panel PC comes ready to run with the Operating System installed on Flash Disk. Apply power and watch either the Linux X Windows or the Windows CE User Interface appear on the vivid color LCD. Interact with the PPC-E7 using the responsive integrated touch-screen. Everything works out of the box, allowing you to concentrate on your application, rather than building and configuring device drivers. Just Write-It and Run-It. Starting at \$495.

For more info visit: www.emacinc.com/panel_pc/ppc_e7.htm

Since 1985
OVER
23
YEARS OF
SINGLE BOARD
SOLUTIONS

EMAC, inc.

EQUIPMENT MONITOR AND CONTROL

Phone: (618) 529-4525 • Fax: (618) 457-0110 • Web: www.emacinc.com

```
}

dest spyware {
    domainlist    spyware/domains
    urllist       spyware/urls
}

acl {
    default {
        pass !remotecontrol !spyware all
        redirect http://www.google.com
    }
}
```

Now that squidGuard is configured and, among other things, knows where to look for its databases, you need to create actual database files for the files and directories under `/var/lib/squidGuard/db`, using this command:

```
bash-$ sudo -u proxy squidGuard -C all
```

This imports all files specified in active Destination Class definitions in `squidGuard.conf`, specifically in this example, the files `remotecontrol/domains`, `remotecontrol/urls`, `spyware/domains` and `spyware/urls`, into Berkeley DB-format databases. Obviously, squidGuard can access the blacklists much faster using database files than by parsing flat text files.

Configuring Squid to Use squidGuard

The last thing you need to do is reconfigure Squid to use squidGuard as a redirector and tell it how many redirector processes to keep running. The location of your squidGuard binary is highly distribution-specific; to be sure, you can find it like this:

```
bash-$ which squidGuard
/usr/bin/squidGuard
```

As for the number of redirector processes, you want a good balance of system resource usage and squidGuard performance. Starting a lot of redirectors consumes resources but maximizes squidGuard performance, whereas starting only a couple conserves resources by sacrificing squidGuard performance. Ubuntu's default of 5 is a reasonable middle ground.

The `squid.conf` parameters for both of these settings (redirector location and number of processes) are different depending on with which version of Squid you're using squidGuard. For Squid versions 2.5 and earlier, they're `redirect_program` and `redirect_children`. For Squid versions 2.6 and later, they're `url_rewrite_program` and `url_rewrite_children`.

For example, on my Ubuntu 9.04 system, which runs Squid version 2.7, I used a text editor (run via `sudo`) to add the following two lines to `/etc/squid/squid.conf`:

```
url_rewrite_program /usr/bin/squidGuard
url_rewrite_children 5
```

As with any other time you edit `/etc/squid/squid.conf`, it's probably a good idea to add custom configuration lines before or after their corresponding comment blocks. `squid.conf`, you may recall, is essentially self-documented—it contains many lines of example settings and descriptions of them, all in the form of comments (lines beginning with `#`). Keeping your customizations near their corresponding examples/defaults/comments both minimizes the chance you'll define the same parameter in two different places, and, of course, it gives you easy access to information about the things you're changing.

By the way, I'm assuming Squid itself *already* is installed, configured and working the way you want it to (beyond blacklisting). If you haven't gotten that far before installing squidGuard, please refer to my previous three columns (see Resources).

Before those changes take effect, you need to restart Squid. On most Linux systems, you can use this command (omitting the `sudo` if you're already in a root shell):

```
bash-$ /etc/init.d/squid reload
```

If you get no error messages, and if when you do a `ps -axuw |grep squid` you see not only a couple Squid processes, but also five squidGuard processes, then congratulations! You've now got a working installation of squidGuard.

But is it actually doing what you want it to do? Given the filters we just put in place, the quickest way to tell is, on some client configured to use your Squid proxy, to point a browser to `http://www.gotomypc.com` (a site in the remotecontrol blacklist). If everything's working correctly, your browser will not pull up gotomypc, but rather Google. squidGuard is passive-aggressively encouraging you to surf to a safer site!

Conclusion

squidGuard isn't the only Squid add-on of interest to the security-conscious. `squidtailed` and `squidview`, for example, are two different programs for monitoring and creating reports from Squid logs (both of them are available in Ubuntu's universe repository). I leave it to you though to take your Squid server to the next level.

This concludes my introductory series on building a secure Web proxy with Squid. I hope you're off to a good, safe start! ■

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

Resources

squidGuard home page, featuring squidGuard's latest source code and definitive documentation: squidguard.org.

OpenSUSE's squidGuard page: en.opensuse.org/SquidGuard.

squidGuard 1.2 RPMs for Fedora, CentOS and RHEL from Dag Wieers: dag.wieers.com/rpm/packages/squidguard.

squidGuard 1.4 RPM for CentOS 5, from Excalibur Partners LLC: www.excaliburtech.net/archives/46.

The Debian Wiki's "Rudimentary squidGuard Filtering" page: wiki.debian.org/DebianEdu/HowTo/SquidGuard.

Wessels, Duane: *Squid: The Definitive Guide*. Sebastopol, CA: O'Reilly Media, 2004 (includes some tips on creating and using a Squid chroot jail).

The Squid home page, where you can obtain the latest source code and binaries for Squid: www.squid-cache.org.

The Ubuntu Server Guide's Squid chapter: <https://help.ubuntu.com/8.10/serverguide/C/squid.html>.

The Squid User's Guide: www.deckle.co.za/squid-users-guide/Main_Page.

Shalla's Blacklists are available at www.shallalist.de (the most current blacklist archive is always at www.shallalist.de/Downloads/shallalist.tar.gz).

"Building a Secure Squid Web Proxy, Part I" by Mick Bauer, *LJ*, April 2009: www.linuxjournal.com/article/10407.

"Building a Secure Squid Web Proxy, Part II" by Mick Bauer, *LJ*, May 2009: www.linuxjournal.com/article/10433.

"Building a Secure Squid Web Proxy, Part III" by Mick Bauer, *LJ*, July 2009: www.linuxjournal.com/article/10488.



KYLE RANKIN

What Really IRCs Me: Instant Messaging

Use Bitlbee to roll all of your instant-messaging accounts into an IRC interface and do all your chatting from one place.

To me, IRC is the ideal interface for quick communication with my friends. I keep a console IRC session (irssi) running on my server at all times within screen. With that setup, I constantly can lurk in all of the channels I want to follow and reconnect to the session, no matter what machine I am using. Because many of my friends use IRC, it's pretty easy to stay in touch. I can chat with them daily, and if they need to tell me something when I'm not around, they can leave me a private message, and I will see it the next time I'm in front of my computer. To me, the IRC interface is best both for group and private chats—so much so that I prefer it to instant messaging.

Of course, not all of my friends use IRC. Even among those who do, they don't all prefer to do all of their communication there. So, in addition to IRC, I maintain instant-messaging accounts. This means to keep in touch with everyone, I

Basically, Bitlbee sets up an IRC server on your local machine that you can connect to like any other IRC server you might already use.

need to keep both an IRC and an instant-messaging program open. Plus, unless I set up a text-based IM client on my server, I'd have to fire up a local client on whatever computer I'm in front of, which isn't possible when I'm using someone else's computer. On top of that, some of my friends have replaced both chat and IM with Twitter, which means yet another account and yet another program open on my desktop—well, it would in theory at least. Instead, I've discovered a few programs that let me roll everything into IRC sessions, so sending someone an IM is as simple as an IRC private message, and everyone's Twitter feeds become just another comment in an IRC channel. In this column, I discuss how to access your IM accounts from within IRC, and in a follow-up column, I will talk about how

to access Twitter as well, because they each require different programs.

IRC Instant Messaging with Bitlbee

The program that makes IM possible within IRC is an IM-to-IRC gateway called Bitlbee (www.bitlbee.org). Basically, Bitlbee sets up an IRC server on your local machine that you can connect to like any other IRC server you might already use. Once you connect to the server, you can join the #bitlbee channel and authenticate with the bot inside. Then, you can configure Bitlbee with your Jabber, MSN, Yahoo or Oscar (AIM/ICQ) accounts. Once you are set up, when your friends are on-line, they join the channel, and when you talk to them or private-message them inside the IRC channel, it translates it to an instant message.

Bitlbee should be packaged for most major distributions, so you can install it like any other program. Otherwise, just pull down and compile the source code from the main project page. Bitlbee uses inetd, so once you connect to the IRC port, inetd automatically spawns a Bitlbee process. Depending on your distribution, the post-install script may or may not set up the line in inetd.conf automatically. If it doesn't, add the following line to /etc/inetd.conf:

```
6667 stream tcp nowait bitlbee /usr/sbin/tcpd /usr/sbin/bitlbee
```

Set Up the Bitlbee Account

Once Bitlbee is installed, go to your IRC program and connect to a new server, but in place of the typical hostname, connect to localhost. Once you connect to the server, join the #bitlbee channel. Bitlbee includes a built-in help program. Simply type `help` to see a list of help topics, or type `help` followed by a particular Bitlbee bot command to see help for that command. In addition to these help topics, Bitlbee also includes a quickstart topic (type `help quickstart`) that will walk you through setting up your Bitlbee account and adding your IM accounts (I cover these same steps below).

The first thing you need to do before you can IM with Bitlbee is register an account with the

server so that it can save all of your IM account settings, contacts and other information, and password-protect it. Type:

```
register password
```

and replace `password` with the password you want to use. The next time you connect to Bitlbee, you must type:

```
identify password
```

in the `#bitlbee` channel so the bot can give you access to your IM accounts.

Add IM Accounts

Once you are registered, you can start adding IM accounts. The `account` command lets you add or remove accounts from Bitlbee, and the syntax for adding an account is:

```
account add protocol username password server
```

The `protocol` above should be replaced with `jabber`, `msn`, `yahoo` or `oscar`, depending on which chat protocol your IM account uses. Then, list your user name and password for that IM account. The final `server` field is needed only for the `oscar` protocol, so it knows whether to connect to the AOL Instant Messenger server (`login.oscar.aol.com`) or the ICQ server (`login.icq.com`). The rest of the protocols don't need it. So, for instance, if I had an AOL Instant Messenger account called `test` with a password of `mypassword`, I would add it with the following command:

```
account add oscar test mypassword
login.oscar.aol.com
```

After you have added all of your IM accounts, type:

```
account on
```

in the `#bitlbee` channel, and Bitlbee will enable and log in to all of your accounts. Bitlbee should download your contact list automatically, and those contacts that are on-line will show up as though they joined the channel. Because Bitlbee renames people on your contact list so they have a more IRC-friendly name, you might end up with people from different accounts with similar or at least confusing names. To clear things up, just use the `rename` command followed by the old nickname and then the new nickname you want to use.

Chat in Bitlbee

Once your accounts are set up, you can chat with any person who is currently in the `#bitlbee` channel.

Simply type their nickname, followed by a colon (:), and then say what you want to say. Alternatively, you can use `/msg` to set up a private chat, just like with any other IRC channel.

Contact List Management

Once you start using Bitlbee, you probably will get to the point where you need to add or remove contacts from your contact list. The add and remove commands take care of this, but first, type:

```
account list
```

to get a list of the accounts you have registered and their Bitlbee number. Then, to add a user, type `add`, by the number you saw in the account list associated with the account, and, finally, add the user's handle. So, if I wanted to add a user named `mybuddy` to the first account I set up (so it would be `account 0`), I would type:

```
add 0 mybuddy
```

To remove that user from my contact list, I would type:

```
remove mybuddy
```

Because Bitlbee gives each user a unique nickname in the channel, you don't have to specify

Bitlbee should download your contact list automatically, and those contacts that are on-line will show up as though they joined the channel.

the IM account with which a nickname is associated when you remove it.

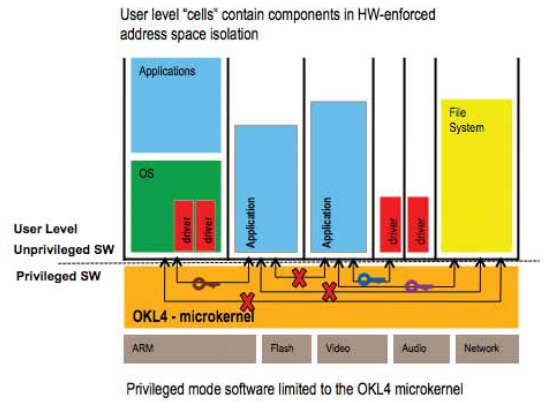
There you have it. Now you're ready to do all of your IM chatting within your IRC session. Of course, there are a number of other commands and settings you can tweak in Bitlbee. To start, type `help` and look at all the available help topics and tutorials within the channel. I recommend you run through all of the quickstart topics first and then branch out into the rest of the commands. As for me, I'll be lurking in IRC (such as the `#linuxjournal` channel on `irc.freenode.net`) as always. ■

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

Open Kernel Labs' OK:Symbian

In an effort to expand the reach of the newly open Symbian platform, Open Kernel Labs has released OK:Symbian, an open-source, ready-to-run, paravirtualized version of Symbian. OK:Symbian enables the Symbian platform to be used as a guest operating system running in a secure hypercell on top of the OK Labs OKL4 microvisor. Open Kernel Labs says that its HyperCell Architecture allows for higher levels of security and robustness and enables the use of the Symbian platform in new lower-cost devices and in new ways. OK:Symbian also lets handset OEMs, MNOs and mobile-phone users benefit from the tens of thousands of existing Symbian applications and the global developer community for the platform. In related news, Open Kernel Labs announced its joining of the Symbian Foundation, as well as its contribution of OK:Symbian to the Symbian Open Source community.

www.ok-labs.com



Openbravo ERP

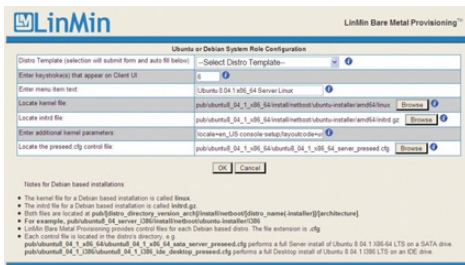
The new Openbravo ERP 2.50 was recently released by its eponymous parent company. In the new 2.50 version, the Web-based, open-source Enterprise Resource Planning (ERP) and Point of Sale (POS) solution for businesses is now available as a professional subscription service. The new release also introduces a modular architecture and adds a host of new features and functions, such as support for right-to-left languages (such as Arabic and Chinese), additional smart-build processes, autosave, more user alerts and enhanced support for complex organizations. Furthermore, Openbravo states that it is easier than ever before for the community and third parties to customize and create their own new features and functions. One can browse and use shared functionality created by other users or deploy third-party modules shared on the new Openbravo Forge.

www.openbravo.com

Marketcetera Trading Platform

The Marketcetera open-source platform for automated trading recently announced a significantly upgraded version 1.5. According to Marketcetera, stock-market data volumes are exploding and "automated trading is becoming more prevalent on the buy side and across more asset classes, not just equities". The company also says that its application helps investment firms make fast, intelligent trading decisions at lower costs per transaction. New features in the latest release include real-time intraday position and profit-and-loss monitoring; simplicity and security for multiuser installations; Level 2 and depth-of-book market data and strategy agent integration via the new Strategy Studio.

www.marketcetera.com

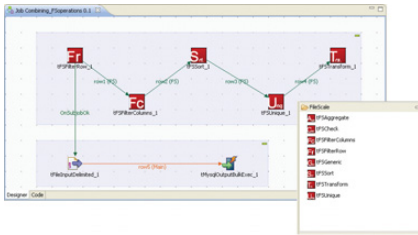


LinMin Bare Metal Provisioning

The news out of LinMin is its upgraded version 5.4 of LinMin Bare Metal Provisioning, a system provisioning and imaging solution that can be implemented by IT organizations of any size with limited budgets. Combining server provisioning and disk imaging in a single product, LinMin is a solution for deploying, repurposing and recovering the commodity hardware infrastructure layer used in hosting, corporate, cloud and other data-center environments. LinMin says that the new release adds the "Turbo-Imaging" high-performance disk imaging subsystem for disaster recovery, new operating system media management, updated

Linux and Windows Server provisioning, extensive logging and others. The firm also adds that Turbo-Imaging adds automatic filesystem detection, intelligent compression and other capabilities to ease the rollback of systems back to a known good state.

www.linmin.com



Talend Integration Suite MP^x

The good folks at Talend have released the Talend Integration Suite MP^x, a new enterprise data-integration platform that the company says “is designed to help organizations attain the highest levels of performance and shatter the limits typically associated with traditional data integration processes”. The solution is based on the Talend Integration Suite while adding the new FileScale technology, a breakthrough that allows organizations to conduct highly parallelized processing and reduce limitations inherent in traditional data-integration architectures. It further allows integration processes to sort, filter and

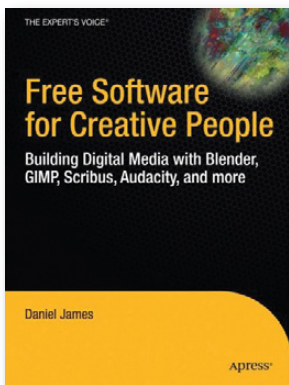
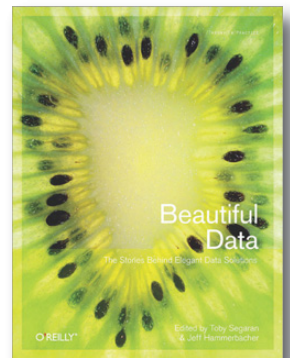
merge data, perform aggregation and arithmetic functions, and transform and ensure the compliance of data. Finally, the new package features multiple levels of massive parallelization, allowing the execution of separate subprocesses in parallel, breakdown of data sets into many parallel streams and the ability to leverage parallel database loaders.

www.talend.com

Toby Segaran and Jeff Hammerbacher's *Beautiful Data* (O'Reilly)

Take a more esoteric and artistic book to the beach this month with Toby Segaran and Jeff Hammerbacher's new title *Beautiful Data* from O'Reilly. In a nutshell, the book illustrates how wide-ranging and beautiful working with data can be. Thirty-nine of the best data practitioners in the field explain how they developed simple and elegant solutions on projects ranging from the Mars lander to a Radiohead video. Other topics include the visualization of trends in urban crime using maps and data mashups, how crowdsourcing and transparency have combined to advance the state of drug research and how data processing systems are designed to work within the constraints of space travel.

www.oreilly.com



Daniel James' *Free Software for Creative People* (Apress)

Get your creative groove on with Daniel James' *Free Software for Creative People: Building Digital Media with Blender, GIMP, Scribus, Audacity, and More*. Published by Apress, the book is a guide primarily for users of Linux (but also of Windows and Mac OS X!) who want to meet their creative goals without having to pay money for very powerful, world-class tools. Apress calls the book “your university of 2-D and 3-D graphics, and of video-based art and Web presentation”. The book also explains how to chain creative applications together, work cross-platform, share creative work with others, locate software development communities to connect with other creative types and configure Linux systems featuring these applications.

www.apress.com

Opengear's IP-KVM1001 KVM-Over-IP Switch

Opengear has added the IP-KVM1001 KVM-Over-IP Switch to its product lineup, a palm-sized KVM-over-IP solution aimed at branch offices, labs, Web-hosting providers and data centers. The company bills the IP-KVM1001 as a cost-effective “Zero U” device for KVM control “at the rack” or remotely over IP, providing full control of systems during bootup, BIOS maintenance or server/OS lockups. The product is browser-based and offers universal connectivity for USB and PS2 interfaces in one unit. Finally, besides virtual media support for transferring files and installing patches and upgrades over IP, the switch offers a serial port for serial console access to switches, routers, firewalls or an external modem for out-of-band access.

www.opengear.com



Please send information about releases of Linux-related products to newproducts@linuxjournal.com or New Products c/o *Linux Journal*, PO Box 980985, Houston, TX 77098. Submissions are edited for length and content.

Fresh from the Labs

LongoMatch—Sports Video Analysis

www.ylatuya.es

I mentioned LongoMatch in last month's Projects at a Glance sidebar, hoping I could get it to work this month. I assumed it would be tricky to install, being a video-related project, but much to my surprise and delight, it installed with very little fuss, and I'm proud to feature it here this month. According to LongoMatch's Web site:

LongoMatch is a sports video analysis tool for coaches to assist them in making game video analysis. You can tag the most important plays of a game and group them by categories to study each detail of game strategy. A list with all the tagged plays lets you review them with a simple click, even in slow motion. The timeline gives a quick overview of the game and lets you adjust the lead and lag time of each play frame by frame. LongoMatch has support for playlists, an easy way to create presentations with plays from different games. Besides, you can create new videos with your favorite plays using the video editing feature.

Installation My hat is off to Andoni Morales who made the entire installation process pain-free. Head to the download page, and your choices are repositories to add for Ubuntu (I know this isn't *Ubuntu Journal*, sorry) or a source tarball. If you are an Ubuntu user, add the repositories mentioned, update apt-get, and select the package longomatch.

If you're not an Ubuntu user and have to use the source, don't panic, Andoni has marked out the packages you need: autotools-dev, pkg-config, mono-gmcs, libdb4o6.0-cil, libgtk2.0-cil, libgstreamer0.10-dev, libgstreamer-plugins-base0.10-dev, libgtk2.0-dev, libmono2.0-cil and libmono-cairo2.0-cil.

If you use a Debian-based distro, you can use this command (to be run

as root or sudo):

```
apt-get install autotools-dev pkg-config mono-gmcs
libdb4o6.0-cil libgtk2.0-cil libgstreamer0.10-dev
libgstreamer-plugins-base0.10-dev libgtk2.0-dev
libmono2.0-cil libmono-cairo2.0-cil
```

Once the dependencies are out of the way, grab the source tarball, extract it, and open a terminal in the new folder. Enter the commands:

```
$ ./configure
$ make
```

And, as root or sudo:

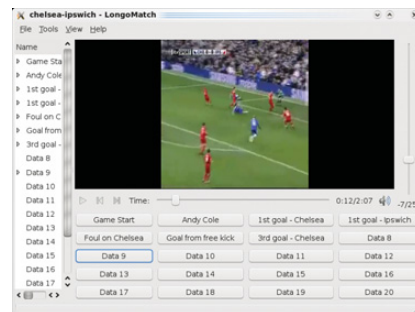
```
# make install
# ldconfig
```

Once the source has finished compiling, you can run the program by entering:

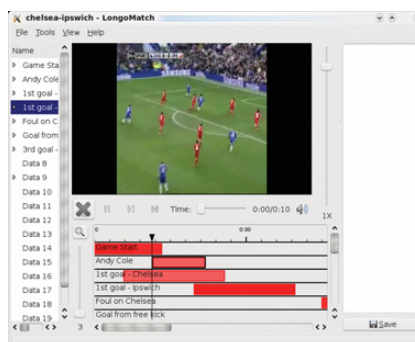
```
$ longomatch
```

Usage Before starting, you need a video to analyze. Let's assume it's a sporting video with two teams, but it doesn't have to be (more on that later). Once inside, you need to start a New Project under the File menu. The New Project window prompts you to enter the name of a Local Team and a Visitor Team, the number of each team's goals, as well as the date of the match. These tools simply make things easier for a sport that fits this profile—soccer being the main example—but they're not requirements, especially if your sport doesn't fit these constraints (my sport is rock climbing for instance, for which video analysis is also very helpful).

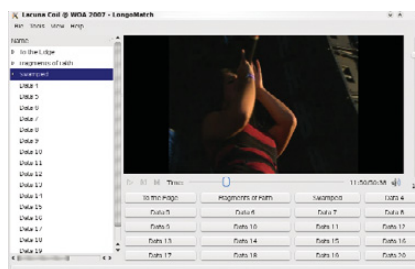
Once you've selected your video and are ready to go, LongoMatch automatically goes to its main screen, Capture Mode, and starts playing your chosen video. When you want to highlight something, such as a scored goal, a penalty and so on, click the Pause button. Then, click the Data 1 button below to note your first bit of information. If you notice the field on the far left of the screen (where it also says Data 1, Data 2, Data 3 and so on),



LongoMatch allows you to mark out important parts of a game or any video.



This is Analyze Mode, where you can fine-tune the timing of your highlights.



Despite being designed for sporting matches, LongoMatch can be used to mark out sections of any video. Here it's being used to mark out a music concert.

you'll see a marking arrow has been placed next to the name of Data 1. If you double-click on it, you now can rename it to whatever you want, such as "free kick".

Click the Play button again, and the video plays until the next event you want to mark. Repeat the steps for Data 1 on the Data 2 button. From here, you will start to build a timeline of events,

Double-click on any of these marked-out sections, and LongoMatch takes you straight to that point in the video.

and obviously, you take the same steps for each data point. If you want to fast-forward or rewind, you can use the slider bar, which is very handy on tedious junior soccer matches! The vertical slider bar on the right controls the playback speed, allowing you to slow down to catch an important Chelsea goal or speed up when you're watching nine-year-olds play out their enthralling 90 minutes of football.

Once you're finished with the Capture Mode and are ready to look at what you've marked for analysis, click on the View menu and choose the Analyze Mode. From here, you can see all the moments marked out on a timeline. Double-click on any of these marked-out sections, and LongoMatch takes you straight to that point in the video.

However, each marked-out point is given a default length that may be too long or too short for what you want to highlight. Click on the start or end of a data point's red bar and drag it left or right, which makes the bar longer or shorter in each section, letting you tighten up the contents of your highlights. The cool thing is that the video updates as you slowly drag the bar around, which really helps with accuracy and pinpointing crucial playback moments.

What I quickly realized with LongoMatch is that it doesn't need to be used only for sporting analysis. It can be used to analyze any video. If you simply ignore the prompts at the start in the New Project section, you can load any video you like and then mark out interesting points. For instance, I used it to mark out interesting sections of a music video, which I then can fast-forward to in the Analyze Mode's timeline section. Getting back to the subject of sports, I can use it on climbing footage to mark out crucial sections of a route to learn the moves and study each one individually.

Andoni has made something very useful here that will find an audience with many coaches and referees, plus it will ease a great deal of stress for

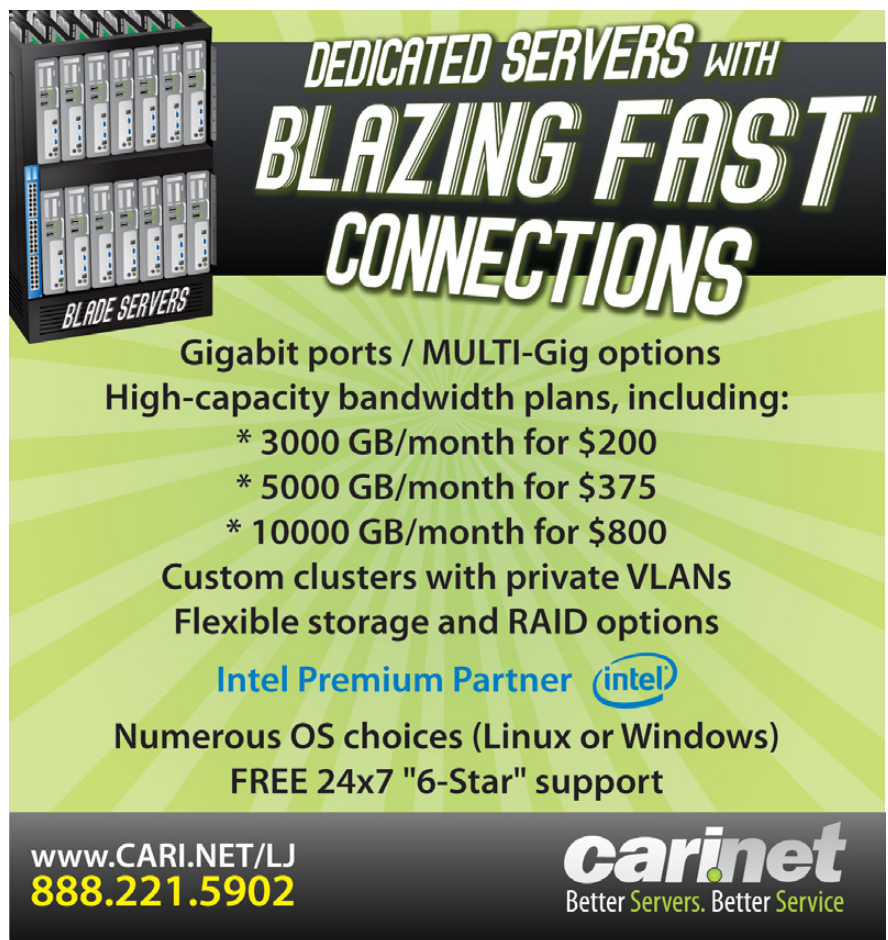
poor dads who've been stuck with the job of highlighting their kids' sporting matches! As LongoMatch is developed in Mono, it also runs on Windows, meaning that the audience for this project is potentially quite broad. It's already getting a following amongst hockey and rugby teams. And, let's not forget the added bonus that this can be used as a general video analysis tool as well, which could make it popular with those who want to use it for multimedia purposes, provided they ignore the sporting features (perhaps a fork is in order, or an alternative GUI?). I'm interested to see where this goes, in the likes of The GIMP or Blender it could become a distro mainstay in one form or another.

Kanatest—Japanese Flashcard Tool

www.clayo.org/kanatest

I was delighted this month to see that Kanatest still is in development, as it's a tool I've been using for quite some time now, and I used it to learn Katakana for the first time (part of a Japanese phonetic alphabet). Recently, there have been improvements and added features, so I thought I'd jump in and have a look. According to the Web site:

Kanatest is a Japanese kana (Hiragana and Katakana) simple flashcard tool. During testing, Kanatest displays a randomly selected kana char (respecting mode and lesson) and waits for the user to answer with the expected romaji equivalent. This process continues until all questions are answered or all questions are answered correctly (depends on options). At the end of the test, a short info page




**DEDICATED SERVERS WITH
BLAZING FAST
CONNECTIONS**

Gigabit ports / MULTI-Gig options
High-capacity bandwidth plans, including:

- * 3000 GB/month for \$200
- * 5000 GB/month for \$375
- * 10000 GB/month for \$800

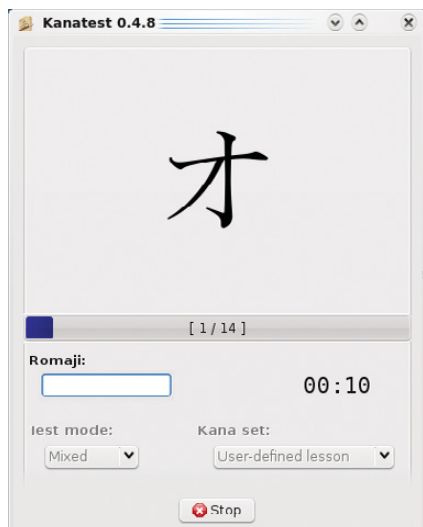
Custom clusters with private VLANs
Flexible storage and RAID options

Intel Premium Partner 

Numerous OS choices (Linux or Windows)
FREE 24x7 "6-Star" support

www.CARI.NET/LJ
888.221.5902

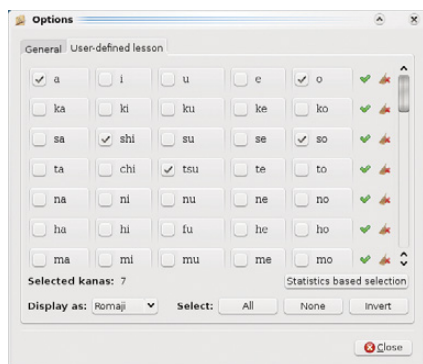
carinet
Better Servers. Better Service



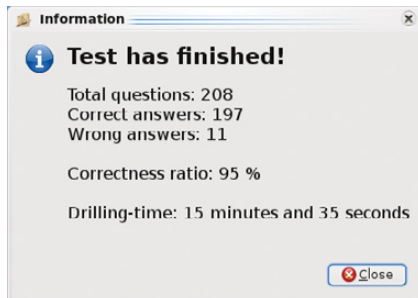
Kanatest presents you with a character in Japanese (kana), and you type it in below in the Roman alphabet.



Newer features in Kanatest include a character chart.



Kanatest also includes user-defined lessons.



At the end of each test, you are presented with your score and accuracy rate—it's been a while since my last lesson, as you can see!

about drilling time and correctness ratio is displayed. The results are stored, and users can review their performance at any time.

Installation Although Kanatest is available in a number of repositories, chances are that it's an old version (at least it was at the time of this writing). I went with the tarball for the latest features and didn't run into any pesky requirements during the compilation. To compile it yourself, head to the Web site, grab the latest tarball, extract it, and open a terminal in the new folder.

Compiling Kanatest is the usual case of:

```
$ ./configure
$ make
$ sudo make install
```

Once it's compiled, you might find it in your menu (mine was under Applications), or you can start it with:

```
$ kanatest
```

Usage Once the program has loaded, Kanatest is ready to go with that tempting Start button on the bottom left, but you might want to define what kind of lesson you want to run first. If you go with the defaults, Kanatest tests you on Katakana and goes through all the characters. If you want to change the lesson, check the drop-down box called Test mode:, where you can choose between Hiragana, Katakana and Mixed. And, for those who know what they're doing, under Kana set:, you can choose from All kanas, Basic kanas, A-I-U-E-O, KA-KI-KU-KE-KO and so on.

When you're ready to go, click Start.

You'll be presented with a screen that shows Japanese phonetic characters above, with an input box where you type in what the character says in Romaji (Japanese using the Roman alphabet). In the middle is a progress bar that shows how many characters you have to go through, and a timer keeps track for those wanting to improve their reading and reaction times. When you get a character wrong on the default settings, the program corrects you, reading out the proper answer, and tallies your correct answers and mistakes against the stopwatch at the end.

What's great for advanced students is that you also can be tested on kana combos such as kyo, ji, kyu and so on. For beginners who have only seen kana for the first time, don't panic, as there's a kana chart available on the program's main screen in that cluster of four buttons on the right. Included in the chart are all the characters in both Hiragana and Katakana, as well as their combinations, such as pya, myu and so on.

For those who aren't really into tweaking things and happily accept defaults, I do recommend checking out the Options section, as this is where you can choose your wanted characters in a user-defined lesson. I thoroughly endorse this move, as there's certain characters I always get confused over, such as Katakana's n, so, shi and tsu. You also can choose to repeat wrongly answered questions instead of having it correct you, as well as change colors, fonts and so on.

Check out the Statistics section too, because here you can keep track of your test scores over time and see your correctness ratio for each kana character.

If you're a Japanese-language student and haven't used Kanatest before, you should do so. It's simple, elegant, painless and probably the best choice there is for testing yourself against kana. If you have used Kanatest before, and it's been a while, check the latest releases, as it really has improved. The user-defined lessons, combined with the Statistics section in particular, make things much better than before. ■

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



LINUXCON

PORTLAND 2009

SEPTEMBER 21 - 23



Guiding the Linux Ecosystem

The **Linux Foundation** presents a brand new technical conference gathering developers, administrators and users of Linux for collaboration, advancement and interaction. Attend **LinuxCon** and leave more knowledgeable and better positioned for success in the year to come.

- 75 Conference Sessions, Tutorials and Mini Summits
- Developer, Operations and Business Tracks
- Education and Collaboration Opportunities
- Speakers include: **Linus Torvalds**, **Mark Shuttleworth**, **Bob Sutor**, **James Bottomley** and **Matt Asay**.

For more information, to become a sponsors or to **register**, please visit: <http://events.linuxfoundation.org/events/linuxcon>



SOFTWARE

KOffice 2.0

The long-awaited upgrade to KOffice has arrived. It looks good and provides a great base for its future evolution.

BRUCE BYFIELD

More than a year after KDE 4.0 unveiled a radically revised desktop, KOffice 2.0 is preparing to release an equally revised office suite, which should be released before this article is published (KOffice 2.0-RC-1 was released in April 2009).

What users will see is not an extensive new feature set, but only a few additions here and there. Instead, just as KDE 4.0 provided the foundation for future developments on the desktop, KOffice promises to provide a solid basis for future improvements. Reflecting changes in the toolkit and library, the newest version of KOffice delivers a common interface across applications, enhanced graphical capacities and new accessibility to existing tools—all wrapped up in a look and feel proving that eye candy can be as much about usability and functionality as about superficial aesthetics. These changes are especially visible in major applications like KWord, KSpread, KPresenter, and Krita and Karbon14 (the main graphics programs), although they are evident in other KOffice applications as well.

This emphasis means that those who were hoping KOffice 2.0 would finally allow the office suite to match the rival OpenOffice.org feature for feature are going to be disappointed. If the late beta I am working from is a guide, KPresenter still will not have the ability to use sound or video, and KSpread will continue to lack filters and pivot tables. In fact, some features of KOffice 1.6.3, the previous official release, such as comments and expressions (autotext) in KWord or tables in KPresenter, may not find their way into KOffice 2.0 either. When you do find new features, they are apt to be fundamental ones, such as more printing options for KSpread.

However, this focus does not mean KOffice is lacking in scope. By any standard, KOffice 2.0 is an ambitious undertaking. With 11 applications to OpenOffice.org's six, and a considerably worse ratio of programmers, any release of KOffice is an exercise in logistics second only to a new version of KDE itself—and version 2.0 is more challenging than most releases. The new release not only marks KOffice's transition to the Qt 4.x toolkit, like most KDE-related software, but also new ports to OS X and Windows.

If that were not enough, version 2.0 also marks the first use of two major libraries: Flake, which introduces a new concept of shapes, together with tools to manage them; and Pigment, a color management library. No wonder, then, that the release is happening 16 months after the KDE 4.0 release and has staggered through ten alpha and seven

beta releases. But, when KOffice 2.0 finally reaches release, the result promises to be a revamping that will allow the project developers to add smaller enhancements in point releases.

Introducing the Interface

Like the KDE 3.0 series, KOffice 1.6.3 is functional but easy to underestimate, because it looks like a refugee from the late 1990s. By contrast, KOffice 2.0 looks as though it is designed to ensure that nobody ever will dismiss it solely on the basis of appearance.

Ever since Microsoft Office 2007 replaced menus and toolbars with ribbons, rival office suites have been faced with the dilemma of either copying and looking modern or retaining the functionality of traditional program design and looking out of date. OpenOffice.org 3.0 met the challenge with a compromise that kept the traditional structure but increased the number of floating palettes or windows—selections of tools that could be positioned anywhere on the desktop or docked in the toolbar or against one side of the editing window. In version 2.0, KOffice's developers have opted for a similar solution, calling them dockers and adding controls for turning each one on or off in the Settings menu.

Dockers are accompanied by two panes to either side of the editing

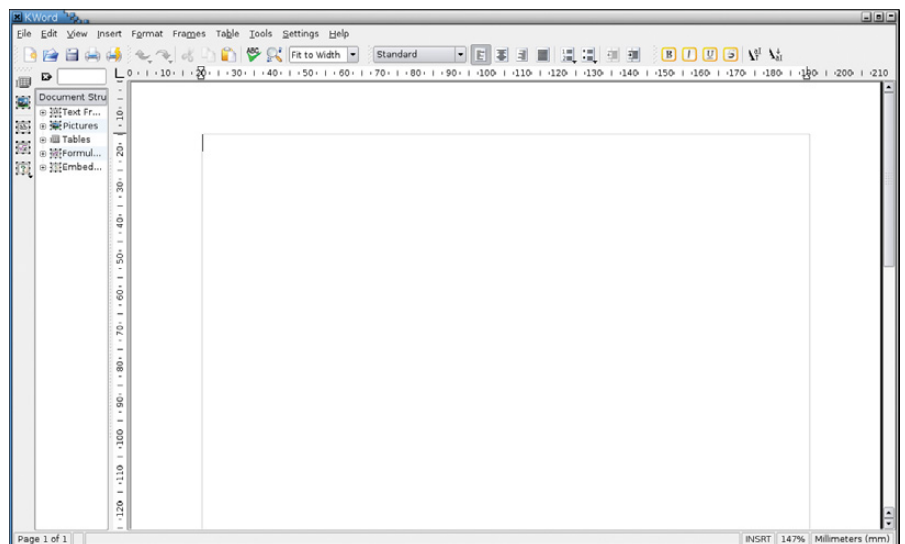


Figure 1. Old KOffice

Just as the Drawing toolbar added increased graphic capacity to the major OpenOffice.org applications, so dockers give KOffice applications more ability to handle pictures and primitives.

graphics programs actually have had a very similar arrangement in earlier releases, which may be where the design originated. But in KSpread, it might seem like worthless clutter, because many of the dockers have to do with graphics or layout, neither of which many spreadsheets need. Similarly, if your word processing never extends beyond a memo, you might find that the default docker pane is overkill. The same is true in KPresenter if you don't do original diagrams.

Still, despite their initially formidable appearance, these panes and dockers do have the advantage of removing

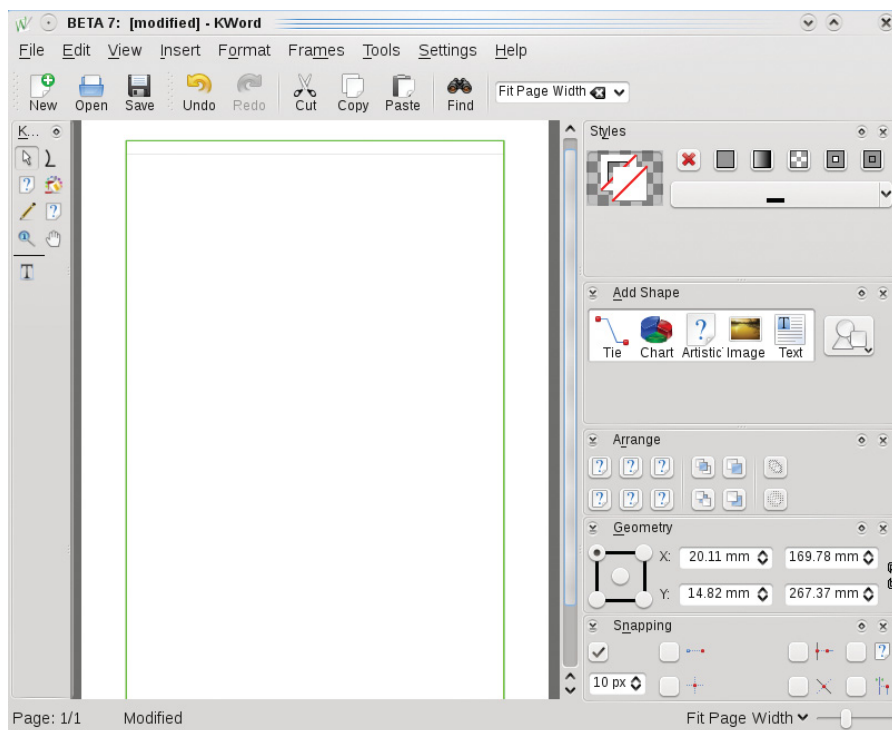


Figure 2. KOffice 2.0 introduces an eye-catching but highly functional interface.

window. On the left is a pane with icons specific to the application. On the right is the pane containing multiple dockers. Click on an icon in the application pane, and the available dockers on the right change. The application pane, the docker pane or any individual docker can be removed from its position to float freely by dragging its title bar with the mouse. You also can drag dockers into different positions on the right-hand pane.

Alternatively, you can close panes, toolbars or dockers, or change the horizontal space given to the docker pane. Unless you are working with a maximized window on a wide-screen monitor, sooner or later, you probably will want to use these customizations to give yourself room to work.

Possibly too, you might want to reduce the number of dockers, especially when you are first learning KOffice 2.0. Otherwise, the effect is like sitting down in the cockpit of a commercial airliner and trying not to be overwhelmed by the dozens of controls available.

The success of this interface varies with the application and your use of it. The layout works best in feature-rich programs, such as Krita and Karbon14, where they increase the accessibility of

tools (although at first you might find yourself peering anxiously as you wait for the mouse-over text to tell you what each icon does). In fact, both these

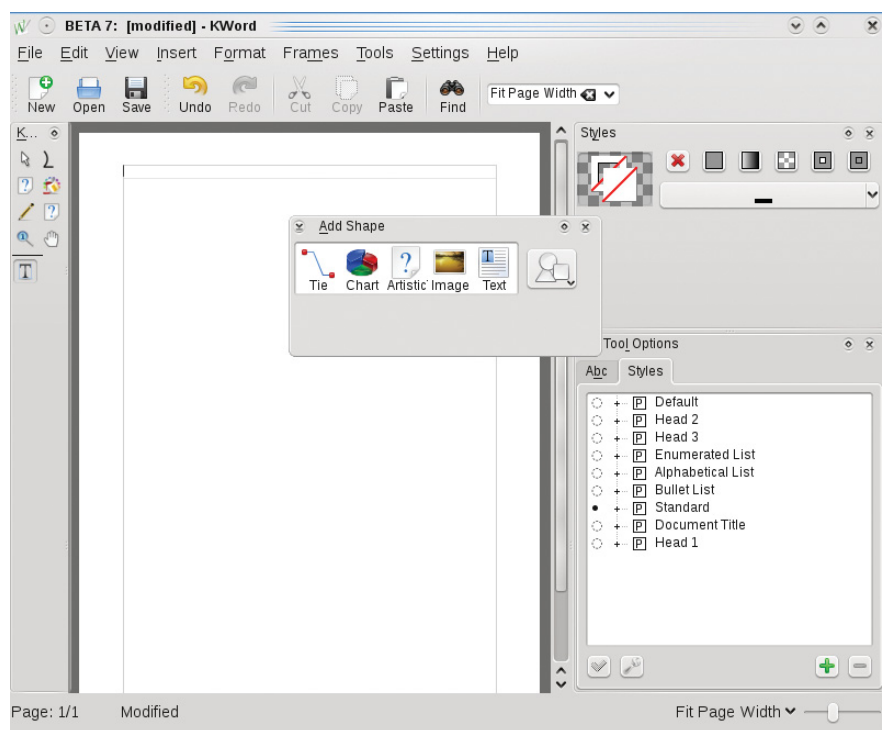


Figure 3. Dockers, toolbars and panes can all be unmoored from their positions to float freely.

Now, in KOffice 2.0, the concept of frames has been replaced with the less abstract and better-labeled ones of shapes—no doubt as a result of implementing the new Flake library.

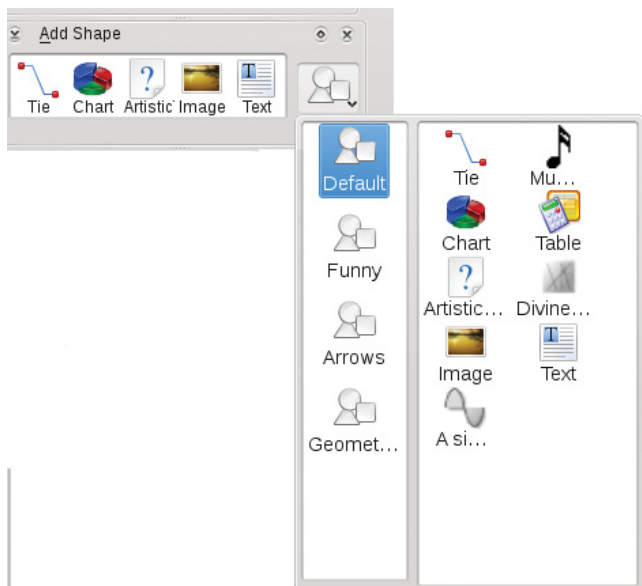


Figure 4. The Add Shape docker not only makes economical use of space, but also makes clear that everything you add to a document is treated the same way.

many tools from their hiding places in the menu and placing them where users can become curious and investigate them. You may find yourself learning more about KOffice applications than ever before, simply because you can see more of the possibilities.

Enhanced Graphical Capacity

Just as the Drawing toolbar added increased graphic capacity to the major OpenOffice.org applications, so dockers give KOffice applications more ability to handle pictures and primitives.

Some of this enhanced capacity is new, such as the calligraphy tool that resembles Inkscape's, or the availability of artistic text—graphical text that can follow angled or curving baselines. Similarly, the addition of ties or connectors gives KPresenter a large boost by adding the ability to create and manipulate organizational charts.

However, a good deal of the across-the-board graphical capacity is simply a reordering of existing tools to make

them more accessible. For example, from the Add Shapes docker, you can not only select basic shapes, such as ties, chart, artistic text and text frames, but also choose from a miniature clip-art gallery that includes arrows, geometric shapes and callouts.

The vaguely named Styles docker provides a similar capacity for the backgrounds of objects. In a docker that is maybe 2" x .5" high on my laptop screen, the Styles docker gives you a selection of

background colors, gradients, patterns and fills, or lets you remove them with a click of a button. These choices can be customized by selecting tools on the application pane, or sometimes, by making selections in other dockers.

As a side benefit, by having these graphical tools in most applications, KOffice also increases its common interface. The result is that both the applications in general and their new graphical capabilities in particular are quick to learn.

Making Old Concepts Clearer

Another advantage of KOffice 2.0's interface is that basic concepts often become clearer. This change is especially obvious in KWord.

In 1.6.3, the latest officially released version, KWord's frame tools provided a tree view of document structure rather like the OpenOffice.org Navigator. However, this view was locked in place and too narrow by default even when KWord was maximized. Nor were the

arcane icons for different types of objects beside the tree view very helpful to users. As a result, most users I have talked to ignore them. Many confess to hiding the tree view to avoid being intimidated by them.

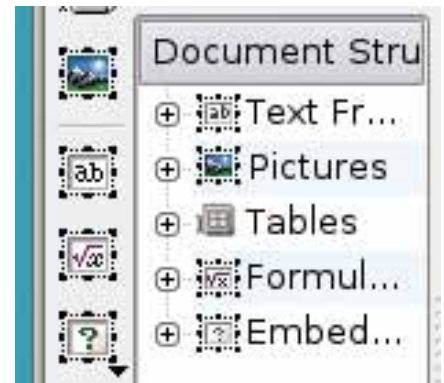


Figure 5. The document structure pane in earlier versions of KOffice confused as much as it enlightened. It's been replaced by dockers such as Add Shape.

Now, in KOffice 2.0, the concept of frames has been replaced with the less abstract and better-labeled ones of shapes—no doubt as a result of implementing the new Flake library. As in earlier releases, you still have to select a type then drag with the mouse in the editing window to create it, but now, with the default set of dockers, you are more likely to notice and use the tool.

Moreover, once you have created an object, you easily can use dockers like Geometry and Snapping to align and orient a shape or arrange it on a grid. Although the functionality is the same as in earlier releases, ease of use is far higher.

Just as important, because Add Shapes lists ties, charts, artistic text and pictures as possible selections, it reinforces the fact that all these possibilities are essentially the same kind of object so far as KOffice is concerned, and all can be manipulated in much the same way in the editing window. In other words, the Add Shapes docker makes clear a unifying concept in a way that separate sub-items in a menu or a collection of unconnected icons could never hope to match.

A second basic concept that becomes clearer in KOffice 2.0 is styles—the formatting equivalent of declaring a variable once and reusing

it as needed. Most word processors have the concept of character and paragraph styles, but they vary widely in their emphasis on them. For instance, AbiWord and MS Office tend to make manual formatting more prominent, while OpenOffice.org requires the use of styles if you want to use many advanced features. In the past, KOffice has been closer to AbiWord than OpenOffice.org, including styles, but keeping them in the menus where they can be missed and their features buried several layers below the top menu.

By contrast, KOffice 2.0 edges closer to emphasizing styles. If you select the text tool from KWord's application pane, you have a Styles docker (not to be confused with the one for backgrounds that uses the same name) that places manual formatting of text and styles only a tab apart. At first, this arrangement might seem to give both approaches to formatting equal weight, but the truth is that styles have been so underemphasized that, just by making them more prominent, the Styles docker increases the chances that users will investigate the time-saving possibilities of working with styles. At the same time, unlike in OpenOffice.org, KOffice 2.0 does not compel users to switch from manual formatting if they choose not to.

The Possible Reception

KOffice 2.0 does have new features that stand on their own, such as additional functions for KSpread and the option to encrypt files while saving them. However, such enhancements seem minor compared to those in the interface. While altering KOffice under the hood, its development team also has made serious efforts to enhance the interface—so much so that the user experience is almost completely different in 2.0 from that of earlier releases.

Some of these changes work better than others. In particular, some of the names could be better chosen, at least in English. Apart from the potential confusion from having two dockers called Styles, some, such as the shapes library called Funny, simply seem inappropriate. Then too, the name dockers itself is always going to set North Americans to thinking of business-casual pants.

Still, KOffice 2.0's final release is less likely to be met with the same hostility that KDE 4.0 encountered. True, the possibility of some missing features remains strong—barring a last-minute coding blitz—and some users will complain about any changes.

However, although KOffice 2.0's changes are impossible to miss, they are far less radical than KDE 4.0's. They are not so much changes in the basic concepts you need to use the office

suite as improvements in usability. Provided users are not immediately intimidated by the array of dockers, they should find KOffice 2.0 more accessible and quicker and easier to use than those of previous releases. These improvements make KOffice 2.0 a joy to use and more than justify the long wait for the final release. ■

Bruce Byfield is a computer journalist who writes regularly for the NewsForge and Linux Journal Web sites.

ServerBeach

by geeks, for geeks™

Linux servers from
\$75/mo



When **YouTube** first started to experience its exponential growth and our hosting needs changed, ServerBeach offered us great flexibility. They continually redesigned our streaming architecture for optimum performance while keeping our hosting costs in check.

STEVE CHEN Founder | **YouTube**



ValuePack (always included)

- > 24/7 live customer service
- > 24/7 ticketing system
- > Personal account manager
- > Lots of bandwidth
- > Free OS reloads
- > Free Rapid Reboot
- > Free Rapid Rescue
- > Super fast PEER 1 network
- > Rock-solid IT infrastructure
- > 100% uptime guarantee
- > Choose your data center - East Coast, West Coast and Central

serverbeach.com **1.800.741.9939**

A PEER 1 COMPANY

Say Goodbye to Reboots with Ksplice[®]

Tired of rebooting for kernel updates?
Good news—now you don't have to,
thanks to Ksplice Uptrack.

WASEEM DAHER

Everyone hates rebooting for updates. When system administrators reboot their servers, they have to manage an inconvenient outage window—quite possibly during the middle of the night—and they have to deal with the lost productivity and annoyed users that result from the disruption. Similarly, rebooting your desktop means losing all of your valuable state—your favorite editor with the 35 open files you were working on, your 14 terminals, and, of course, your paused game of *Frozen Bubble*.

But the alternative—not installing updates right away—is even more unpleasant. If your parents were anything like mine, they insisted that you do two things: eat your vegetables and install your software updates. Why? Well, first, vegetables provide your body with much-needed nutrients.

Second, most exploits take advantage of well-known software vulnerabilities—vulnerabilities that do not exist on patched systems. So staying up to date goes a long way in keeping your systems secure and reliable.

So is this it? Will we forever be forced to choose between security and availability? Fortunately, the answer is no. Ksplice, a startup company founded by MIT alumni, has developed technology that can install software updates, without requiring a reboot.

Using this technology, they are offering Ksplice Uptrack, a service that keeps your Linux systems up to date and secure without any hassle. Additionally, experienced kernel developers also can use the Ksplice tools to create their own rebootless updates.

Getting Started with Ksplice Uptrack

You can start using Ksplice Uptrack without any advance preparation. Follow the directions on the Ksplice Uptrack Web site, which allows you to install the software using your package manager.

Once you've done this, a K icon appears in your notification area. When you see the K, you know that you have the latest security fixes for your Linux kernel. When new updates are available, a warning sign appears over the K.

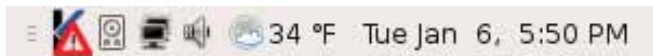


Figure 1. Ksplice Uptrack notifies you that new rebootless updates are available by displaying a K with a warning sign in the notification area.

When this happens, click on the K to view a list of the available updates. Install the updates by clicking the green Install all updates button. The listed updates will be installed on your running system in seconds, as your applications continue to run without interruption.

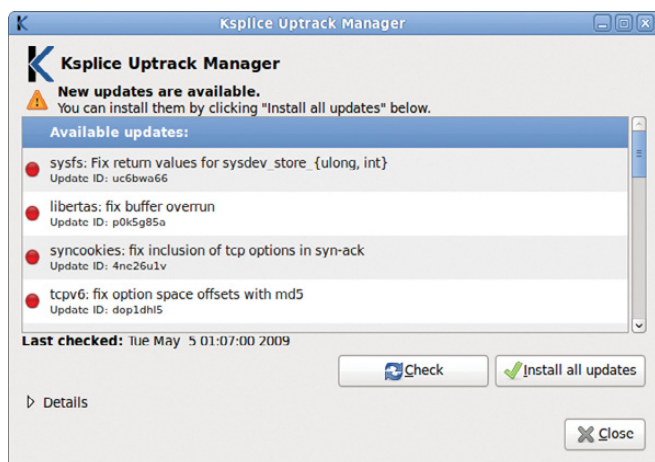


Figure 2. The Ksplice Uptrack manager shows you a list of the available kernel updates. These updates correct security and reliability problems in the kernel. You can install these updates, without disrupting your running applications, by clicking Install all updates.

Like any good Linux tool, Ksplice Uptrack also can be controlled from the command line, with four simple commands. Each update has an ID associated with it, which you use to name it. You can install or remove individual updates, just like with any package manager. Here are the Ksplice Uptrack Commands:

- `uptrack-upgrade`: downloads and installs the latest kernel updates available for your system.
- `uptrack-install id`: installs the update named `id`.
- `uptrack-remove id`: removes the update named `id`.
- `uptrack-show id`: shows more detail about the update named `id`.

What about when you actually do reboot? Well, you can boot in to your brand-new kernel that you've installed the traditional way, using your package manager. Everything will continue to work nicely, and when Ksplice Uptrack detects new updates for this kernel, it will notify you, just like before.

Alternatively, you can reboot into your old kernel. In this case, Ksplice Uptrack will re-apply the rebootless updates early in the boot process. This approach may be more desirable for some system administrators, because it ensures that the machine is in the exact same configuration both before and after the reboot.

How It Works

New research originally conducted at MIT makes this rebootless update software possible. Three basic actions are related to rebootless updates: creating a rebootless update from a source code patch, applying a rebootless update to a running system and reversing an update. I describe each of these actions below.

To follow along with these examples on your own computer, you need to install the Ksplice utilities. Your distribution likely already includes these utilities, so you can install them using your package manager. If not, you can download them from the Ksplice Web site.

Creating a Rebootless Update

To prepare a Ksplice rebootless update, you need a few ingredients. First, you need the source code of the running kernel—your Linux distribution typically makes this available through your package manager. You also need the kernel configuration file and the `System.map` file. Finally, you need to point Ksplice at your kernel headers by creating a symbolic link.

Ideally, you also would like the versions of the compiler and assembler on your system to be the same as the ones that built the original kernel. If they are too different, the Ksplice tools will notice and complain before trying to install the update. (I explain why later in this article.)

With all of the materials mentioned above, you can build a replica of your running kernel.

In these examples, I assume that the directory `/usr/src/linux` already contains the running kernel's source. The following commands prepare your setup appropriately, as described above:

```
$ mkdir /usr/src/linux/ksplice
$ cp /boot/config-`uname -r` /usr/src/linux/ksplice/.config
$ cp /boot/System.map-`uname -r` /usr/src/linux/ksplice/System.map
$ ln -s /lib/modules/`uname -r`/build /usr/src/linux/ksplice/build
```

Next, you need the patch to the kernel that you want to apply. This can be an ordinary patch taken from Linus Torvalds' git tree or a patch of your own design. Let's use an example patch that modifies the behavior of `printk`, the Linux kernel function that is responsible for printing messages to the kernel log. I assume that you have placed this patch in `~/printk.patch`:

```
--- linux-2.6/kernel/printk.c ...
+++ linux-2.6-new/kernel/printk.c ...
@@ -609,6 +609,7 @@
```

```

va_list args;
int r;

+ vprintf("Quoth the kernel:\n", NULL);
  va_start(args, fmt);
  r = vprintf(fmt, args);
  va_end(args);

```

Once this patch is applied, all messages that are printed using `printf` will be preceded by the message "Quoth the kernel:".

To create the rebootless update, run the following command from the directory `/usr/src/linux/kernel`:

```

ksplice-create --patch=~/.printk.patch /usr/src/linux

```

It should output something like `Ksplice update tarball written to ksplice-8c4o6ucj.tar.gz`. This is the rebootless update that corresponds to your source code patch.

Feeding your patch and the kernel's source code into `ksplice-create` will do the following: first, it compiles your kernel twice—once without the patch and once with the patch applied.

Second, it compares the output of the two compilations, looking for differences. In particular, it needs to find functions that have changed. For each changed function, it pulls out a copy of both the old and the new versions and puts them in the output file.

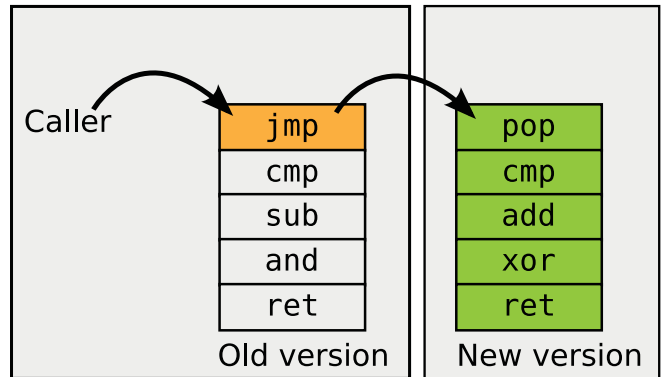


Figure 3. To replace a function, first a new version is loaded into memory. Then, at a safe time, a trampoline is inserted at the start of the old version of the function, redirecting all callers to the new code.

versions (that is, the versions that should be in memory right now) and the new versions.

First, it has to locate the functions that it's trying to change. So if it's trying to change `printf`, as in this example, it first needs to find it in kernel memory.

Once it has found it, it compares it to the old copy of `printf` that it has in the tarball. Remember that this version of `printf` was compiled from the unmodified kernel source, with the same compiler and assembler. So the two versions should match exactly. If they do not match, we act conservatively and give up. This safety check is why Ksplice requires the same compiler and assembler in the `ksplice-create` step.

Now that it has found the old copy of the function and confirmed that it is the correct code, it needs to replace it. It accomplishes this by first loading the new version of the function elsewhere in memory, using the kernel's module loader. Next, at a safe time, it overwrites the first instruction of the old function with a jump instruction that goes to the new function. This is called a trampoline, because it "bounces" all of the callers of the old function immediately over to the new function.

When is it safe to do this replacement? At a high level, we want to replace the code when no one else is using it. If the code is being used while it is being replaced, we potentially could end up with a problem. For example, if the old version of a function locked a resource in one way, and the new version locks it in another, and both run at the same time, we could end up in a situation in which they step on each other's toes.

So how does Ksplice make sure that no one is using the code while it is being replaced? It examines the stack of every kernel thread to ensure that no one has a pointer into the code that is being replaced. Said another way, if no one can reference the old code, no one is using the old code, so it's safe to replace it. This whole process takes place while the machine is briefly paused using Linux's `stop_machine` mechanism, to make sure that no new references get added when we're not looking.

If this check concludes that it is not a safe time to update the code (that is, if someone is holding a reference to the old code), `ksplice-apply` aborts the update process. Trying again is

So how does Ksplice make sure that no one is using the code while it is being replaced?

At this point, Ksplice has determined what functions have been changed by the source code patch, and it has saved old and new versions of the changed functions. Now, it must figure out how to install the new versions of the functions safely, while the system is running.

Applying a Rebootless Update

Applying the update from your perspective is quite simple. As root, run:

```

ksplice-apply ./ksplice-8c4o6ucj.tar.gz

```

from the directory `/usr/src/linux/kernel`; `ksplice-8c4o6ucj.tar.gz` is the name of the tarball created in the step above.

If the update has been applied successfully, kernel messages should appear with "Quoth the kernel" in front of them. Let's verify this by running `dmesg`, which allows us to look at the kernel's log.

If all has gone well, you will see something like:

```

# dmesg | tail -n2
Quoth the kernel:
ksplice: Update 8c4o6ucj applied successfully

```

What's happening under the hood to make this possible? Remember that Ksplice has a list of functions that need to change in the running kernel. In particular, it has the old

harmless, however, and if the update does not apply right away, it will generally apply after a few tries. This is because essentially none of the code in the kernel is *constantly* in use.

Reversing a Rebootless Update

Anything that can be applied also can be reversed, and the Ksplice tools let you do this easily. To undo an update, simply run `ksplice-undo` with the update's ID, like so:

```
# ksplice-undo 8c406ucj
```

If this process succeeds, a message will show up in the kernel's log:

```
# dmesg | tail -n1
ksplice: Update 8c406ucj reversed successfully
```

If you forget the update ID, fear not. `ksplice-view` will list the Ksplice updates currently installed on your system.

Under the hood, reversing an update is very similar to applying an update. Ksplice finds the old and new functions, and at a safe time, it *removes* the trampoline. Now, all callers of the old function will continue to get the old function. In the context of a removal, a safe time is determined as it was before, except now Ksplice makes sure that the new code is not in use. Because the tarball contains the old code, it's easy for Ksplice to determine with what code to replace the trampoline.

Determining which updates are installed is also easy. Remember that the new code gets into the kernel by being loaded as a module. As a result, it appears in the `lsmod` output, which `ksplice-view` can examine.

Pushing the Limits

Can this technology really be used to keep production machines up to date for extended periods of time? Absolutely. In fact, a Ksplice evaluation of all of the serious Linux security vulnerabilities between May 2005 and May 2008 shows that all of them can be applied as rebootless updates.

However, there is a caveat: a programmer needs to write a small amount of additional code (about 17 lines per patch, on average) for about 12% of these patches. So what sorts of patches require this additional code, and why?

Let's say that we find a bug in a kernel function that gets called only when the machine is booting and never gets called again. Let's say that this function was supposed to set a flag, but doesn't.

We can create a Ksplice update that fixes the function, but that doesn't really accomplish anything, because the function never will be called again (so the bug never will be corrected).

Instead, a kernel programmer needs to write some additional code that transforms the state of the kernel to correct the bug. In this case, the update will need to set the flag when it is applied.

However, determining whether a patch is safe to apply without additional code is tricky, as is writing the additional code. In general, patches that change initialization values or add new fields to data structures require additional code, but this is not a hard-and-fast rule. As a result, you should not construct your own Ksplice updates for use on production systems unless you are an experienced kernel developer.

That said, you still can reap the benefits of this new technology by using the Ksplice Uptrack service without having to do any of the work, because the Ksplice Uptrack folks have done it for you.

Rebootless updates represent an exciting step forward—and, with Ksplice Uptrack, Linux is the first mainstream operating system that does not require reboots for security updates, ever.

So, say goodbye to reboots, and keep working on that high score of yours. ■

Waseem Daher is a cofounder of Ksplice. He lives and works in Cambridge, Massachusetts, and can be reached at wdaher@ksplice.com.

Resources

About the Ksplice Uptrack service, including instructions for installing and getting started with the service:

www.ksplice.com/uptrack.

Sign up for the Ksplice mailing list if you're interested in hearing more: lists.ksplice.com.

A detailed technical paper on the internals of Ksplice's core technology: www.ksplice.com/paper.

Liberty Health
Software Foundation
presents:

FOSHealth 09
unconference

<http://fosshealth.eventbrite.com>

Friday July 31 to
Sunday, August 2
in Houston, T.X.

If you want to use
FOSS in healthcare,
this is the place to be.
Use the registration
code of 'lvmag' for
\$100 off registration.



libertyhsf.org

Real-Time Linux Kernel Scheduler

The `-rt` patchset, or just `-rt`, adds real-time scheduling capabilities to the Linux kernel.

ANKITA GARG

Many market sectors, such as financial trading, defense, industry automation and gaming, long have had a need for low latencies and deterministic response time. Traditionally, custom-built hardware and software were used to meet these real-time requirements. However, for some soft real-time requirements, where predictability of response times is advantageous and not mandatory, this is an expensive solution. With the advent of the `PREEMPT_RT` patchset, referred to as `-rt` henceforth, led by Ingo Molnar, Linux has made great progress in the world of real-time operating systems for “enterprise real-time” applications. A number of modifications were made to the general-purpose Linux kernel to make Linux a viable choice for real time, such as the scheduler, interrupt handling, locking mechanism and so on.

A real-time system is one that provides guaranteed system response times for events and transactions—that is, every operation is expected to be completed within a certain rigid time period. A system is classified as hard real-time if missed deadlines cause system failure and soft real-time if the system can tolerate some missed time constraints.

Design Goal

Real-time systems require that tasks be executed in a strict priority order. This necessitates that only the N highest-priority tasks be running at any given point in time, where N is the number of CPUs. A variation to this requirement could be strict priority-ordered scheduling in a given subset of CPUs or scheduling domains (explained later in this article). In both cases, when a task is runnable, the scheduler must ensure that it be put on a runqueue on which it can be run immediately—that is, the real-time scheduler has to ensure system-wide strict real-time priority scheduling (SWSRPS). Unlike non-real-time systems where the scheduler needs to look only at its runqueue of tasks to make scheduling decisions, a real-time scheduler makes global scheduling decisions, taking into account all the tasks in the system at any given point. Real-time task balancing also has to be performed frequently. Task balancing can introduce cache thrashing and contention for global data (such as runqueue locks) and can degrade throughput and scalability. A real-time task scheduler would trade off throughput in favor of correctness, but at the same time, it must ensure minimal task ping-ponging.

The standard Linux kernel provides two real-time scheduling policies, SCHED_FIFO and SCHED_RR. The main real-time policy is SCHED_FIFO. It implements a first-in, first-out scheduling algorithm. When a SCHED_FIFO task starts running, it continues to run until it voluntarily yields the processor, blocks or is preempted by a higher-priority real-time task. It has no timeslices. All other tasks of lower priority will not be scheduled until it relinquishes the CPU. Two equal-priority SCHED_FIFO tasks do not preempt each other. SCHED_RR is similar to SCHED_FIFO, except that such tasks are allotted timeslices based on their priority and run until they exhaust their timeslice. Non-real-time tasks use the SCHED_NORMAL scheduling policy (older kernels had a policy named SCHED_OTHER).

In the standard Linux kernel, real-time priorities range from zero to (MAX_RT_PRIO-1), inclusive. By default, MAX_RT_PRIO is 100. Non-real-time tasks have priorities in the range of MAX_RT_PRIO to (MAX_RT_PRIO + 40). This constitutes the nice values of SCHED_NORMAL tasks. By default, the -20 to 19 nice range maps directly onto the priority range of 100 to 139.

This article assumes that readers are aware of the basics of a task scheduler. See Resources for more information about the Linux Completely Fair Scheduler (CFS).

Overview of the -rt Patchset Scheduling Algorithm

The real-time scheduler of the -rt patchset adopts an active push-pull strategy developed by Steven Rostedt and Gregory Haskins for balancing tasks across CPUs. The scheduler has to address several scenarios:

1. Where to place a task optimally on wakeup (that is, pre-balance).
2. What to do with a lower-priority task when it wakes up but is on a runqueue running a task of higher priority.
3. What to do with a low-priority task when a higher-priority

task on the same runqueue wakes up and preempts it.

4. What to do when a task lowers its priority and thereby causes a previously lower-priority task to have the higher priority.

A push operation is initiated in cases 2 and 3 above. The push algorithm considers all the runqueues within its root domain (discussed later) to find the one that is of a lower priority than the task being pushed.

A pull operation is performed for case 4 above. Whenever a runqueue is about to schedule a task that is lower in priority than the previous one, it checks to see whether it can pull tasks of higher priority from other runqueues.

Real-time tasks are affected only by the push and pull operations. The CFS load-balancing algorithm does not handle real-time tasks at all, as it has been observed that the load balancing pulls real-time tasks away from runqueues to which they were correctly assigned, inducing unnecessary latencies.

Important -rt Patchset Scheduler Data Structures and Concepts

The main per-CPU runqueue data structure struct rq, holds a structure struct rt_rq that encapsulates information about the real-time tasks placed on the per-CPU runqueue, as shown in Listing 1.

Listing 1. struct rt_rq

```
struct rt_rq {
    struct rt_prio_array  active;
    ...
    unsigned long        rt_nr_running;
    unsigned long        rt_nr_migratory;
    unsigned long        rt_nr_uninterruptible;
    int                  highest_prio;
    int                  overloaded;
};
```

Real-time tasks have a priority in the range of 0–99. These tasks are organized on a runqueue in a priority-indexed array active, of type struct rt_prio_array. An rt_prio_array consists of an array of subqueues. There is one subqueue per priority level. Each subqueue contains the runnable real-time tasks at the corresponding priority level. There is also a bitmask corresponding to the array that is used to determine effectively the highest-priority task on the runqueue.

rt_nr_running and rt_nr_uninterruptible are counts of the number of runnable real-time tasks and the number of tasks in the TASK_UNINTERRUPTIBLE state, respectively.

rt_nr_migratory indicates the number of tasks on the runqueue that can be migrated to other runqueues. Some real-time tasks are bound to a specific CPU, such as the kernel thread softirq-timer. It is quite possible that a number of such affined threads wake up on a CPU at the same time. For example, the softirq-timer thread might cause the softirq-sched kernel thread to become active, resulting in two real-time tasks

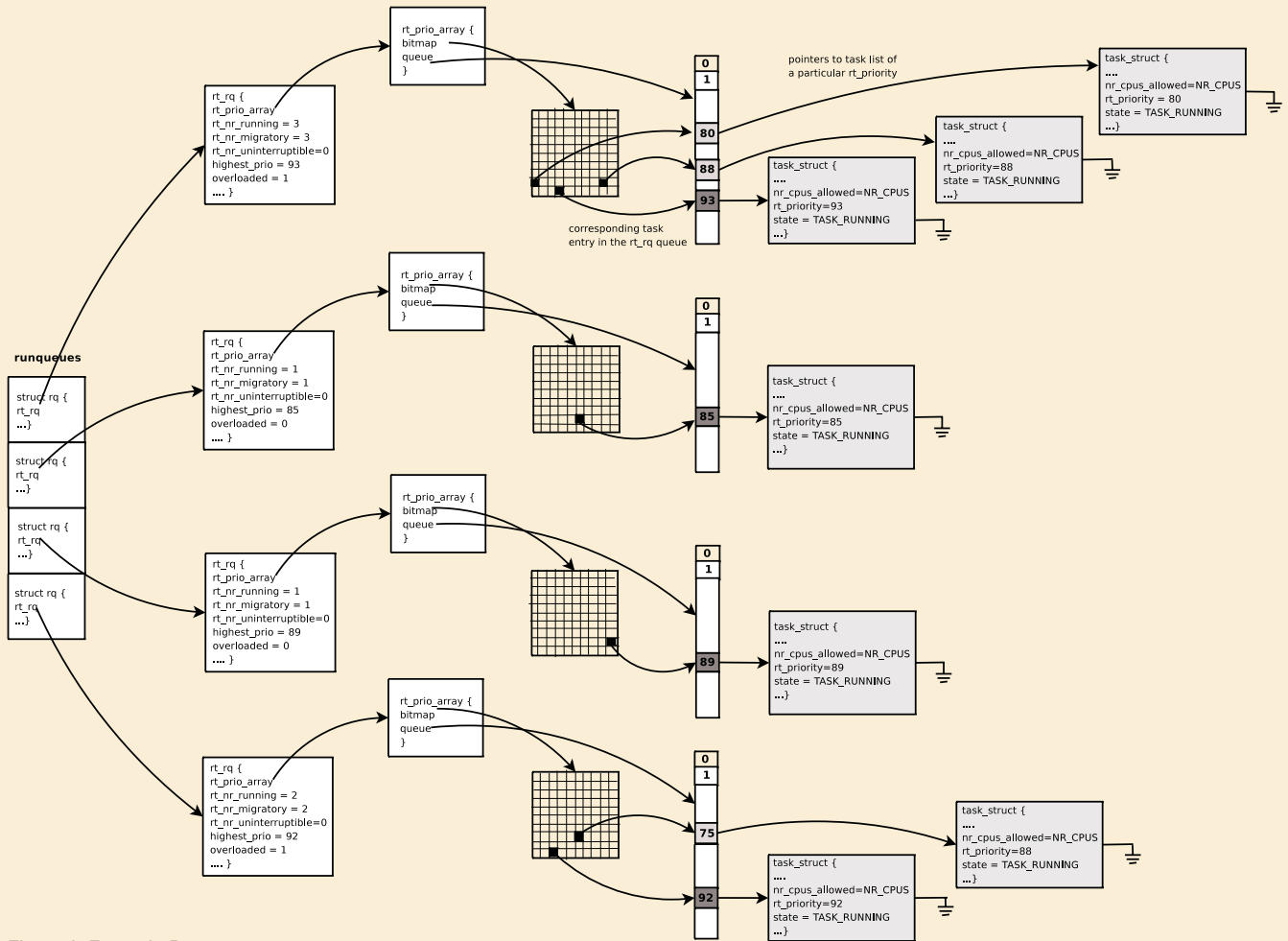


Figure 1. Example Runqueues

becoming runnable. This causes the runqueue to be overloaded with real-time tasks. When overloaded, the real-time scheduler normally will cause other CPUs to pull tasks. These tasks, however, cannot be pulled by another CPU because of their CPU affinity. The other CPUs cannot determine this without the overhead of locking several data structures. This can be avoided by maintaining a count of the number of tasks on the runqueue that can be migrated to other CPUs. When a task is added to a runqueue, the hamming weight of the task->cpus_allowed mask is looked at (cached in task->rt.nr_cpus_allowed). If the value is greater than one, the rt_nr_migratory field of the runqueue is incremented by one. The overloaded field is set when a runqueue contains more than one real-time task and at least one of them can be migrated to another runqueue. It is updated whenever a real-time task is enqueued on a runqueue.

The highest_prio field indicates the priority of the highest-priority task queued on the runqueue. This may or may not be the priority of the task currently executing on the runqueue (the highest-priority task could have just been enqueued on the runqueue and is pending a schedule). This variable is updated whenever a task is enqueued on a runqueue. The value of the highest_prio is used when scanning every runqueue to find the lowest-priority runqueue for pushing a task. If the

highest_prio of the target runqueue is smaller than the task to be pushed, the task is pushed to that runqueue.

Figure 1 shows the values of the above data structures in an example scenario.

Root Domain

As mentioned before, because the real-time scheduler requires several global, or system-wide, resources for making scheduling decisions, scalability bottlenecks appear as the number of CPUs increase (due to increased contention for the locks protecting these resources). For instance, in order to find out if the system is overloaded with real-time tasks—that is, has more runnable real-time tasks than the number of CPUs—it needs to look at the state of all the runqueues. In earlier versions, a global rt_overload variable was used to track the status of all the runqueues on a system. This variable would then be used by the scheduler on every call to the schedule() routine, thus leading to huge contention.

Recently, several enhancements were made to the scheduler to reduce the contention for such variables to improve scalability. The concept of root domains was introduced by Gregory Haskins for this purpose. cpusets provide a mechanism to partition CPUs into a subset that is used by a process or a group of processes. Several cpusets could overlap. A

Listing 2. struct root_domain

```
struct root_domain {
    atomic_t    refcount; /* reference count for the domain */
    cpumask_t   span;     /* span of member cpus of the domain*/
    cpumask_t   online;   /* number of online cpus in the domain*/
    cpumask_t   rto_mask; /* mask of overloaded cpus in the domain*/
    atomic_t    rto_count; /* number of overloaded cpus */
    ....
};
```

cpuset is called exclusive if no other cpuset contains overlapping CPUs. Each exclusive cpuset defines an isolated domain (called a root domain) of CPUs partitioned from other cpusets or CPUs. Information pertaining to every root domain is stored in struct root_domain, as shown in Listing 2. These root domains are used to narrow the scope of the global variables to per-domain variables. Whenever an exclusive cpuset is created, a new root domain object is created with information from the member CPUs. By default, a single high-level root domain is created with all CPUs as members. With the rescoping of the rt_overload variable, the cache-line bouncing would affect only the members of a particular domain and not the entire system. All real-time scheduling decisions are made only within the scope of a root domain.

CPU Priority Management

CPU Priority Management is an infrastructure also introduced by Gregory Haskins to make task migration decisions efficient. This code tracks the priority of every CPU in the system. Every CPU can be in any one of the following states: INVALID, IDLE, NORMAL, RT1, ... RT99.

CPUs in the INVALID state are not eligible for task routing. The system maintains this state with a two-dimensional bitmap: one dimension for the different priority levels and the second for the CPUs in that priority level (priority of a CPU is equivalent to the rq->rt.highest_prio). This is implemented using three arrays, as shown in Listing 3.

Listing 3. struct cpupri

```
struct cpupri {
    struct cpupri_vec  pri_to_cpu[CPUPRI_NR_PRIORITIES];
    long               pri_active[CPUPRI_NR_PRI_WORDS];
    int                 cpu_to_pri[NR_CPUS];
};
```

The pri_active bitmap tracks those priority levels that contain one or more CPUs. For example, if there is a CPU at priority 49, pri_active[49+2]=1 (real-time task priorities are mapped to 2-102 internally in order to account for priorities INVALID and IDLE), finding the first set bit of this array would yield the lowest priority that any of the CPUs in a given cpuset is in.

The field cpu_to_pri indicates the priority of a CPU.

The field pri_to_cpu yields information about all the CPUs of a cpuset that are in a particular priority level. This is

Listing 4. struct cpupri_vec

```
struct cpupri_vec {
    raw_spinlock_t  lock;
    int              count; /* number of cpus at a priority level */
    cpumask_t       mask; /* mask of cpus at a priority level */
};
```

encapsulated in struct cpupri_vec, as shown in Listing 4.

Like rt_overload, cpupri also is scoped at the root domain level. Every exclusive cpuset that comprises a root domain consists of a cpupri data value.

The CPU Priority Management infrastructure is used to find a CPU to which to push a task, as shown in Listing 5. It should be noted that no locks are taken when the search is performed.

Listing 5. Finding a CPU to Which to Push a Task

```
int cpupri_find(struct cpupri      *cp,
                struct task_struct *p,
                cpumask_t          *lowest_mask)
{
    ...
    for_each_cpupri_active(cp->pri_active, idx) {
        struct cpupri_vec *vec = &cp->pri_to_cpu[idx];
        cpumask_t mask;

        if (idx >= task_prio)
            break;

        cpus_and(mask, p->cpus_allowed, vec->mask);

        if (cpus_empty(mask))
            continue;
        *lowest_mask = mask;
        return 1;
    }
    return 0;
}
```

If a priority level is non-empty and lower than the priority of the task being pushed, the lowest_mask is set to the mask corresponding to the priority level selected. This mask is then used by the push algorithm to compute the best CPU to which to push the task, based on affinity, topology and cache characteristics.

Details of the Push Scheduling Algorithm

As discussed before, in order to ensure SWSRPS, when a low-priority real-time task gets preempted by a higher one or when a task is woken up on a runqueue that already has a higher-priority task running on it, the scheduler needs to search for a suitable target runqueue for the task. This operation of searching a runqueue and transferring one of its tasks to another runqueue is called pushing a task.

The push_rt_task() algorithm looks at the highest-priority

non-running runnable real-time task on the runqueue and considers all the runqueues to find a CPU where it can run. It searches for a runqueue that is of lower priority—that is, one where the currently running task can be preempted by the task that is being pushed. As explained previously, the CPU Priority Management infrastructure is used to find a mask of CPUs that have the lowest-priority runqueues. It is important to select only the best CPU from among all the candidates. The algorithm gives the highest priority to the CPU on which the task last executed, as it is likely to be cache-hot in that location. If that is not possible, the sched_domain map is considered to find a CPU that is logically closest to last_cpu. If this too fails, a CPU is selected at random from the mask.

The push operation is performed until a real-time task fails to be migrated or there are no more tasks to be pushed. Because the algorithm always selects the highest non-running task for pushing, the assumption is that, if it cannot migrate it, then most likely the lower real-time tasks cannot be migrated either and the search is aborted. No lock is taken when scanning for the lowest-priority runqueue. When the target runqueue is found, only the lock of that runqueue is taken, after which a check is made to verify whether it is still a candidate to which to push the task (as the target runqueue might have been modified by a parallel scheduling operation on another CPU). If not, the search is repeated for a maximum of three tries, after which it is aborted.

Details of the Pull Scheduling Algorithm

The pull_rt_task() algorithm looks at all the overloaded runqueues in a root domain and checks whether they have a real-time task that can run on the target runqueue (that is, the target CPU is in the task->cpus_allowed_mask) and is of a priority higher than the task the target runqueue is about to schedule. If so, the task is queued on the target runqueue. This search aborts only after scanning all the overloaded runqueues in the root domain. Thus, the pull operation may pull more than one task to the target runqueue. If the algorithm only selects a candidate task to be pulled in the first pass and then performs the actual pull in the second pass, there is a possibility that the selected highest-priority task is no longer a candidate (due to another parallel scheduling operation on another CPU). To avoid this race between finding the highest-priority runqueue and having that still be the highest-priority task on the runqueue when the actual pull is performed, the pull operation continues to pull tasks. In the worst case, this might lead to a number of tasks being pulled to the target runqueue, which would later get pushed off to other CPUs, leading to task bouncing. Task bouncing is known to be a rare occurrence.

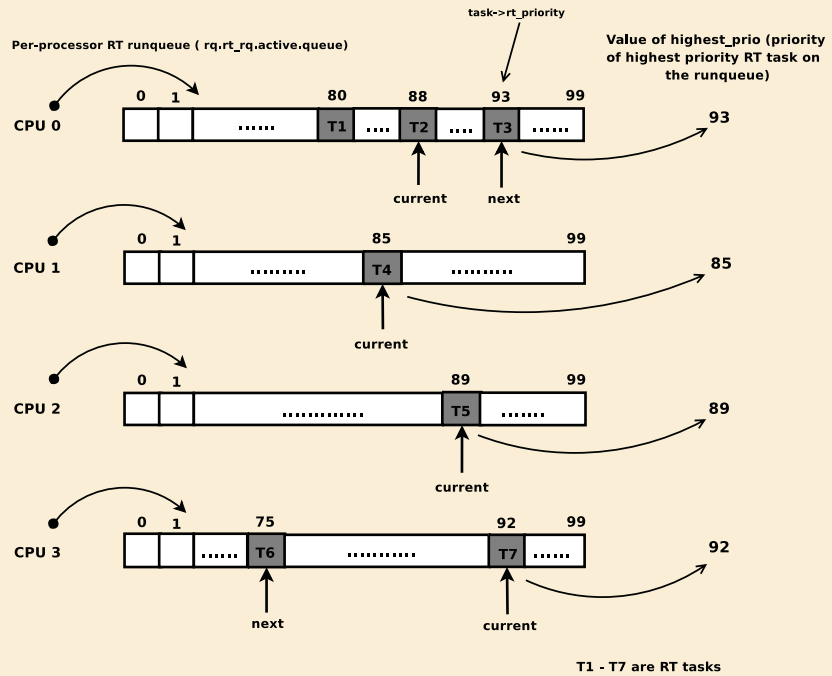


Figure 2. Runqueues Showing Currently Running Tasks and the Next Tasks to Be Run Just before the Push Operation

Scheduling Example

Consider the scenario shown in Figure 2. Task T2 is being preempted by task T3 being woken on runqueue 0. Similarly, task T7 is voluntarily yielding CPU 3 to task T6 on runqueue 3. We first consider the scheduling action on CPU 0 followed by CPU 3. Also, assume all the CPUs are in the same root domain. The pri_active bitmap for this system of CPUs will look like Figure 3. The numbers in the brackets indicate the actual priority that is offset by two (as explained earlier).

On CPU 0, the post-schedule algorithm would find the runqueue under real-time overload. It then would initiate a push operation. The first set bit of pri_active yields runqueue

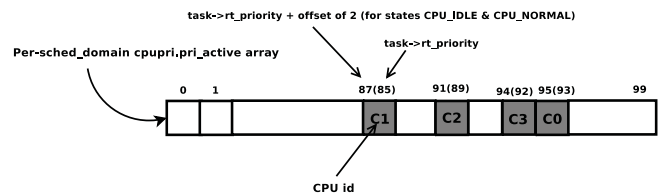


Figure 3. Per-sched Domain cpupri.pri_active Array before the Push Operation

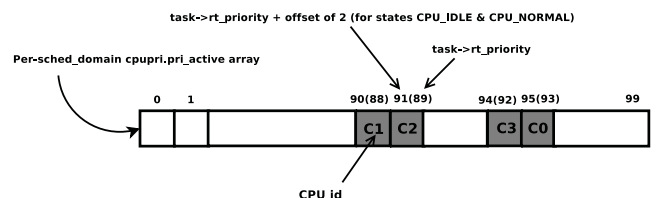


Figure 4. Per-sched Domain cpupri.pri_active Array after the Push Operation

Celebrating 15 years of *Linux Journal*,
we've brought together every article
ever published in the world's #1
Linux magazine and packaged it
in one convenient CD.



With nearly 4,000 articles written by industry experts on everything from cool projects, desktop how-tos, security, embedded systems, networking, virtualization, multimedia, system administration and programming tricks and techniques—this unique collection is a must-have for every Linux enthusiast.

Get your **NEW *Linux Journal* Archive CD** today featuring
all issues from 1994 through 2008. **Just \$34.95.**

www.linuxjournal.com/archivecd

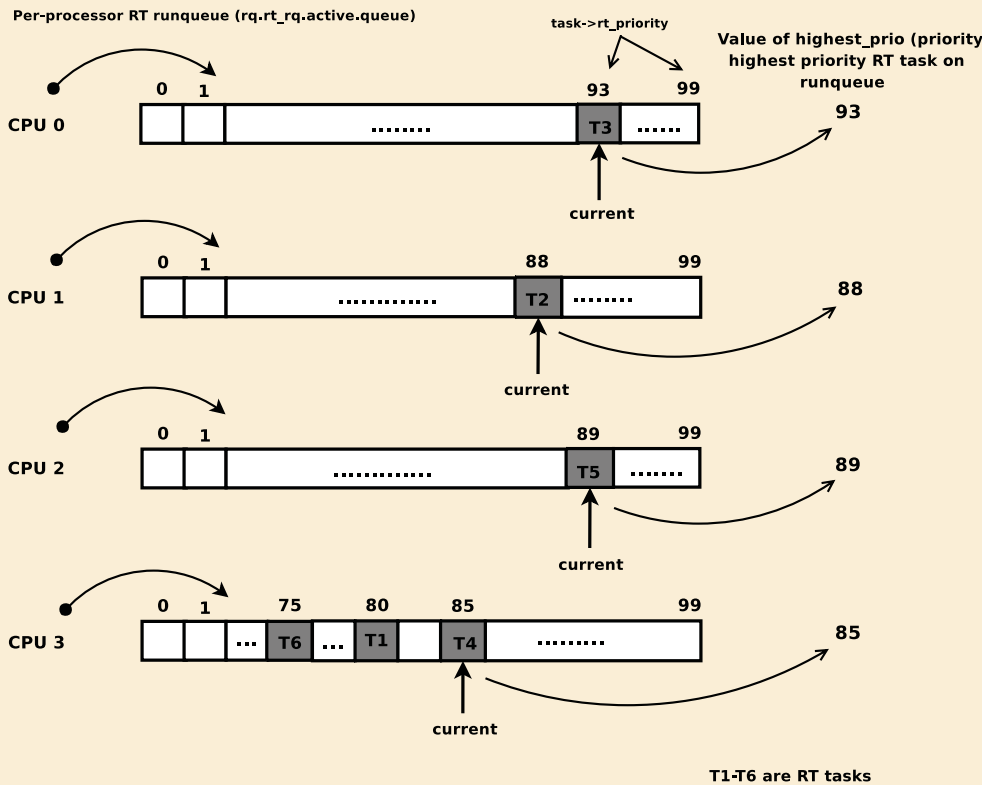


Figure 5. Runqueues after the Push and Pull Operations

of CPU 1 as the lowest-priority runqueue suitable for task T2 to be pushed to (assuming all the tasks being considered are not affined to a subset of CPUs). Once T2 is pushed over, the push algorithm then would try to push T1, because after pushing T2, the runqueue still would be under RT overload. The `pri_active` after the first operation would be as shown in Figure 4. Because the lowest-priority runqueue has a priority greater than the task to be pushed (T1 of priority 85), the push aborts.

Now, consider scheduling at CPU 3, where the current task of priority 92 is voluntarily giving up the CPU. The next task in the queue is T6. The pre-schedule routine would determine that the priority of the runqueue is being lowered, triggering the pull algorithm. Only runqueues 0 and 1 being under real-time overload would be considered by the pull routine. From runqueue 0, the next highest-priority task T1 is of priority greater than the task to be scheduled—T6, and because $T1 < T3$ and $T6 < T3$, T1 is pulled over to runqueue 3. The pull does not abort here, as runqueue 1 is still under overload, and there are chances of a higher-priority task being pulled over. The next highest task, T4 on runqueue 1, also can be pulled over, as its priority is higher than the highest priority on runqueue 3. The pull now aborts, as there are no more overloaded runqueues. The final status of all the runqueues is as shown in Figure 5, which is in accordance with scheduling requirements on real-time systems.

Although strict priority scheduling has been achieved, runqueue 3 is in an overloaded state due to the pull operation. This scenario is very rare; however, the community is working on a solution.

A number of locking-related decisions have to be made

by the scheduler. The state of the runqueues would vary from the above example, depending on when the scheduling operation is performed on the runqueues. The above example has been simplified for this explanation.

Summary

The most important goal of a real-time kernel scheduler is to ensure SWSRPS. The scheduler in the `CONFIG_PREEMPT_RT` kernel uses push and pull algorithms to balance and correctly distribute real-time tasks across the system. Both the push and pull operations try to ensure that a real-time task gets an opportunity to run as soon as possible. Also, in order to reduce the performance and scalability impact that might result from increased contention of global variables, the scheduler uses the concept of root domains and CPU priority management. The scope of the global

variables is reduced to a subset of CPUs as opposed to the entire system, resulting in significant reduction of cache penalties and performance improvement.

Legal Statement

This work represents the views of the author and does not necessarily represent the view of IBM. Linux is a copyright of Linus Torvalds. Other company, product and service names may be trademarks or service marks of others. ■

Ankita Garg, a computer science graduate from the P.E.S. Institute of Technology, works as a developer at the Linux Technology Centre, IBM India. She currently is working on the Real-Time Linux Kernel Project. You are welcome to send your comments and suggestions to ankita@in.ibm.com.

Resources

Index of /pub/linux/kernel/projects/rt (Ingo Molnar): www.kernel.org/pub/linux/kernel/projects/rt

[patch] Modular Scheduler Core and Completely Fair Scheduler [CFS] (Ingo Molnar): lwn.net/Articles/230501

Multiprocessing with the Completely Fair Scheduler, Introducing the CFS for Linux: www.ibm.com/developerworks/linux/library/l-cfs/index.html

RT Wiki: rt.wiki.kernel.org

IT is Continually Evolving, Be Sure to Keep Up.

Attend the most comprehensive IT events of the year, and gain the end-to-end views on enterprise technology that will help you keep up with the evolving needs of your data center.

➔ Complimentary events for qualified attendees!

 is now

OpenSource
world

co-located with

 **NGDC**
NEXT GENERATION DATA CENTER

and


CloudWorld

Three events. Tangible benefits. Immediate results.

From cost-effective, open source solutions and data center tools to cloud computing strategies, these events cover integrated, enterprise technologies aimed at increasing data center efficiency and reducing costs. The co-location of OpenSource World, NGDC and CloudWorld provides a unique value proposition that will maximize learning and use your time away from the office efficiently.

These events will enable you to:

- Take home solutions and best practices that will immediately increase data center efficiency, while saving on IT costs.
- Get an in-depth look at technology trends and meet face-to-face with leading solutions providers.
- Meet with peers and share case studies for data center management, open source adoption, cloud computing implementation and much more.

REGISTER NOW to Qualify for Free Attendance!

www.opensourceworld.com

Attendance is limited to IT and business professionals who meet qualifying criteria.

For sponsorship opportunities, visit www.opensourceworld.com

AN  IDG WORLD EXPO EVENT

AUGUST 12-13, 2009

MOSCONE CENTER WEST

SAN FRANCISCO, CA

www.opensourceworld.com

Making Root Unprivileged

Mitigate the damage of setuid root exploits on your system by removing root's privilege.

Serge Hallyn

There was a time when, to use a computer, you merely turned it on and were greeted by a command prompt. Nowadays, most operating systems offer a security model with multiple users.

Typically, the credentials you present at login determine the amount of privilege that programs acting upon your behalf will have. Everyday tasks can be accomplished using unprivileged userids, minimizing the risks due to user error, accidental execution of malware downloaded from the Internet and so on. Any program needing to exercise privilege must be executed using a privileged userid. In UNIX, that is userid 0, the root user. Unfortunately, this means any software needing even just a bit of privilege can lead to a complete system compromise should it misbehave or be attacked successfully.

POSIX capabilities address this problem in two ways. First, they break the notion of all-or-nothing privilege into a set of semantically distinct privileges. This can limit the amount of privilege a task may have, so that, for example, it only is able

to create devices or trace another user's tasks.

Second, the notion of privilege is separated from userids. Instead, privilege is wielded by the files (programs) a process executes. After all, users sitting at the keyboard just invoke programs to run on their behalf. It is the programs that actually do something. And, it is the programs that an administrator may entrust with privilege, based on knowledge of what the program does, who wrote it and who installed it.

POSIX capabilities have been implemented in Linux for years, but until recently, Linux supported only process capabilities. Because files are supposed to wield privilege, the lack of file capabilities meant that Linux required workarounds to allow administrators and system services to run with privilege. The POSIX capability rules were perverted to emulate a privileged root user.

Although file capabilities are now supported, the privileged root user remains the norm. In this article, I demonstrate a lazy prototype of a system with an unprivileged root.

The File Effective Set, aka the Legacy Bit

Linux has an unfortunate discrepancy between the setcap command API and what it actually does. Although setcap expects the user to define a file effective set, the kernel simply knows about a “legacy bit”. Practically speaking, if the file effective set is empty, the legacy bit is not set. If all bits in the file permitted and inherita-

ble sets are in the effective set, the legacy bit is set. If only a subset of those bits are in the effective set, setcap will return an error.

The reasoning for the setcap API command is that Linux is loathe to change userspace APIs. The reason for using the legacy bit is that we

want to encourage applications to begin with an empty effective set if they are capability-aware. Hence, the file effective set should be empty unless the application is not capability-aware. But if the application is not capability-aware, all capabilities available to it must be in its effective set from the start.

POSIX Capabilities Overview

Each process has three capability sets:

- The effective set (pE) contains the capabilities it can use right now.
- The permitted set (pP) contains those that it can add back into its effective set.
- The inheritable set (pI) is used to determine its new sets when it executes a new file.

Files also have effective (fE), permitted (fP) and inheritable (fI) sets used to calculate the new capability sets of a process executing it.

At any time, a process can use `cap_set_proc()` to remove capabilities from any of the three sets.

Capabilities can be added to the effective set only if they are currently in its permitted set. They never can be added to the permitted set. And, they can be added to the inheritable set only if they are in the permitted set or if `CAP_SETPCAP` is in pE.

When a process executes a new file, its new capability sets are calculated as follows:

- The inheritable set remains unchanged.
- The new permitted set is filled with both the file permitted set (masked with a bounding set, but for this article, I assume that always is full) and any capabilities present in both the file and process inheritable sets.
- Capabilities in the file permitted set will be available to the new process—an example use for this is the ping program. Ping needs only the capability `CAP_NET_RAW` in order to craft raw network packets. It is typically `setuid-root`, so all users can run it. By placing `CAP_NET_RAW` in ping's permitted set, all users will receive `CAP_NET_RAW` while running ping.
- Capabilities in the file inheritable set are available to a process only if they also are in the process inheritable set—an

example of this would be to allow some users to renice other users' tasks. Simply arrange (as I explain a bit) for `CAP_SYS_NICE` to be placed in their pl on login, as well as in fl for `/usr/bin/renice`. Now, ordinary users can run `renice` without privilege, and the “special” users can run `renice`, but no other programs, with `CAP_SYS_NICE`.

- The effective set is by default empty, or, if the legacy bit is set (see The File Effective Set, aka the Legacy Bit sidebar), it is set to the new permitted set.

The Privileged Root

As mentioned previously, Linux continues to emulate a privileged root user. This is done by perverting the capabilities behavior of `exec()` and `setuid()`. Briefly, if your effective userid is 0 when you execute a file, or if you execute a `setuid` root file, your permitted and effective sets are filled. If only your “real”

Unfortunately, this means any software needing even just a bit of privilege can lead to a complete system compromise should it misbehave or be attacked successfully.

userid is 0 (for instance, a root-owned process executes a `setuid-nonroot` file), only your permitted set is filled. When you change part of your userid from root to nonroot, your effective capability set is cleared. When you permanently change your userid from root to nonroot, your permitted set is cleared as well. And, if you switch your effective userid back to root, your permitted set is copied back into your effective set.

This behavior is controlled by a per-process set of securebits. One controls the `setuid()` behavior, and another controls the `exec()` behavior. They can be turned on using `prctl()`, and they can be locked such that neither the task nor its descendants can turn the bits back off.

System Preparation

In order to exploit POSIX capabilities fully, both kernel and userspace must be set up properly. The easiest and safest way to experiment with such core changes is to do so in a virtual machine. Although everything shown here could just as well be done on your native Linux installation, for simplicity, I

FEATURE Making Root Unprivileged

assume you are installing a minimal, stock Fedora system under qemu or kvm.

My first working prototype of a rootless system was done on a Gentoo system. Ubuntu Intrepid and SLES11 should come with file capabilities enabled. However, Fedora 10 wins as being the most capability-ready distribution to date, so I use it for this demonstration. To get started, download a Fedora 10 DVD from download.fedoraproject.org (call the file f10.iso), then create a qemu hard disk image and boot kvm using:

```
# qemu-img create f10.img 6G
# kvm -hda f10.img -cdrom f10.iso -m 512M -boot d
```

Then, proceed with the Fedora installation instructions. Make sure to install software development, and skip office and

In order to exploit POSIX capabilities fully, both kernel and userspace must be set up properly.

productivity tools for this image.

After rebooting, disable SELinux through the menu entries System→Administration→SELinux management. Change the top entry from Enforcing to Disabled, then reboot. Although there is no inherent reason why SELinux cannot be used with file capabilities, it does require some SELinux policy modifications.

Because you will be removing the root user's privilege, you'll want other users to receive ambient privileges at login. This is done using the pam_cap.so PAM module. To enable its use, add the line:

```
auth required pam_cap.so
```

to /etc/pam.d/system-auth. The order of these entries *does* matter, and improper order can prevent your entry from being used. I made it the second entry, after pam_env.so. Now, test by creating a user with some privilege:

```
# adduser -m netadmin
# passwd netadmin
# for f in /sbin/ifconfig /sbin/ip /sbin/route; do
#   setcap cap_net_admin=ei $f
# done
```

The above creates user netadmin and sets his password, then adds the cap_net_admin capability to the inheritable and effective sets for three network configuration programs. If you now type `ls /sbin/ifconfig`, you'll notice the entry is marked in red. This is similar to how setuid binaries, such as /bin/ping, are marked, and it's a nice touch to let you easily tell which binaries ought to be treated with extra care or to detect mistaken privilege leakage.

You also must create the /etc/security/capability.conf file, which pam_cap.so will consult on each login. The file should contain:

```
cap_net_admin netadmin
none *
```

The first line says that when user netadmin logs in, pam_cap.so should add the cap_net_admin capability to pl for its login shell. The second line, which is very important, says that everyone else (*) should receive no capabilities. Now, log in as user netadmin and play with the network:

```
hallyn@kvm# su - netadmin
netadmin@kvm# ifconfig eth0 down
```

Success! You just downed the network as a nonroot user. Now you're ready to make root unprivileged. As a first step, you will just restrict network logins over SSH. To make this as easy as possible, simply start sshd through a wrapper that sets and locks all securebits before calling the real sshd. The source for the wrapper is shown in Listing 1.

The wrapper locks itself into secure_noroot and secure_nosuidfixup mode using the prctl() system call. Then, it executes its first argument (ssh), passing the remaining arguments to the newly executed program, ssh. Compile capwrap, and copy it into /sbin:

```
# gcc -o capwrap capwrap.c -lcap
# cp capwrap /sbin/
```

Then, edit /etc/init.d/sshd to execute capwrap. Find the start() function, and place /sbin/capwrap in front of the line that actually executes sshd. That line then becomes:

```
/sbin/capwrap $SSHD $OPTIONS && success || failure
```

Of course, sshd will require some privilege to change userid and groupid among other things. Being lazy, for now, just set all capabilities using the command:

```
hallyn@kvm# setcap all=ei /usr/sbin/sshd
```

If you try restarting sshd right now, you'll be met with a silent failure. Instead, try this to start it by hand and see debugging output:

```
hallyn@kvm# /etc/init.d/sshd stop
hallyn@kvm# /sbin/capwrap /usr/sbin/sshd -Dd
```

Among other things, you'll see:

```
debug1: permanently_set_uid: 74/74
permanently_set_uid: was able to restore old [e]gid
```

sshd is complaining that it is able to restore its uid after switching to uid 74 (the ssh userid). This is problematic. Because you locked ssh into nosuid_fixup mode, switching from uid 0 to a non-0 uid does not clear out pE automatically. This means the process keeps CAP_SETUID and CAP_SETGID, so it is able to reset itsuid to 0 at any time.

The right solution is to modify the sshd source to separate the privilege handling from the userid handling. But, for this

Advertiser Index

CHECK OUT OUR NEW BUYER'S GUIDE ON-LINE.

Go to www.linuxjournal.com/buyersguide where you can learn more about our advertisers or link directly to their Web sites.

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

Listing 1. Wrapper to Execute a Program with Unprivileged Root

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/prctl.h>
#include <sys/capability.h>

int main(int argc, char * argv[])
{
    int i, ret;
    char *cmd;
    char **argvp;
    cap_t cap = cap_get_proc();
    int v[CAP_LAST_CAP+1];

    if (!cap)
        return -1;

    for (i=0; i<=CAP_LAST_CAP; i++)
        v[i] = i;

    if (cap_set_flag(cap, CAP_INHERITABLE,
                    CAP_LAST_CAP+1, v, CAP_SET))
        return -1;

    if (cap_set_proc(cap))
        return -1;

    cap_free(cap);

    ret = prctl(PR_SET_SECUREBITS, 0xf);
    if (ret) {
        perror("prctl securebits");
        exit(ret);
    }
    argvp = &argv[1];
    cmd = argvp[0];
    ret = execv(cmd, argvp);
    perror("execv");
    return ret;
}
```

experiment, let's just stop sshd from complaining! It is wrong, but perhaps not quite as bad as it seems, because when sshd executes the user's login shell, pP and pE will be recalculated anyway.

Download `opensshd_caps.patch` (see Resources), and use the following steps to apply the above patch:

```
# yum install audit-libs-devel tcp_wrappers-devel libedit-devel
# yumdownloader --source openssh
# rpm -i openssh-*.rpm
# cd /root/rpmbuild/
# rpmbuild -bc SPECS/openssh*
# cd BUILD/openssh-*/
# patch < /usr/src/opensshd_caps.patch
# make && make install
```

Advertiser	Page #	Advertiser	Page #
1&1 INTERNET, INC. www.oneandone.com	1	MICROWAY, INC. www.microway.com	C4, 13
ABERDEEN, LLC www.aberdeeninc.com	3	OPENSOURCE WORLD www.opensourceworld.com/live/12/	61
ASA COMPUTERS, INC. www.asacomputers.com	35	POLYWELL COMPUTERS, INC. www.polywell.com	5
CARINET www.cari.net	43	RACKSPACE MANAGED HOSTING www.rackspace.com	C3
CORRID, INC. www.corraid.com	7, 79	SAINT ARNOLD BREWING COMPANY www.saintarnold.com	79
DIGI-KEY CORPORATION www.digi-key.com	79	SERVERBEACH www.serverbeach.com	49
EMAC, INC. www.emacinc.com	36, 79	SERVERS DIRECT www.serversdirect.com	9
EMPERORLINUX www.emperorlinux.com	31	SILICON MECHANICS www.siliconmechanics.com	25, 33
GECAD TECHNOLOGIES/AXIGEN www.axigen.com	79	SYNSEER fosshealth.eventbrite.com	53, 79
GENSTOR SYSTEMS, INC. www.genstor.com	27	TECHNOLOGIC SYSTEMS www.embeddedx86.com	69
LINUX FOUNDATION www.linuxfoundation.org	45	UBIQUIT NETWORKS, INC. www.ubnt.com	C2
LINUX ON WALL STREET www.linuxonwallstreet.com	75	UTILIKILTS www.utilikilts.com	79
LOGIC SUPPLY, INC. www.logicsupply.com	73	IXSYSTEMS, INC. www.ixsystems.com	29
LULLABOT www.lullabot.com	23, 77		

ATTENTION ADVERTISERS

November 2009 Issue #187 Deadlines

Space Close: August 24; Material Close: September 1

Theme: Infrastructure

BONUS DISTRIBUTIONS:

Supercomputing, Vision Embedded Linux Developers Conference, USENIX LISA, SD Best Practices East

Call Joseph Krack to reserve your space
+1-713-344-1956 ext. 118, e-mail joseph@linuxjournal.com

FEATURE Making Root Unprivileged

```
# setcap all=ei /usr/sbin/sshd
# /etc/init.d/sshd start
```

Now ssh in as root, and use capsh to print your capability status:

```
root@kvm# /sbin/capsh --print
Current: =
Bounding set =(full set of capabilities)
Securebits: 057/0x2f
secure-noroot: yes (locked)
secure-no-suid-fixup: yes (locked)
secure-keep-caps: no (unlocked)
uid=0
```

SSH logins are locked in secure-noroot and secure-nosuid-fixup.

Setting Up Administrative Users

The root userid now carries no privileges, but the system still requires administration. That requires privilege. So, let's define several partially privileged users. At login, each will receive inheritable capabilities sufficient to achieve some task. Working out the most useful combinations of capabilities to assign to select users is an interesting exercise, but for now let's focus on three users: netadmin, which can change network settings; useradmin, which can add and delete users, kill their processes and modify their files; and privadmin, which can change file capabilities and users' inheritable capabilities.

Create the users:

```
# adduser -m privadmin
# passwd privadmin
# adduser -m useradmin
# passwd useradmin
# chown privadmin /etc/security/capability.conf
```

The new capability.conf file follows:

```
cap_net_admin netadmin
cap_chown,cap_dac_override,cap_fowner,cap_kill useradmin
cap_setfcap privadmin
none *
```

privadmin may set file capabilities (cap_setfcap), so make him the owner of the capabilities.conf file, so he can set pl for users. useradmin can manipulate other users' files and processes. netadmin remains unchanged. (Note, privadmin can give himself whatever privilege he wants. A good audit policy and a limited tool for editing capability.conf would help mitigate that risk.)

You also need to set some inheritable file capabilities on system administration utilities to grant these users privilege. Listing 2 shows a small list to get started. For brevity, let's just assign all capabilities to the inheritable set. You can apply these using the script in Listing 3 using sh loopcaps.sh admincaplist. Finally, you'll need to let useradmin execute useradd using chmod o+x /usr/sbin/useradd.

Now, log in as each of these users and play around.

There are still a few problems though. For instance, log in as useradmin and try to change someone's password:

```
useradmin@kvm# passwd netadmin
passwd: Only root can specify a user name.
```

That's no good! The passwd program has noticed that you are not root and won't let you change another user's password. We are finding more and more code, written to accommodate the subtleties of different operating systems, which would now need to be further complicated to support our unprivileged-root model.

You can work around this case for now in one of two easy ways. First, you simply can use the root user instead of useradmin. The root user still will not carry privileges unless it executes a (trusted) file with file capabilities. Second, you can continue to use the useradmin user name, but give it userid 0. Go ahead and try that. Edit /etc/passwd.conf, find the entry for useradmin, and change the first numeric column to 0. Then chown -R 0 /home/useradmin, so that he still can access his home directory. Now, you can log out and back in, and passwd will succeed. Actually, it ends with an error message, but you'll find that you did actually succeed in changing the password.

Listing 2. File Capabilities to Empower Partially Privileged Admins

```
/bin/kill:=ei
/bin/ls:=ei
/bin/cat:=ei
/bin/ls:=ei
/bin/mv:=ei
/bin/touch:=ei
/bin/mount:=ei
/bin/umount:=ei
/bin/vi:=ei
/bin/rm:=ei
/bin/chgrp:=ei
/bin/find:=ei
/bin/chmod:=ei
/bin/chown:=ei
/bin/mkdir:=ei
/usr/sbin/useradd:=ei
/usr/bin/passwd:=ei
/usr/sbin/setcap:cap_setfcap=ei
/bin/ping:cap_net_raw=ep
/bin/su:=ep
```

Listing 3. Script to Apply File Capabilities

```
#!/bin/sh
for l in `cat $1`; do
    fglob=`echo $l | awk -F: '{ print $1 }'`
    p=`echo $l | awk -F: '{ print $2 }'`

    for f in `bin/ls $fglob`; do
        setcap $p $f
    done
done
```

Locking Down init

Now that you have some partially privileged administrative users, let's put the whole system in unprivileged-root mode. You could do this by patching the kernel, but in this case, let's patch init to use `prctl()` the same way that `capwrap` did. A patch to `upstart`, the Fedora init program, can be found in the Resources. You can apply it using steps similar to the `openssh` steps:

```
# yumdownloader --source upstart
# rpm -i upstart*.rpm
# cd rpmbuild
# rpmbuild -bc SPECS/upstart.spec
# cd BUILD/upstart*
# patch -p1 < /usr/src/upstart.patch
# cp /sbin/init /sbin/init.orig
# make && make install
```

Also, in case something goes wrong, edit `/boot/grub/grub.conf`, comment out `hiddenmenu`, and set `timeout` to 10 instead of 0. Now if something goes wrong, you can interrupt the boot process and add `init=/sbin/init.orig` to the end of your boot line.

The patched init enables all capabilities in its inheritable set. It also keeps its permitted and effective sets filled, although you should be able to drop many from its permitted set, keeping its effective set empty for most of its run. You will need to add file capabilities to many of the programs used during system startup. Ideally, you would modify each of these programs so as to avoid setting the legacy bit, but again this is a lazy proof of concept. Listing 4 contains a list that is sufficient for boot to succeed on the F10 image.

You can apply these with the same script as before (Listing 3).

You'll also need to execute:

```
chmod go-x /usr/sbin/console-kit-daemon
```

You're giving it forced rather than inherited permissions in lieu of changing the (`setuid=root`) `dbus-daemon-launch-helper` code so as to fill its inheritable set. This means any user would receive full privilege when executing it, so you allow only root to execute it.

Conclusion

This article demonstrates taking a stock Fedora 10 system and changing the privilege system from one where one `userid` (`root`'s) automatically imparts privilege, to one where only file capabilities determine the privilege available to a caller. The root user turns from a privileged user to simply the `userid` that happens to own most system resources.

You can remove the privileged root user for a whole system. In this experiment, quite a bit of work still needs to be done to make that practical, say, for a whole distribution. Most important, legacy code makes assumptions based on `userids`. Setting up partially privileged users make system administration convenient, while making the privilege separation useful will be an interesting project.

In the meantime, you can exploit the per-process nature of

Listing 4. Capabilities Needed to Boot Fedora with Unprivileged Root

```
/sbin/fsck:=ei
/sbin/udev:=ei
/sbin/shutdown:=ei
/sbin/e2fsck:=ei
/sbin/mingetty:cap_chown,cap_dac_override,cap_sys_tty_config+ei
/sbin/dhclient:=ei
/sbin/reboot:=ei
/sbin/fsck.ext3:=ei
/sbin/hwclock:cap_sys_time=ei
/bin/setfont:cap_sys_admin,cap_sys_resource,cap_sys_tty_config=ei
/bin/hostname:cap_sys_admin=ei
/bin/loadkeys:cap_sys_admin,cap_sys_resource,cap_sys_tty_config=ei
/usr/bin/stat:cap_dac_override,cap_dac_read_search=ei
/sbin/rsyslogd:cap_sys_admin,cap_audit_write=ei
/bin/login:all=ei
/sbin/MAKEDEV:=ei
/sbin/auditd:=ei
/sbin/auditctl:=ei
/sbin/microcode_ctl:=ei
/usr/bin/hal-*=ei
/usr/sbin/hald:=ei
/usr/libexec/hal*=ei
/sbin/insmod:=ei
/sbin/modprobe:=ei
/sbin/rmmod:=ei
/bin/plymouth:=ei
/usr/bin/Xorg:=ei
/usr/sbin/gdm-binary:=ei
/bin/dbus-daemon:=ei
/usr/sbin/avahi-daemon:=ei
/usr/bin/ssh-agent:=ei
/sbin/pam_console_apply:=ei
/usr/sbin/gpm:=ei
/lib/dbus-1/dbus-daemon-launch-helper:=ep
/sbin/initctl:=ei
/usr/sbin/console-kit-daemon:=ep
/usr/sbin/NetworkManager:=ei
/usr/libexec/gdm*=ei
/usr/sbin/gdm-binary:=ei
```

the unprivileged-root mode. This article shows how to remove the privileged root user from any legacy software that always is intended to be unprivileged. You also should design new services to be capability-aware so that they too can run without a privileged root. Doing so can greatly reduce the impact of any bugs or exploits. ■

Serge Hallyn does Linux kernel and security coding with the IBM Linux Technology Center, mostly working with containers, application migration, POSIX capabilities and SELinux.

Resources

`opensshd_caps.patch` and `upstart.patch` are available at ftp.linuxjournal.com/pub/lj/listings/issue184/10249.tgz.

Completely Fair SCHEDULER

Find out how Linux's new scheduler strives to be fair to all processes and eliminate the problems with the old O(1) scheduler.

CHANDANDEEP SINGH PABLA

Most modern operating systems are designed to try to extract optimal performance from underlying hardware resources. This is achieved mainly by virtualization of the two main hardware resources: CPU and memory. Modern operating systems provide a multitasking environment that essentially gives each task its own virtual CPU. The task generally is unaware of the fact that it does not have exclusive use of the CPU.

Similarly, memory virtualization is achieved by giving each task its own virtual memory address space, which is then mapped onto the real memory of the system. Again, the task generally is unaware of the fact that its virtual memory addresses may not map to the same physical address in real memory.

CPU virtualization is achieved by “sharing” the CPU between multiple tasks—that is, each running task gets a small fraction of the CPU at regular intervals. The algorithm used to select one task at a time from the multiple available runnable tasks is called the scheduler, and the process of selecting the next task is called scheduling.

The scheduler is one of the most important components of any OS. Implementing a scheduling algorithm is difficult for a couple reasons. First, an acceptable algorithm has to allocate CPU time such that higher-priority tasks (for example,

interactive applications like a Web browser) are given preference over low-priority tasks (for example, non-interactive batch processes like program compilation). At the same time, the scheduler must protect against low-priority process starvation. In other words, low-priority processes must be allowed to run eventually, regardless of how many high-priority processes are vying for CPU time. Schedulers also must be crafted carefully, so that processes appear to be running simultaneously without having too large an impact on system throughput.

For interactive processes like GUIs, the ideal scheduler would give each process a very small amount of time on the CPU and rapidly cycle between processes. Because users expect interactive processes to respond to input immediately, the delay between user input and process execution ideally should be imperceptible to humans—somewhere between 50 and 150ms at most.

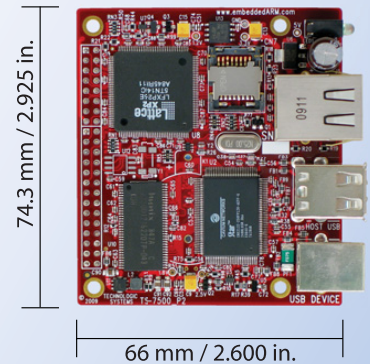
For non-interactive processes, the situation is reversed. Switching between processes, or context switching, is a relatively expensive operation. Thus, larger slices of time on the processor and fewer context switches can improve system performance and throughput. The scheduling algorithm must strike a balance between all of these competing needs.

Like most modern operating systems, Linux is a multitasking operating system, and therefore, it has a scheduler. The Linux scheduler has evolved over time.

TS-7500 Embedded Computer

Faster. Smaller. Cheaper.

Qu. 100 **\$84**



Powered by a
250 MHz ARM9 CPU

- Low power, fanless, < 2 watts
- 64MB DDR-RAM
- 4MB NOR Flash
- Micro-SD Card slot - SDHC
- USB 2.0 480Mbit/s host (2) slave (1)
- 10/100 Ethernet
- Boots Linux in less than 2 seconds
- Customizable FPGA - 5K LUT
- Power-over-Ethernet ready
- Optional battery backed RTC
- Watchdog Timer
- 8 TTL UART
- 33 DIO, SPI, I²C

Dev Kit provides out-of-box development + extra features

- Over 20 years in business
- Never discontinued a product
- Engineers on Tech Support
- Open Source Vision
- Custom configurations and designs w/ excellent pricing and turn-around time
- Most products ship next day



We use our stuff.
visit our TS-7800 powered website at
www.embeddedARM.com
(480) 837-5200

O(1) Scheduler

The Linux scheduler was overhauled completely with the release of kernel 2.6. This new scheduler is called the O(1) scheduler—O(...) is referred to as “big O notation”. The name was chosen because the scheduler’s algorithm required constant time to make a scheduling decision, regardless of the number of tasks. The algorithm used by the O(1) scheduler relies on active and expired arrays of processes to achieve constant scheduling time. Each process is given a fixed time quantum, after which it is preempted and moved to the expired array. Once all the tasks from the active array have exhausted their time quantum and have been moved to the expired array, an array switch takes place. This switch makes the active array the new empty expired array, while the expired array becomes the active array.

The main issue with this algorithm is the complex heuristics used to mark a task as interactive or non-interactive. The algorithm tries to identify interactive processes by analyzing average sleep time (the amount of time the process spends waiting for input). Processes that sleep for long periods of time probably are waiting for user input, so the scheduler assumes they’re interactive. The scheduler gives a priority bonus to interactive tasks (for better throughput) while penalizing non-interactive tasks by lowering their priorities. All the calculations to determine the interactivity of tasks are complex and subject to potential miscalculations, causing non-interactive behavior from an interactive process.

As I explain later in this article, CFS

is free from any such calculations and just tries to be “fair” to every task running in the system.

Completely Fair Scheduler

According to Ingo Molnar, the author of the CFS, its core design can be summed up in single sentence: “CFS basically models an ‘ideal, precise multitasking CPU’ on real hardware.”

Let’s try to understand what “ideal, precise, multitasking CPU” means, as the CFS tries to emulate this CPU. An “ideal, precise, multitasking CPU” is a hardware CPU that can run multiple processes at the same time (in parallel), giving each process an equal share of processor power (not time, but power). If a single process is running, it would receive 100% of the processor’s power. With two processes, each would have exactly 50% of the physical power (in parallel). Similarly, with four processes running, each would get precisely 25% of physical CPU power in parallel and so on. Therefore, this CPU would be “fair” to all the tasks running in the system (Figure 1).

Obviously, this ideal CPU is nonexistent, but the CFS tries to emulate such a processor in software. On an actual real-world processor, only one task can be allocated to a CPU at a particular time. Therefore, all other tasks wait during this period. So, while the currently running task gets 100% of the CPU power, all other tasks get 0% of the CPU power. This is obviously not fair (Figure 2).

The CFS tries to eliminate this unfairness from the system. The CFS tries to

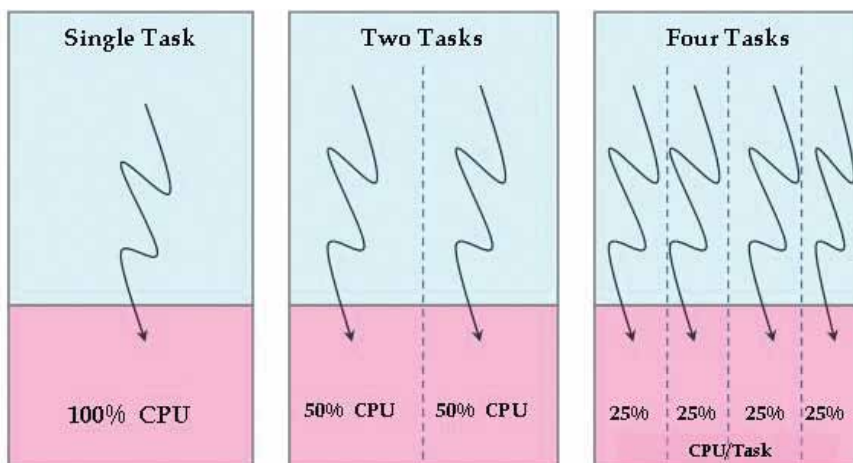


Figure 1. Ideal, Precise, Multitasking CPU

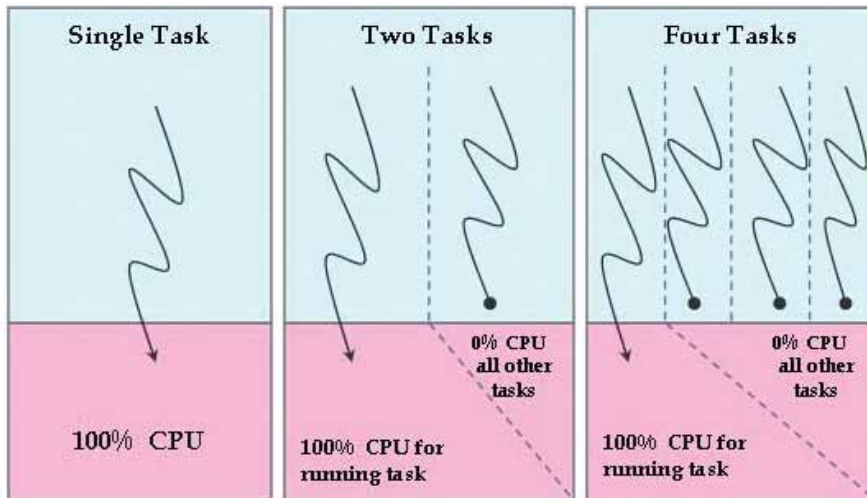


Figure 2. Actual Hardware CPU

keep track of the fair share of the CPU that would have been available to each process in the system. So, CFS runs a fair clock at a fraction of real CPU clock speed. The fair clock's rate of increase is calculated by dividing the wall time (in nanoseconds) by the total number of processes waiting. The resulting value is the amount of CPU time to which each process is entitled.

As a process waits for the CPU, the scheduler tracks the amount of time it would have used on the ideal processor. This wait time, represented by the per-task `wait_runtime` variable, is used to rank processes for scheduling and to determine the amount of time the process is allowed to execute before being preempted. The process with the longest wait time (that is, with the gravest need of CPU) is picked by the scheduler and assigned to the CPU. When this process is running, its wait time decreases, while the time of other waiting tasks increases (as they were waiting). This essentially means that after some time, there will be another task with the largest wait time (in gravest need of the CPU), and the currently running task will be preempted. Using this principle, CFS tries to be fair to all tasks and always tries to have a system with zero wait time for each process—each process has an equal share of the CPU (something an “ideal, precise, multitasking CPU” would have done).

Kernel 2.6.23

In order for the CFS to emulate an “ideal, precise, multitasking CPU” by

giving each runnable process an equal slice of execution time, CFS needs to have the following:

1. A mechanism to calculate what the fair CPU share is per process. This is achieved by using a system-wide runqueue `fair_clock` variable (`cfs_rq->fair_clock`). This fair clock runs at a fraction of real time, so that it runs at the ideal pace for a single task when there are N runnable tasks in the system. For example, if you have four runnable tasks, `fair_clock` increases at one-fourth of the speed of wall time (which means 25% fair CPU power).
2. A mechanism to keep track of the time for which each process was waiting while the CPU was assigned to the currently running task. This wait time is accumulated in the per-process variable `wait_runtime` (`process->wait_runtime`).

CFS uses the fair clock and wait runtime to keep all the runnable tasks sorted by the `rq->fair_clock - p->wait_runtime` key in the rbtree (see the Red-Black Tree sidebar). So, the leftmost task in the tree is the one with the “gravest CPU need”, and CFS picks the leftmost task and sticks to it. As the system progresses forward, newly awakened tasks are put into the tree farther and farther to the right—slowly but surely giving every task a chance to become the leftmost task and, thus, get on the CPU within a deterministic

Red-Black Tree (RBTree)

A red-black tree is a type of self-balancing binary search tree—a data structure typically used to implement associative arrays. It is complex, but it has good worst-case running time for its operations and is efficient in practice. It can search, insert and delete in $O(\log n)$ time, where n is the number of elements in the tree. In red-black trees, the leaf nodes are not relevant and do not contain data. These leaves need not be explicit in computer memory—a null child pointer can encode the fact that this child is a leaf—but it simplifies some algorithms for operating on red-black trees if the leaves really are explicit nodes. To save memory, sometimes a single sentinel node performs the role of all leaf nodes; all references from internal nodes to leaf nodes then point to the sentinel node. (Source: Wikipedia.)

amount of time.

Because of this simple design, CFS no longer uses active and expired arrays and dispensed with sophisticated heuristics to mark tasks as interactive versus non-interactive.

CFS implements priorities by using weighted tasks—each task is assigned a weight based on its static priority. So, while running, the task with lower weight (lower-priority) will see time elapse at a faster rate than that of a higher-priority task. This means its `wait_runtime` will exhaust more quickly than that of a higher-priority task, so lower-priority tasks will get less CPU time compared to higher-priority tasks.

Kernel 2.6.24

CFS has been modified a bit further in 2.6.24. Although the basic concept of fairness remains, a few implementation

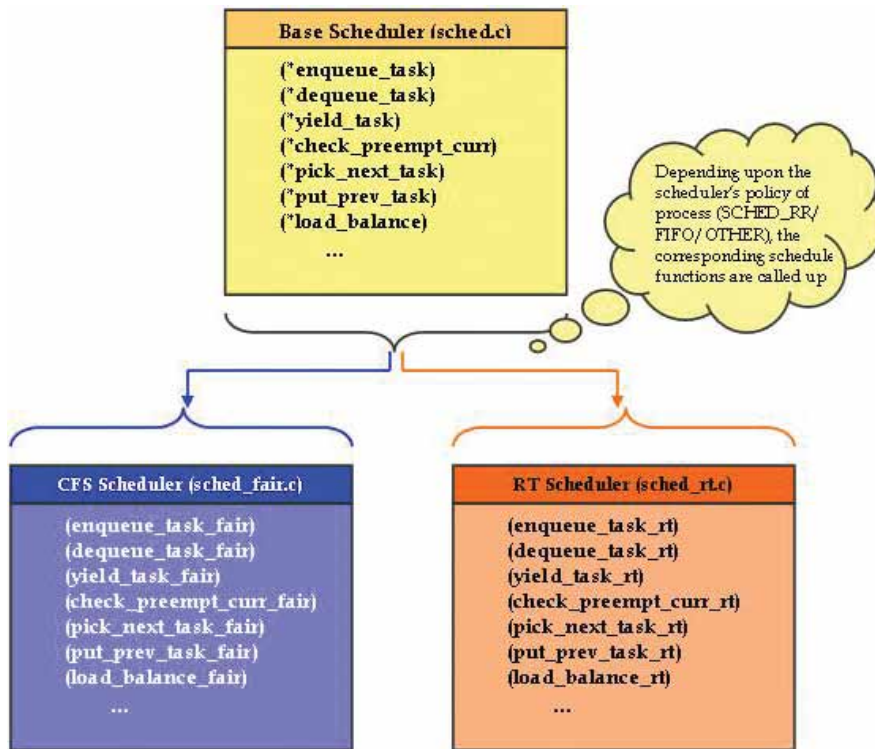


Figure 3. Modular Scheduler

details have changed. Now, instead of chasing the global fair clock (`rq->fair_clock`), tasks chase each other. A clock per task, `vruntime`, is introduced, and an approximated average is used to initialize this clock for new tasks. Each task tracks its runtime and is queued in the RBTree using this parameter. So, the task that has run least (the one that has the gravest CPU need) is the leftmost node of the RBTree and will be picked up by the scheduler. See Resources for more details about this implementation.

In kernel 2.6.24, another major addition to CFS is group scheduling. Plain CFS tries to be fair to all the tasks running in the system. For example, let's say there is a total of 25 runnable processes in the system. CFS tries to be fair by allocating 4% of the CPU to all of them. However, let's say that out of these 25 processes, 20 belong to user A while 5 belong to user B. User B is at an inherent disadvantage, as A is getting more CPU power than B. Group scheduling tries to eliminate this problem. It first tries to be fair to a group and then to individual tasks within that group. So CFS, with group scheduling

enabled, will allocate 50% of the CPU to each user A and B. The allocated 50% share of A will be divided fairly among A's 20 tasks, while the other 50% of the CPU time will be distributed fairly among B's 5 tasks.

Scheduling Classes/Modular Scheduler

With kernel 2.6.23, the Linux scheduler also has been made modular. Each scheduling policy (`SCHED_FIFO`, `SCHED_RR`, `SCHED_OTHER` and so on) can be implemented independently of the base scheduler code. This modularization is similar to object-oriented class hierarchies (Figure 3).

The core scheduler does not need to be aware of the implementation details of the individual scheduling policies. In kernel 2.6.23, `sched.c` (the "scheduler" from older kernels) is divided into the following files to make the scheduler modular:

- `kernel/sched.c`: contains the code of a generic scheduler, thereby exposing functions like `sched()`. The specific

scheduling policy is implemented in a different file.

- `kernel/sched_fair.c`: this is the main file that implements the CFS scheduler and provides the `SCHED_NORMAL`, `SCHED_BATCH` and `SCHED_IDLE` scheduling policies.
- `kernel/sched_rt.c`: provides the `SCHED_RR` and `SCHED_FIFO` policies used by real-time (RT) threads.

Each of these scheduling policies (fair and RT) registers its function pointers with the core scheduler. The core scheduler calls the appropriate scheduler (fair or RT), based on the scheduling policy of the particular process. As with the $O(1)$ scheduler, real-time processes will have higher priority than normal processes. CFS mainly addresses non-real-time processes, and the RT scheduler remains more or less the same as before (except for a few changes as to how non-active/expired arrays are maintained).

With this new modular scheduler in place, people who want to write new schedulers for a particular policy can do so by simply registering these new policy functions with the core scheduler.

Summary

The CFS design is quite radical and innovative in its approach. Features like the modular scheduler ease the task of integrating new scheduler types to the core scheduler. ■

Chandandeep Singh Pabla works at STMicroelectronics. He has extensive experience in the development of embedded software for multimedia (DVD/STB) chipsets on multiple operating systems. He can be reached at chandandeep.pabla@st.com.

Resources

Linux Kernel Source Code (2.6.23/2.6.24): `sched-design-CFS.txt` by Ingo Molnar

IBM Developer Works Article Multiprocessing with the Completely Fair Scheduler: kerneltrap.org/node/8059

Blog Post Related to CFS: immike.net/blog/2007/08/01/what-is-the-completely-fair-scheduler

Anthony Lineberry on /dev/mem Rootkits

Rootkits using /dev/mem could attack your system and leave virtually no trace—it even could be happening now! MICK BAUER

At Black Hat Europe in mid-April 2009, Anthony Lineberry presented an interesting paper on how attackers with root privileges might use a /dev/mem rootkit, hiding their attacks by directly altering kernel memory. Although not a completely new technique, Anthony's BHE presentation put it back in the spotlight. In addition, Lineberry described proof-of-concept tools he's developing to demonstrate how this technique could be exploited in the real world.

On the one hand, once attackers have gained root privileges on your system, it's game over—the attackers have complete control, and all hope for further defense and mitigation on your part is gone. Looked at from that viewpoint, the attackers' ability to write directly to kernel memory isn't too radically different from, or worse than, other things they can do as root.

But, on the other hand, even if your system suffers root compromise, you still want *some* chance of at least *detecting* the compromise in order to do something about it. Because the purpose of rootkits is to prevent that, it behooves you

But, on the other hand, even if your system suffers root compromise, you still want *some* chance of at least *detecting* the compromise in order to do something about it.

to take whatever precautions you can against them. So in this sense, new rootkit techniques actually are very worthy of our attention and concern.

In this article, I provide some background on rootkits and /dev/mem, and Anthony Lineberry sheds further light on /dev/mem rootkits, in the form of a conversation we recently had.

Rootkit Refresher

So, what exactly is a rootkit? Simply put, a rootkit is hostile code that conceals or misrepresents a system's state, as presented to its administrator.

The "kit" part of the term reflects the fact that early UNIX rootkits took the form of collections of one-for-one replacements of system commands, such as ls and ps. The replacement commands behaved, for the most part, like the commands they replaced, except they were selectively blind. A rootkit's ls

command, for example, might omit the attacker's directory /...my_evil_tools in file listings it displays, and a rootkit's ps command might omit the attacker's program erase_recent_logs from process listings. In other words, rootkits are designed to conceal the activities of system attackers once they've achieved a foothold on a target system.

One problem with first-generation rootkits was that their functionality was limited to those specific commands replaced by rootkit versions. What if the system administrator used some command or utility rather than ls to view the contents of a directory containing attack evidence?

Another problem was detectability. If a system is protected with system integrity software like Tripwire, which detects and reports on authorized changes to system files, it can be difficult to replace system commands without being detected.

Both these problems were largely "solved" with the advent of Loadable Kernel Module (LKM) rootkits. An LKM rootkit, as the name implies, consists of one or more kernel modules loaded by attacks. An LKM rootkit re-maps the actual *system calls* (also known as kernel symbols) accessed by system utilities, leaving the system commands themselves unchanged. Needless to say, this is a very powerful technique.

As powerful as LKM rootkits still are, they nonetheless can be detected, for example, by comparing the kernel's system map (a file showing the correct memory addresses of all supported system calls) with the actual system call addresses in memory. On a non-LKM-infested system, those addresses should be the same as in the system map.

/dev/mem and /dev/kmem

That, then, is the problem space in which rootkits operate—concealing attack activity and results in a way that is not itself conspicuous. But, what is /dev/mem, and how is this particular kernel interface different from an LKM?

/dev/mem is a character device that provides root-privileged processes in userspace (that is, programs other than the kernel or kernel modules) direct access to physical memory. /dev/kmem is the same thing, but it uses "virtual" memory addresses like the kernel uses rather than the "raw" addresses of physical memory. Unlike /proc/kcore, which serves a similar function to developers and kernel hackers, /dev/mem and /dev/kmem grant not only read access, but also write access to memory.

You might be forgiven for assuming that, like /dev/eth0, /dev/hda and other special files in /dev, /dev/mem is an essential interface for userspace applications that need to communicate

with the kernel. As it happens, this isn't necessarily the case. Besides kernel developers, historically, the other major user of `/dev/mem` is the X Window System, parts of which still use `/dev/mem` to access video adapters' memory and control registers.

At least in the case of `/dev/kmem`, some people think these particular devices are of greater use to attackers than for more legitimate purposes. As far back as 2005, Jonathan Corbet of *lwn.net* said, "It has been suggested that rootkits are the largest user community for this kind of access" (see Resources for the full context; he was speaking specifically of `/dev/kmem`).

Hopefully, I'm not overstating this case, because being neither a kernel developer nor an X Windows System expert, I would not presume to argue for abolishing `/dev/mem` or `/dev/kmem` myself. Rather, I'm trying to put all of this into a useful context—which brings us to Anthony Lineberry.

The Interview

Anthony Lineberry is a security software engineer and Linux security researcher. The concept of using `/dev/kmem` to rootkit Linux systems was first written about by Silvio Cesare in 1998 and by *devik* in *Phrack* magazine in 2001. Besides bringing this seldom-discussed attack vector back to people's attention, Anthony Lineberry has uncovered some new ways of tricking the kernel to allocate memory for injected code. Anthony and I chatted via e-mail immediately before and after his Black Hat Europe presentation.

MB: Hi, Anthony. Thanks for taking the time to talk to *Linux Journal*! It looks like this attack has ramifications very similar to those of the Loadable Kernel Module rootkit. Obviously, this isn't the best forum for a detailed dissertation, but could you describe your `/dev/mem` attack for our readers?

AL: We are essentially using the mem device to inject code directly into the kernel. `/dev/mem` is just a character device that provides an interface to physically addressable memory. Seeking to an offset and performing a read will read from that physical location in memory. Translating virtual addresses in the kernel to the physical addresses they map to, you can use simple reads and writes to this device to hot-patch code directly into the kernel. Using various heuristics, you can locate various important structures in the kernel and manipulate them. At that point, you are able to control behavior and manipulate almost anything inside the kernel, including system call tables, process lists, network I/O and so on.

MB: Does the attacker have to be root to locate and manipulate these structures in memory?

AL: Yes, you would definitely have to be root to be able to read/write to this device and manipulate any structures inside the kernel.

SMALL, EFFICIENT COMPUTERS WITH PRE-INSTALLED UBUNTU.

GS-Lo8 Fanless Pico-ITX System

Ultra-Compact, Full-Featured Computer
Excellent for Industrial Applications



3677 Intel Core 2 Duo Mobile System

Range of Intel-Based Mainboards Available
Excellent for Mobile & Desktop Computing



DISCOVER THE ADVANTAGE OF MINI-ITX.

Selecting a complete, dedicated platform from us is simple: Pre-configured systems perfect for both business & desktop use, Linux development services, and a wealth of online resources.



LOGIC
SUPPLY

www.logicsupply.com

MB: How does this differ from LKM rootkits?

AL: LKMs, in general, will create a lot of “noise” when loaded into the kernel. Using these techniques, we avoid all of that because of the fact that we are injecting directly into physical memory. Using an LKM does make it much easier to develop a rootkit. All of the effort can go into the actual code, rather than having to determine reliably where everything is inside the kernel. Although we can read/parse the export table inside kernel memory to locate almost all exported symbols.

The general suggested way to mitigate kernel rootkits for Linux is to configure a non-modular kernel and have all devices being used compiled in. In this scenario, we are still able to get code into kernel space.

MB: Have you tested the attack in virtualized environments? Does virtualized memory behave any differently?

AL: Yes, these methods will work in a virtualized environment. The main difference I ran into was that some special instructions handled by hypervisors behaved differ-

ently. Specifically in this case, the lidt instruction I used for locating the IDT/System Call Table in memory would return a bogus virtual address, but these problems were mostly trivial to overcome.

MB: What are the best defenses against /dev/mem attacks?

AL: The best defense is to enable CONFIG_STRICT_DEVMEM (originally called CONFIG_NONPROMISC_DEVMEM in 2.6.26) in the kernel, which limits all operations on the mem device to the first 256 pages (1MB) of physical memory. This limitation will allow things like X and DOSEMU, which use this device legitimately to still function properly, but keep anyone else from reading outside of those low areas of memory. Unfortunately, the default configuration leaves this protection disabled.

MB: Have you contacted any of the major Linux distributors (Red Hat, Novell and so forth)? Have any of them committed to enabling this setting in their default kernels?

AL: No, [although] many major distros do enable this setting by default in their releases. I would like to plan on compiling a list of who does/doesn't enable this.

Do you take
"the computer doesn't do that"
 as a personal challenge?
 So do we.

**LINUX
 JOURNAL**
Since 1994: The Original Monthly Magazine of the Linux Community

Subscribe today at www.linuxjournal.com

Resources

"Malicious Code Injection via /dev/mem" by Anthony Lineberry: dtors.org/papers/malicious-code-injection-via-dev-mem.pdf

"Alice in Kernel Land: Malicious Code Injection via /dev/mem" (slides to Anthony Lineberry's Black Hat Europe 2009 presentation): dtors.org/papers/injection-via-dev-mem.pdf

"Runtime Kernel kmem Patching" by Silvio Cesare: doc.bughunter.net/rootkit-backdoor/kmem-patching.html

"Linux on-the-fly kernel patching without LKM" by sd and devik, *Phrack* 58 (December 28, 2001): www.trust-us.ch/phrack/show.php?p=58&a=7

"Linux Kernel Rootkits" by Rainer Wichmann: www.la-samhna.de/library/rootkits/index.html

"Who needs /dev/kmem?" by Jonathan Corbet: lwn.net/Articles/147901

"The details on loading rootkits via /dev/mem" by Jonathan Corbet: lwn.net/Articles/328695

Some Notes on Mitigation

As Anthony said, short of ripping /dev/mem and /dev/kmem out of your kernel (which almost certainly would break things, especially in the X Window System), the best defense is to compile CONFIG_STRICT_DEVMEM=y in your kernel. The default kernels for Fedora and Ubuntu systems already have this option compiled in. RHEL goes a step further, by using an SELinux policy that also restricts access to /dev/mem.

If you don't know whether your system's kernel was compiled with CONFIG_STRICT_DEVMEM=y, there are several different ways to find out. Depending on your Linux distribution, your system's running kernel's configuration file may be stored in /boot, with a name like config-2.6.28-11-generic. If so, you can grep that file for DEVMEM. If not, your kernel may have a copy of its configuration in the form of a file called /proc/config.gz, in which case you can use the command:

```
zcat /proc/config.gz | grep DEVMEM
```

Otherwise, you need to obtain source code for your running kernel, do a make oldconfig (which actually extracts your running kernel's configuration), and check the resulting

.config file. In any of these cases, if CONFIG_STRICT_DEVMEM is set to n rather than y, you need to compile a custom kernel.

To do so, after having done make oldconfig, which even if you already knew your kernel lacked CONFIG_STRICT_DEVMEM enablement is a good idea, because you're probably interested in only changing CONFIG_STRICT_DEVMEM and leaving the rest of the kernel the same, you can do either make menuconfig or make xconfig. In the resulting menu, select kernel hacking, look for the option Filter access to /dev/mem, set it to y, exit, save your configuration, and re-compile.

If this entire kernel-compiling process is new to you, refer to your Linux distribution's official documentation for more detailed instructions. The process of compiling a custom kernel is, nowadays, rather distribution-specific, especially if you want to generate a custom RPM or deb package (which is the least disruptive way to actually *install* a custom kernel on RPM- and deb-package-based systems).■

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



6th Annual

HIGH PERFORMANCE COMPUTING ON WALL STREET

September 14, Monday Roosevelt Hotel, New York, NY

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

Plan to Attend This Focused Wall Street IT Conference

Meet the challenges of a downturned economy. Our 6th annual focuses on reducing computing costs on Wall Street.

Wall Street and the financial markets are looking for **cost saving alternatives** in this economic downturn. The Show will feature High Performance, Linux, Open Source, Virtualization, Cloud Computing, Grid, Blade, among other cost-saving technologies

Join IBM and other featured sponsors and exhibitors at the largest High Performance Computing Show in New York City this year.

High Performance Computing is driving new strategic trading, analytics and risk management systems.

2008 Platinum Sponsors

Microsoft

Enigmatec
CORPORATION

2008 Gold Sponsors

IBM

CISCO

intel

redhat

hp

Novell

Sun
microsystems

BLADE
NETWORK TECHNOLOGIES

SUPERMICRO

Solace Systems

ClearSpeed

2008 Silver Sponsors

Celoxica

NetApp

APPR

QLOGIC

TERVELA

RAPID MIND

2008 Media Sponsors

LINUX
JOURNAL

ON-DEMAND
ENTERPRISE

WINDOWS
TECHNOLOGY CENTER

LINUX
HPC

HPC

Global Investment
Technology

SDTimes

VSTX
Virtual Storage Technology

Show Management:

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

A-TEAMGROUP

Conference Management:
Pete Harris
917-379-7287
pete@a-teamgroup.com

www.highperformanceonwallstreet.com



KYLE RANKIN



BILL CHILDERS

Twitter

This month, Kyle and Bill go toe to toe on one of the hottest Internet waves to hit in recent memory.

That's right, we're covering microblogging—more specifically, Twitter. Kyle thinks Twitter is just another rehash of tried-and-true tech, while Bill thinks it fills an interesting niche in people's on-line lives. What's the reality though? Read on to find out. (Bill is simu-tweeting this, so his replies are limited to 140 characters, just to prove that Twitter *can* be useful.)

BILL: Kyle, I found a Twitter client for you. It's text-based and plugs in to an IRC client like another channel. It might help with your Web 2.0 hate.

KYLE: You know, if I could add it as yet another IRC channel or tie Bitlbee into it, then I probably would. That way, it wouldn't be that annoying, just another thing to lurk in.

BILL: Besides, Shawn Powers is on Twitter. You can tweet with more people than just me. It'll be fun!

KYLE: Bill, forget it. Someone already took greenfly. It's not fun if they took my name.

BILL: Wow. I can see you pout from here.

KYLE: Oh well, I'll just wait for the next bandwagon. One's bound to clatter by any time now. Or, maybe I'll just use IRC or IM to do the *exact same thing*.

BILL: Kyle, I thought the same thing. I was so wrong. Twitter isn't anything like chat.

KYLE: Yeah, well, Twitter matches your iPhone better. *Kyle follows Bill's mom.*

BILL: Hey, leave my mom out of this. Besides, she's not on Twitter. The closest to that is my sister, who is on Facebook.

KYLE: Okay, Bill. It's obvious you like Twitter. But seriously, why is someone who's "following" you different from someone having you on a buddy list or being in an IRC room?

BILL: Because Twitter is totally public.

KYLE: So is IRC; it's just not written in AJAX.

BILL: Not that way. At any given time there is a

limited number of people in an IRC room. With Twitter, my content is open to the whole world...

BILL: ...whether they follow or not. Not only is it visible to everyone, but others can respond to it without following me.

KYLE: Nice cheat there. Having to hack around the limitations of your service.

BILL: Some see it as a feature that enables creativity by constraining the amount of content. It's quality, not quantity.

KYLE: Considering all the posts that talk about what someone had for lunch, I don't know that everyone on Twitter shoots for quality either. Really, people use Twitter like a chatroom. You can replace "what are you doing right now?" with what you put in an IM or IRC away message.

BILL: Sometimes, but it's more like a blog or threaded forum. Have you heard the term microblogging? It fits Twitter perfectly.

KYLE: So basically, it's IRC applied to cell phones. I will give you this though, with Twitter, at least there are fewer normal blogs full of one-sentence-long posts linking to other content.

BILL: How can you argue that when you've not even *tried* it?

KYLE: They took my screen name, so there's little point in using it. "Krankin" isn't nearly as fun as "greenfly".

BILL: Let me get this straight. Because you can't get your handle on Twitter, it's a useless, duplicate service?

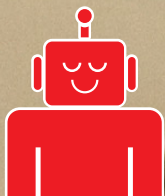
KYLE: No, it was a useless and redundant service before I tried it, but now that I can't have the handle, I also have no motivation to do it as a favor to you.

BILL: Okay, I get that you're annoyed. But admit it, you're still judging without fully understanding.

KYLE: What's there to understand? People read text; people occasionally reply to text. You get into a

Lullabot- Powered

The most super powered sites in the world are created in Drupal, by you and Lullabot.



Lullabot™ New Lullabot Learning Series training DVDs at Lullabot.com

Suzi Arnold
Director of New Media
Sony Music

LINUXTM JOURNAL

Linux News and Headlines Delivered To You



Linux Journal
topical RSS feeds
NOW AVAILABLE

http://www.linuxjournal.com/rss_feeds

POINT/COUNTERPOINT

group with certain people and all of them see what you write and (potentially) vice versa.

BILL: The difference is the audience is broader. Broader than arguably a blog. You could use it as a good marketing tool, if you quit judging.

BILL: But noooo, just go ahead and sit in your tower, and pout.

KYLE: It's just that I don't see it filling a need that wasn't already filled with IM and IRC. I honestly think half the reason it's so popular is everyone from all the news outlets, celebrities and now Oprah have all jumped on the bandwagon, so they can seem hip and on the cutting edge of this modern age. When they talk about tweeting, it just reminds me of everyone stumbling over MySpace a few years ago. Once everyone's mom got an account, they made a mass exodus. I think the same thing will happen soon with Twitter, once it's no longer a hip, niche technology.

BILL: Perhaps that will happen. But Twitter is not anything like MySpace. I think of Twitter more like the old UNIX wall command...

BILL: ...you shout out what you want to say and not only do your followers see it, but it also winds up on the Twitter main page...

BILL: ...and if it's actually usable, your followers can tell others. Word spreads very quickly via that path (retweeting).

BILL: Texting is a valid form of communication. Twitter takes texting and extends it to the Web and the world.

KYLE: One final thing is that even though Twitter's API is opened up so you can write your own client, you still are beholden to its servers to use the service. At least with IRC, if I don't like any of the available servers, I can start up my own IRC daemon on my own hardware. At least, with IRC and IM, it's not painful to have a low-latency real-time conversation with others. The bottom line for me is I just don't see anything special you can do with Twitter that couldn't be done better with other technologies that let you write more than one sentence at a time.

BILL: That will change with time. Laconica/Identicia offer a microblogging service that's open, free and federated. Much like AIM...

BILL: ...that IM service was and is closed. Jabber has since come along and surpassed it. I suspect Laconica will do the same.

BILL: Whatever your opinion, Kyle, it's likely that microblogging will be around. It may be a niche, but it's a useful niche. ■

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

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

TEXAS' OLDEST CRAFT BREWERY

WWW.SAINTARNOLD.COM

EtherDrive[®]

The **AFFORDABLE** Network Storage

Fiber Channel speeds at Ethernet prices

CORAID | **vmware[®]** | technology alliance **PARTNER**
ESX 3.5 compatible EtherDrive[®] HBA

ARM9 System on Module
Internet Appliance Engine **SoM-9G20**

- Atmel ARM9 400Mhz CPU
- 10/100 BaseT Ethernet
- SD/MMC Flash Card Interface
- 2 USB 2.0 Host Ports & 1 Device Port
- 6 Serial Ports, 2 SPIs & Audio Interface

The SoM-9G20 is the ideal processor engine for your next design. The System on Module (SoM) approach provides the flexibility of a fully customized product at a greatly reduced cost. Single unit pricing starts at \$155.

EMAC Linux 2.6 Kernel

EMAC, inc.
EQUIPMENT MONITOR AND CONTROL

Phone: (618) 529-4525 • Fax: (618) 457-0110 • Web: www.emacinc.com

FOSHealth 09
unconference

<http://fosshealth.eventbrite.com>

Friday July 31 to
Sunday, August 2
in Houston, T.X

Use registration code
'ljpgm' for \$100 off.

libertyhsf.org

INNOVATION ON THE GO
ORDER YOUR BEAGLE BOARD FROM DIGIKEY.COM

AVAILABLE EXCLUSIVELY AT DIGI-KEY

beagleboard

only \$149⁰⁰

**LOW-COST, NO FAN,
SINGLE-BOARD
COMPUTER**

Digi-Key[®]
CORPORATION

www.digikey.com

axigen
MAIL SERVER

The Mail Server
for IT Professionals

www.axigen.com

American made Utility Kilts for Everyday Wear

UTILIKILTS.com



The Mania of Owning Things

I think I could turn and live with animals, they are so placid and self-contained, I stand and look at them sometimes half the day long.

*They do not sweat and whine about their condition,
They do not lie awake in the dark and weep for their sins,
They do not make me sick discussing their duty to God,
No one is dissatisfied, not one is demented with the mania of owning things,
Not one kneels to another, nor to his kind that lived thousands of years ago,
Not one is respectable or industrious over the whole earth.*

—Walt Whitman

DOC SEARLS

For the second time in six months, our house has been threatened by wildfire—the city’s third in less than a year. That threat continues right now, and it’s so absorbing that I have no choice but to write about it—or about a subject close enough to serve our purposes here.

Our house is in Santa Barbara, a town that exists by temporary exception to the vicissitudes of nature. Soil studies show that it has burned about four times per century, going back to the Pleistocene. Earthquakes violent enough to level rock walls occur on an average of every 75 years. Tsunamis average once every 500 years. The last was in 1812, along with an earthquake that leveled the city’s famous mission, for the first of several times. All of this means that people putting a city in this place is a bit like ants building a nest in a footpath.

Yet we do. Ambition and industry in the face of inevitable destruction is the job of life. So I’m not in full agreement with Whitman on that one. Nor do I agree with other assertions in that passage. But, the line I chose for the title always has hit home for me.

I believe in ownership—not for economic reasons, but because possession is 9/10ths of the three-year-old. We are all still toddlers in more ways than we’d like to admit—especially when it comes to possessions.

We are grabby animals. We like to own stuff—or at least control it. Where would a three-year-old be without the

first-person possessive pronoun? No response is more human than “Mine!” And yet possessions are also burdens. I have a friend whose childhood home was burned twice by the same nutcase. He’s one of the sanest people I know. I can’t say it’s because he has been relieved of archives and other non-negotiables, but it makes a kind of sense to me. I have tons of that stuff, and I’ve thought lately about what it would mean if suddenly they were all cremated. Would that really be all bad? What I’d miss most are old photos that haven’t been scanned and writing that hasn’t been digitized in some way. But is my digital stuff all that safe either?

I just figured that I have about 4TB of digital stuff. Eliminate duplicate files and I’d guess the sum goes down to about 600GB. Most of those are photos. Now that big drives are cheap, I have backups of backups. Some are here in Boston (where we have what our kid calls “alt.home” or “shift_home”). Some are back in Santa Barbara. None are safe from fire or theft.

I’ve just started backing them up “in the cloud”. But how safe is that? Or secure? Companies are temporary. Servers are temporary. Hell, everything is temporary.

When I was young, I acknowledged death as part of the cycle of life. Now I think it’s the other way around. Life is part of the cycle of death. Life generates fuel for death. It’s a carbon-based refinery for lots of interesting and helpful stuff.

Think about it. Marble. Limestone. Travertine. Oil. Gas. Coal. Wood. Linoleum. Cement. Paint. Plastics. Paper. Asphalt. Textiles. Medicines. Even the heat used to smelt iron and shape glass comes mostly from burning fossil fuel. The moon has abundant aluminum ores. But how would you produce the heat required for extraction, or do anything without the combustive assistance of oxygen? Ninety-eight percent of the oxygen in Earth’s atmosphere is produced by plants. Most of the sources are now dead, their energies devoted to post-living purposes.

The Internet grows by an odd noospheric process: duplication. In “Better Than Free”, Kevin Kelly makes an observation so profound and obvious that you can’t shake it once it sinks in: “The Internet is a copy machine.” As a result, the Net is turning into what Bob Frankston calls a “sea of bits”. This too is an ecosystem of sorts. Is it, like Earth’s ecosystem, a way that death makes use of life? I wonder about that too.

This fire brings home an observation made by my son Allen way back in 2003: that a live Web feeds the static one. We see this now with Twitter. I use a Greasemonkey script to make Twitter search results appear atop my Google searches. Thanks to these, tweets appear in Google searches an instant after they are posted. Thanks to tweeting, I rely on a long tail of interested parties to keep me up on #jesusitafire. Many of these tweets point to other live Web postings of immediate interest. In time, most of these scroll to the static Web of archival material that comprises most of what you find on Google. The live Web is a system of rivers feeding the sea of bits. That too is a source of life. It is dead stuff that feeds the noosphere’s cerebrum, which in turn produces more of it.

In life, each of us takes what we brought. What we leave is what matters to the rest of us. ■

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

You know one hour of downtime is a big deal. Make sure your hosting provider does, too.



Your E-commerce concerns center around availability, security, compliance and scalability. Our responsibility as Linux hosting experts is to help you eliminate those worries when it comes to the hosting infrastructure behind your site. Bottom line — if your E-commerce site needs to be online and stay online, it needs to be hosted at Rackspace.

With our 200 RHCEs, we have the expertise and support to keep you online all of the time. And we understand the stakes — your reputation and revenue can suffer drastically from even a few minutes of downtime. You need to be up and running under any circumstances. **We work around the clock to make sure you are.**

Always Open



Microsoft

CISCO



AMD



rackspace.com/linuxjournal • 888-571-8976 | *experience fanatical support*

rackspace
HOSTING

More GFLOPS, Less WATTS

Intel® Nehalem is here!

**Higher Memory Bandwidth with DDR3 and QPI
Clusters and Servers Consume Less Power**

Four Servers in a 2U Chassis with all Hot-Swap:

- ▶ 1200 Watt 1+1 supply, 12 Drives, and Server Modules!

**FasTree™ ConnectX® QDR and DDR InfiniBand
Switches and HCAs**

Intel Professional Compiler Suite and Cluster Toolkit

- ▶ Version 11 with Nehalem Enhancements
- ▶ Academic Pricing Available



Configure your next Cluster today!
www.microway.com/quickquote



GPU Computing



WhisperStation™
With 1 to 4 Tesla GPUs

Tesla C1060 GPU Performance:

- ▶ 1 TFLOPS per GPU
- ▶ 4 GB DDR3 per GPU
- ▶ 102 GB/Sec Bandwidth
- ▶ CUDA SDK

Run MATLAB® on Tesla with "Jacket"

Clusters With Tesla™
S1070 - 4 GPU Servers

- ▶ 36 GPUs + 36 CPUs + 24 TB in 24U
- ▶ 40 Gbps FasTree™ InfiniBand
- ▶ InfiniScope™ Network Monitoring

FREE 15-day trial available
at microway.com

Microway
Technology you can count on™

508-746-7341
microway.com



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