Cassandra | DirB | sc | rdiff-backup | Mutt | MeeGo | Virtual Box

LINUX JOURNAL

Since 1994: The Original Magazine of the Linux Community

OCTOBER 2010 | ISSUE 198 | www.linuxjournal.com

Intro to MeeGo

Can We Fix 3G?

REVIEWED: Ben NanoNote

Learning Linux

ls -1 ~/features

Command-Line_Application_Roundup

DirB_Directory_Bookmarks_for_Bash

rdiff-backup_to_Back_Up_and_Restore

sc_the_Spreadsheet_Calculator

ls -1 ~/etc

Mutt_Configuration_Primer

Virtualization_with_VirtualBox

Get_Started_with_Cassandra

Calculating_Daylight_with_Bash

Controlling_Processes

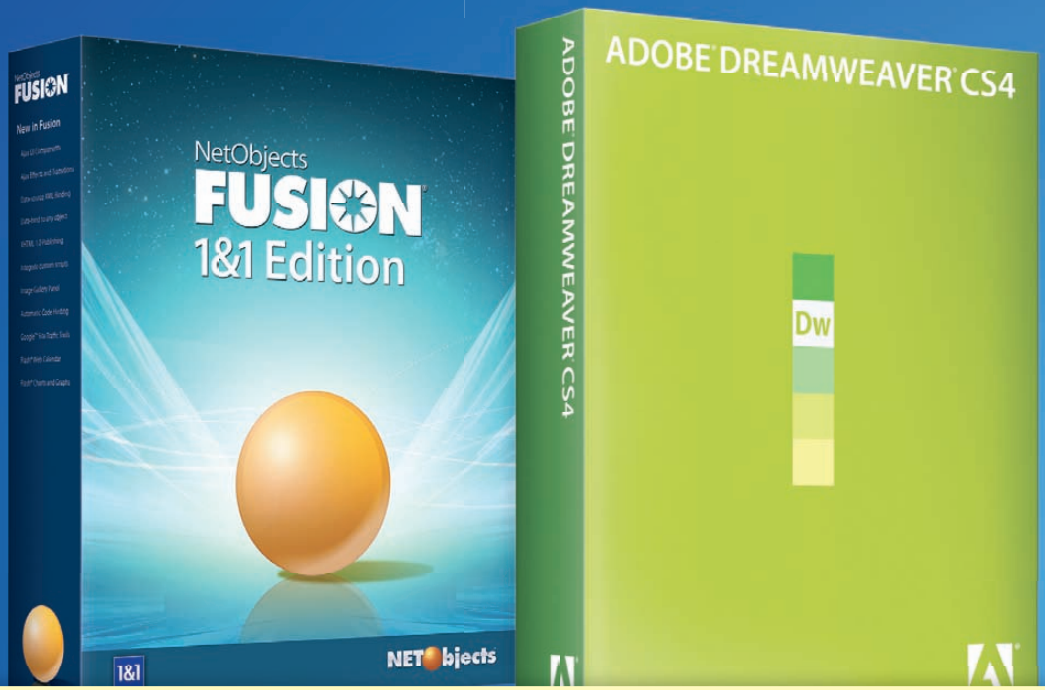
What's_New_in_Kernel_Development

Google_TV

PLUS: Point/Counterpoint: Sane Defaults vs Configurability

JOIN THE MOBILE REVOLUTION WITH THE WOR

WEBSITES



FREE SOFTWARE*

INCLUDED WITH 1&1 HOSTING PLANS. CHOOSE FROM:

NetObjects Fusion® 1&1 Edition is a website design application which creates sites that are optimized for mobile viewing. The 1&1 Edition was designed specifically for 1&1 web hosting packages and includes additional mobile templates as an extra bonus.

Adobe® Dreamweaver® CS4 is a sophisticated website design application for creating professional websites. Dreamweaver® includes the Adobe® Drive Central module, enabling web designers to test their websites on mobile devices by emulating the latest smartphones.

TURN PAGE FOR DETAILS OR VISIT WWW.1AND1.COM.

EPA GREEN POWER PARTNER

LD'S LARGEST WEB HOST. AT 1&1 INTERNET:

GO MOBILE

More and more people are browsing the web via their iPhone, BlackBerry®, or smartphone. Don't miss out on these customers! At 1&1, you get the software you need to create additional websites that are optimized for mobile viewing.

- ✓ **Layouts, designs and wizards enhanced for the latest smartphones, like iPhone and BlackBerry®**
- ✓ **Compatible across multiple platforms**
- ✓ **Valued at up to $479!**

1&1®

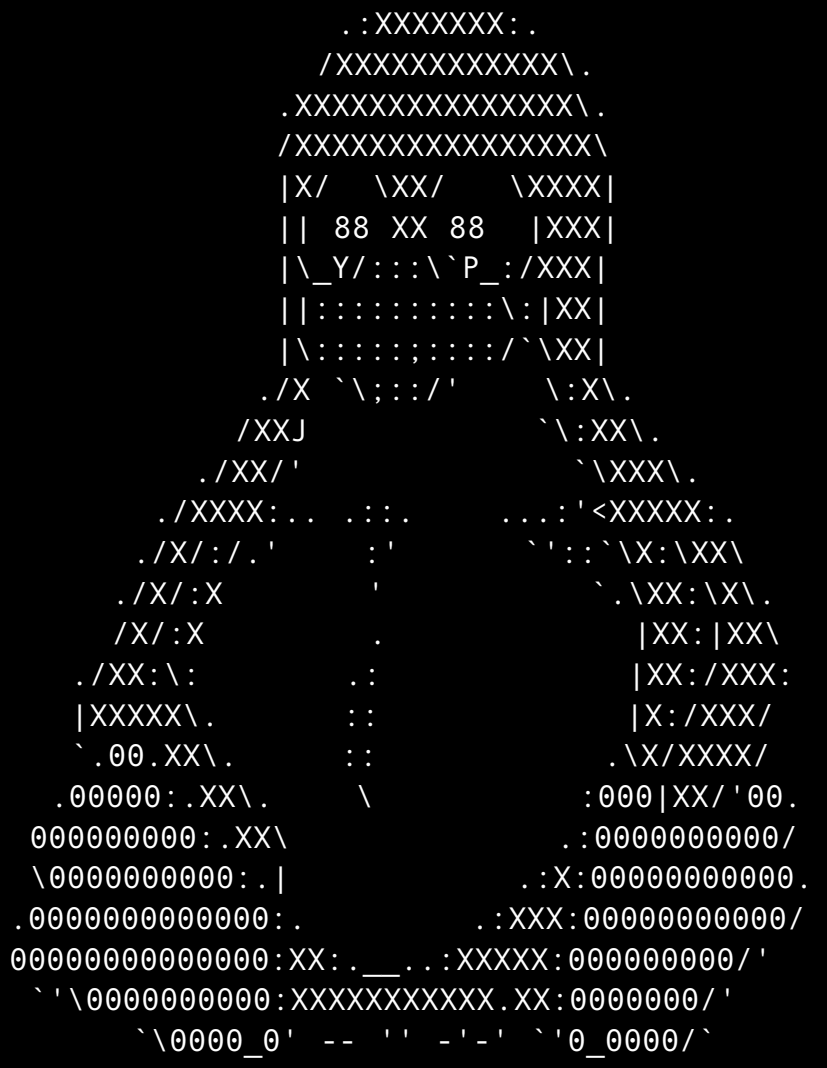
MEMBER OF united internet

Get started today, call 1-877-GO-1AND1 **www.1and1.com**

*Software offer valid with select 1&1 web hosting plans, and is available for download in the 1&1 Control Panel only. 12 month minimum contract term, setup fee, and other terms and conditions may apply. Visit www.1and1.com for full promotional offer details. 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. © 2010 1&1 Internet, Inc. All rights reserved.

CONTENTS

OCTOBER 2010
Issue 198



FEATURES

COMMAND LINE

42
Command-Line Application Roundup
A quick overview of some popular command-line tools.
Jes Fraser

48
DirB, Directory Bookmarks for Bash
Making the command line go faster.
Ira Chayut

54
sc: the Venerable Spreadsheet Calculator
A spreadsheet you can run in a terminal.
Serge Hallyn

60
Using rdiff-backup and rdiffWeb to Back Up and Restore
Are you backed up?
Adrian Klaver

ON THE COVER

- Command-Line Application Roundup, p. 42
- DirB, Directory Bookmarks for Bash, p. 48
- rdiff-backup to Back Up and Restore, p. 60
- sc: the Spreadseet Calculator, p. 54
- Mutt Configuration Primer, p. 28
- Virtualization with VirtualBox, p. 72
- Get Started with Cassandra, p. 20
- Calculating Daylight with Bash, p. 24
- Controlling Processes, p. 16
- What's New in Kernel Development, p. 14
- Google TV, p. 18
- Intro to MeeGo, p. 66
- Can We Fix 3G?, p. 80
- Reviewed: Ben NanoNote, p. 38
- Point/Counterpoint: Sane Defaults vs. Configurability, p. 76

BIG SAVINGS

ON PROFESSIONAL WEB SOLUTIONS

With 1&1, you get premium web design software, and 50% off the first 6 months on our most popular web hosting plans.

NOW GET 50% OFF PLUS FREE SOFTWARE*

1&1® HOME PACKAGE

- 2 Domain Names Included (.com, .net, .org, .info or .biz.)
- 150 GB Web Space
- UNLIMITED Traffic
- 10 FTP Accounts
- 25 MySQL Databases
- Extensive Programming Language Support: Perl, Python, PHP4, PHP5, PHP6 (beta) with Zend® Framework
- NetObjects Fusion® 1&1 Edition

~~$6.99 per month~~

$3.49* per month for the first 6 months

1&1® BUSINESS PACKAGE

- 3 Domain Names Included (.com, .net, .org, .info or .biz.)
- 250 GB Web Space
- UNLIMITED Traffic
- 25 FTP Accounts
- 50 MySQL Databases
- Extensive Programming Language Support: Perl, Python, PHP4, PHP5, PHP6 (beta) with Zend® Framework
- NetObjects Fusion® 1&1 Edition or Adobe® Dreamweaver CS4

~~$9.99 per month~~

$4.99* per month for the first 6 months

1&1® DEVELOPER PACKAGE

- 5 Domain Names Included (.com, .net, .org, .info or .biz.)
- 300 GB Web Space
- UNLIMITED Traffic
- 50 FTP Accounts
- 100 MySQL Databases
- Extensive Programming Language Support: Perl, Python, PHP4, PHP5, PHP6 (beta) with Zend® Framework
- NetObjects Fusion® 1&1 Edition or Adobe® Dreamweaver CS4
- NEW: 1&1 Power Plus Performance Guarantee

~~$19.99 per month~~

$9.99* per month for the first 6 months

ALSO ON SALE:

.us domains $0.99/first year*

.com domains $7.99/first year*

Visit our website for a full list of special offers.

1&1®

MEMBER OF united internet

Get started today, call 1-877-GO-1AND1

www.1and1.com

*12 month minimum contract term required for software offer. Setup fee and other terms and conditions may apply. Software available for download in the 1&1 Control Panel only. Domain offer valid first year only. After first year, standard pricing applies. Visit www.1and1.com for full promotional offer details. 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. © 2010 1&1 Internet, Inc. All rights reserved.

CONTENTS

OCTOBER 2010
Issue 198

COLUMNS

20 Reuven M. Lerner's
At the Forge
Cassandra

24 Dave Taylor's
Work the Shell
Function Return Codes and Daylight Calculations

28 Kyle Rankin's
Hack and /
Take Mutt for a Walk

76 Kyle Rankin and Bill Childers'
Point/Counterpoint
Sane Defaults vs. Configurability

80 Doc Searls'
EOF
3G Hell

REVIEW

38 A Look at the Ben NanoNote
Daniel Bartholomew

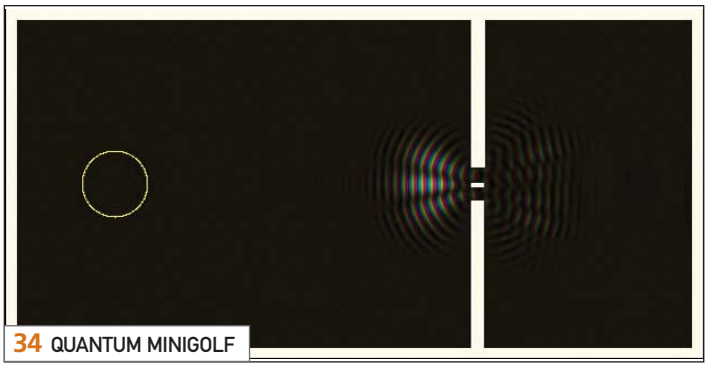
INDEPTH

66 Introduction to the MeeGo
Software Platform
Maemo + Moblin == MeeGo
Ibrahim Haddad

72 Virtualization the Linux/OSS Way
Manage VirtualBox from the command line.
Greg Bledsoe

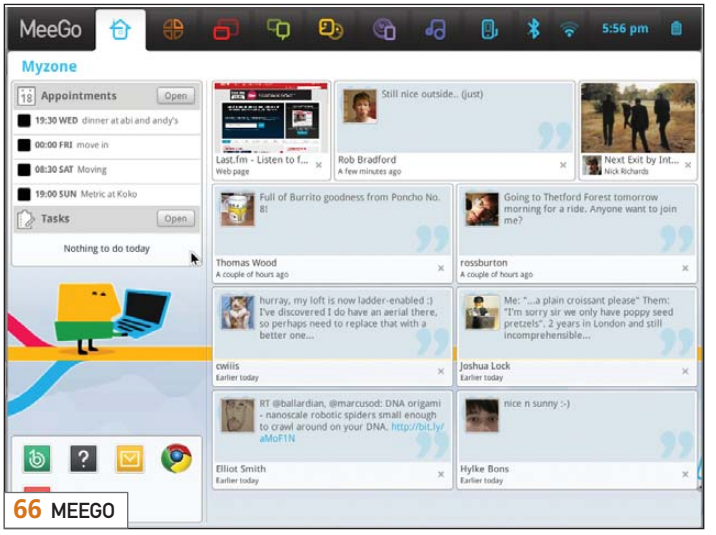
IN EVERY ISSUE

8 Current_Issue.tar.gz
10 Letters
14 UPFRONT
32 New Products
34 New Projects
65 Advertisers Index
78 Marketplace



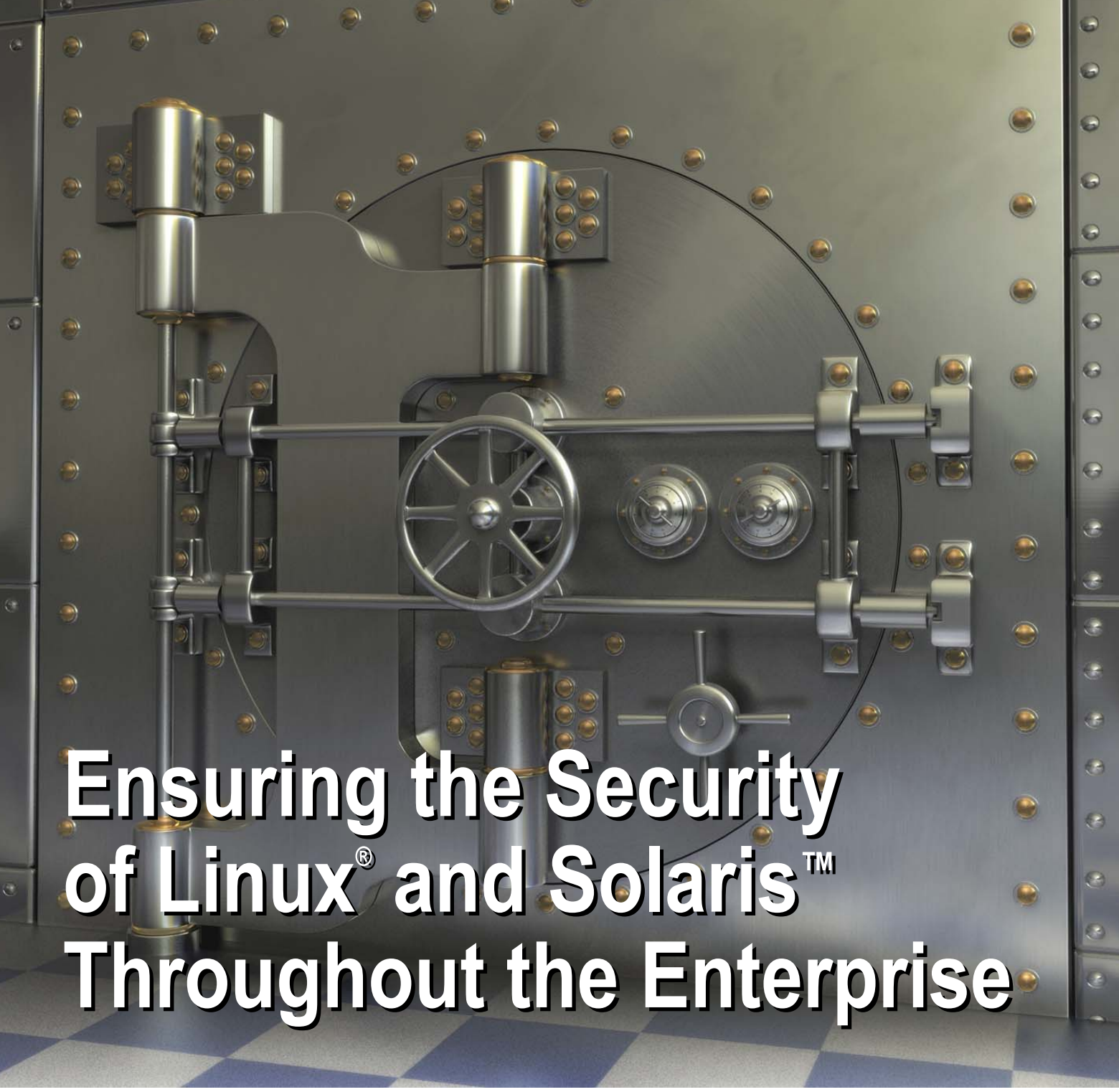
34 QUANTUM MINIGOLF

38 BEN NANONOTE



66 MEEGO

USPS *LINUX JOURNAL* (ISSN 1075-3583) (USPS 12854) is published monthly by Belltown Media, Inc., 2121 Sage Road, Ste. 310, Houston, TX 77056 USA. Periodicals postage paid at Houston, Texas and at additional mailing offices. Cover price is $5.99 US. Subscription rate is $29.50/year in the United States, $39.50 in Canada and Mexico, $69.50 elsewhere. POSTMASTER: Please send address changes to *Linux Journal*, PO Box 16476, North Hollywood, CA 91615. Subscriptions start with the next issue. Canada Post: Publications Mail Agreement #41549519. Canada Returns to be sent to Bleuchip International, P.O. Box 25542, London, ON N6C 6B2



Ensuring the Security of Linux® and Solaris™ Throughout the Enterprise

Automatically Configure and Manage Operating System Security Configurations with Security Blanket®

Everyone knows that improper, inconsistent security configurations can leave you vulnerable. Security Blanket, from Trusted Computer Solutions, centrally manages any number of Linux and Solaris systems, across multiple platforms, and automatically configures the security settings to industry standard or customized policies. Enterprise and government organizations save time, reduce risk, and maintain a consistent security posture with Security Blanket. You can too.

Try it for free at www.trustedcs.com/SecurityBlanket or call us at 1.866.230.1317 for more information.

SECURITY BLANKET BY TCS™

LINUX JOURNAL™
Since 1994: The Original Magazine of the Linux Community

DIGITAL EDITION NOW AVAILABLE!

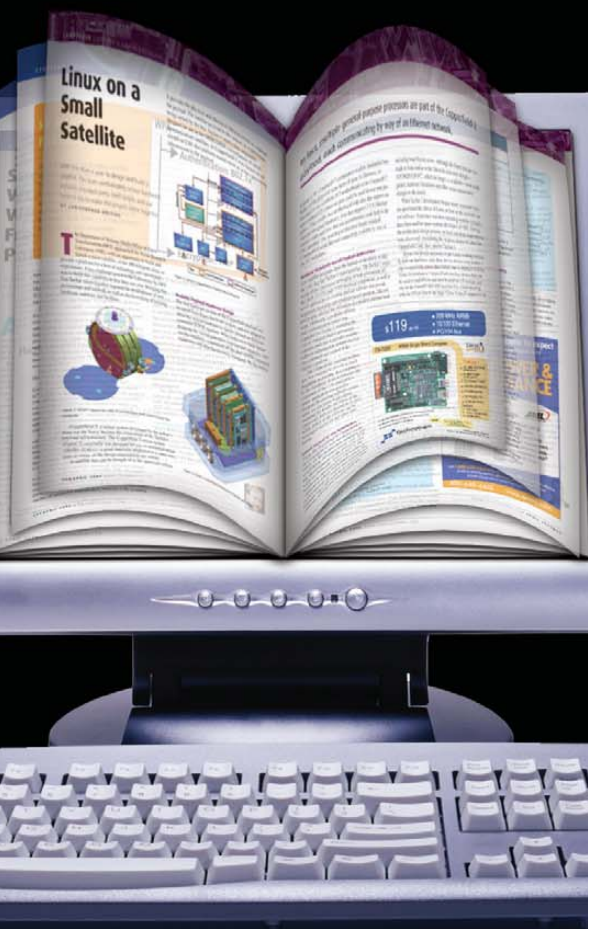
Read it first
Get the latest issue before it hits the newsstand

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/DLISSUE



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
Security Editor	Mick Bauer mick@visi.com
Hack Editor	Kyle Rankin lj@greenfly.net
Virtual Editor	Bill Childers bill.childers@linuxjournal.com

Contributing Editors
Ibrahim Haddad • Robert Love • Zack Brown • Dave Phillips • Marco Fioretti • Ludovic Marcotte
Paul Barry • Paul McKenney • Dave Taylor • Dirk Elmendorf • Justin Ryan

Proofreader Geri Gale

Publisher	Carlie Fairchild publisher@linuxjournal.com
General Manager	Rebecca Cassity rebecca@linuxjournal.com
Senior Print Media 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

Editorial Advisory Panel
Brad Abram Baillio • Nick Baronian • Hari Boukis • Steve Case
Kalyana Krishna Chadalavada • Brian Conner • Caleb S. Cullen • Keir Davis
Michael Eager • Nick Faltys • Dennis Franklin Frey • Alicia Gibb
Victor Gregorio • Philip Jacob • Jay Kruizenga • David A. Lane
Steve Marquez • Dave McAllister • Carson McDonald • Craig Oda
Jeffrey D. Parent • Charnell Pugsley • Thomas Quinlan • Mike Roberts
Kristin Shoemaker • Chris D. Stark • Patrick Swartz • James Walker

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.

PRINTED WITH SOY INK®

More TFLOPS, Fewer WATTS

Microway delivers the fastest and greenest floating point throughput in history

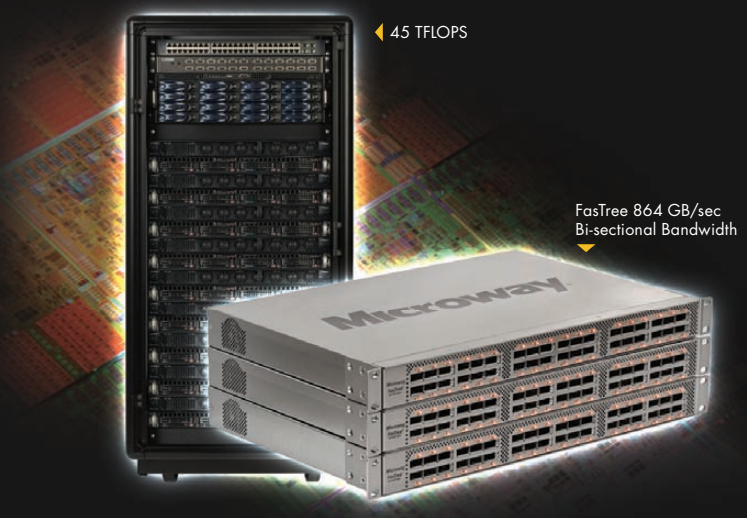
Enhanced GPU Computing with Tesla Fermi

- 480 Core NVIDIA® Tesla™ Fermi GPUs deliver 1.2 TFLOP single precision & 600 GFLOP double precision performance!
- New Tesla C2050 adds 3GB ECC protected memory
- New Tesla C2070 adds 6GB ECC protected memory
- Tesla Pre-Configured Clusters with S2070 4 GPU servers
- WhisperStation - PSC with up to 4 Fermi GPUs
- OctoPuter™ with up to 8 Fermi GPUs and 144GB memory

New Processors

- 12 Core AMD Opterons with quad channel DDR3 memory
- 8 Core Intel Xeons with quad channel DDR3 memory
- Superior bandwidth with faster, wider CPU memory busses
- Increased efficiency for memory-bound floating point algorithms

Configure your next Cluster today!
www.microway.com/quickquote
508-746-7341



FasTree™ QDR InfiniBand Switches and HCAs

- 36 Port, 40 Gb/s, Low Cost Fabrics
- Compact, Scalable, Modular Architecture
- Ideal for Building Expandable Clusters and Fabrics
- MPI Link-Checker™ and InfiniScope™ Network Diagnostics

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

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

NVIDIA Tesla CUDA | Intel Channel Partner Premier 2010 | AMD Fusion Partner Premier | GSA

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

Microway®
Technology you can count on℠

SHAWN POWERS

The Most Grepping Issue of the Year!

Bad puns aside, the Command-Line issue is one of our favorites. Although Linux has evolved into an elegant operating system complete with GUI front ends and a stylish visual appeal, at its core, Linux is still text configs, symbolic links and log files. Around these parts, we consider that a feature, so this month, we've dedicated our issue focus to the command line.

Reuven M. Lerner starts us off in the world of text as he continues his series on non-relational databases. This month it's Cassandra, which appears to have amazing scaling abilities. I suspect Dave Taylor's column this month is really a subtle joke about coders spending too much time in their parents' basements. He shows us how to use a script to determine whether the sun is up. Granted, we could peek outside, but why do all that needless work when our computers could do it for us!

Daniel Bartholomew reviews the Ben NanoNote—this fascinating little sub-$100 device is a real computer inside a case the size of a large cell phone. What might a computer that small be good for? Read Daniel's article to find out. When you're finished doing that, check out Jes Fraser's roundup of CLI-based applications. She covers everything from multimedia to editors to Web browsing, and even instant messaging. Perhaps an SSH session and the Ben NanoNote will be all you need for a computer! (Assuming you don't stray more than 15 feet from a bigger computer. Be sure to read Daniel's article.)

When you stick to the command line, a surprising number of solutions will keep all that GUI stuff out of your hair. Whether you want to bookmark directories in Bash (Ira Chayut shows how) or to kick it old-school with a text-based spreadsheet (Serge Hallyn covers that task), the command line can make a superhero out of anyone. If you don't believe me, take a look at our resident command-line superhero on page 18. Kyle Rankin knows root is the true master of the Linux universe, and he sports his powers proudly. In fact, if you've ever hung around with Kyle, you know that although he has a fancy high-powered laptop, his aversion to all things GUI makes it unnecessary. He uses it to `ssh` into an 800MHz server and does pretty much all his work from there. This month, he shows us part of his elaborate e-mail setup with mutt. If you've ever doubted the power of mutt, you won't after reading his column.

I'm sure many of you love the command line for those things best done on the command line, but prefer a more point-and-clicky interface for other stuff. We can respect that. In fact, although I do much of my sysadmin work on the command line, things like e-mail and spreadsheets just make more sense when they're GUI, at least for me. Ibrahim Haddad gives us an intro to MeeGo, which is a combination of Nokia's Maemo and Intel's Moblin. It's a GUI-based operating system for small screens. Of course, there's more to it than that, so you'll want to check it out yourself. We also have an article by Adrian Klaver that covers rdiff-backup, a command-line backup and restore system, but he also includes an intro to a Web-based front end to rdiff-backup called rdiffWeb. Finally, as one of those applications that can go either CLI or GUI, Greg Bledsoe shows us how we can use the normally graphical virtualization solution VirtualBox in a headless, command-line way.

Although choice is something we pride ourselves on in the Linux community, and those command-line-only folks can happily live with their GUI neighbors, Kyle Rankin and Bill Childers don't always agree on things. This month, feel free to take sides as they argue over sane defaults or extensive configurability in their Point/Counterpoint column. While they state their cases, in honor of the command-line issue, I think I'll go play a text adventure. I hope I don't get eaten by a grue.■

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

servers DIRECT®

MORE PRODUCTS, BETTER SERVICE, GUARANTEED.

GO STRAIGHT TO THE SOURCE!

1.877.727.7887 | www.ServersDirect.com

WANT YOUR BUSINESS TO BE MORE PRODUCTIVE?

The ServersDirect® Systems powered by Intel® Xeon® Processor provides the quality and dependability to keep up with your growing business.

HIGH EFFICIENCY POWER & HIGH STORAGE CAPACITY SERVER
SDR-S4314-T36

ENTERPRISE-LEVEL HIGH CAPACITY STORAGE SERVER

+ Maximum 3.5"" hot-swap drives density
  - 36x (24 front + 12 rear) Hard Drive Bays
  - E1: Expander supports SAS
+ Redundant (1+1) 1400W Gold Level power supply with PMBus function
+ Redundant, Hot-pluggable cooling system, Power Supplies, Hot-swap drives
+ Support up to 192GB DDR3 1333/ 1066/ 800MHz ECC Reg DIMM
+ Intel® 82576 Dual-Port Gigabit Ethernet Controller

80 PLUS

SDR-S1343-T04

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

- Supermicro 1U Rackmount Server with HE 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

80 PLUS

SDR-S2318-T06

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

- Supermicro 2U Rackmount Server w/650W HE 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 192GB DDR3 1333/ 1066 / 800MHz ECC Reg DIMM
- 6x 3.5" Hot-swap SAS / SATA Drive Bays
- Dual Intel® 82574L Gigabit Ethernet Controller

80 PLUS

SDR-S3308-T08

3U INTEL® XEON® PROCESSORS 5500 SERIES SERVER W/ 8 X 3.5" HOT-SWAP DRIVES TRAYS

- Supermicro 3U Rackmount Server w/650W HE 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 192GB DDR3 1333/ 1066 / 800MHz ECC Reg DIMM
- 8 x 3.5"" Hot-swap Drives Trays 6 x SATA Hard Drives Supported
- Dual Intel® 82574L Gigabit Ethernet

SDR-S4309-T08

TOWER/4U INTEL® XEON® PROCESSORS 5500 SERIES SERVER W/ 8 X 3.5" HOT-SWAP DRIVES TRAYS

- Supermicro Tower 800W 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 192GB DDR3 1333/ 1066 / 800MHz ECC Reg DIMM
- 24x 3.5" Hot-swap SATA Drive Bay
- Intel® 82576 Dual-port Gigabit Ethernet Controller

80 PLUS GOLD

SDR-S4314-T08

TOWER/4U INTEL® XEON® PROCESSORS 5500 SERIES SERVER W/ 8X 3.5" HOT-SWAP SATA BAYS (SUPPORT UP TO 4 DOUBLE-WIDTH GPU)

- 4U Rackmountable / Tower with 1400W Gold Level Redundant Power Supply Optional Rackmount Kit
- Supermicro Server Board w/ Dual Intel® Dual Chipset ICH10R
- Support up to Dual Intel® 5500 series Xeon® Quad/Dual-Core, with QPI up to 6.4 GT/s
- Support up to 192GB DDR3 1333/ 1066 / 800MHz ECC Reg DIMM
"• 8 x 3.5"" Hot-swap Drives Trays Supports up to 6 SATA Drives"
- Intel® Dual 82574L Gigabit Ethernet
- **4x NVIDIA Tesla C1060 GPU Cards**

SDB-S7002-T00

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

- Supermicro SBE-714D-D28 Enclosure chassis with two 1400W power supplies. Up to 14 hot-plug processor blades
- Support Intel based blades
- One Management modules comes standard with each enclosure
- One hot-plug management modules providing remote KVM and IPMI 2.0 functionalities
- Support one hot-plug Gigabit Ethernet switch, Pass - Through Module

10th year

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.

intel Xeon inside™

Powerful. Intelligent.

1.877.727.7886 / www.ServersDirect.com

Intel, Intel logo, Intel Inside, Intel Inisde 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.

letters

Temper Temper

Great journal—I always find many articles very interesting. Regarding Kyle Rankin's article in the August 2010 issue, as a cheaper alternative to a $15 USB stick, I suggest the DS18S20 –55°C to +125°C one-wire chip. These $2 T092 devices interface to a comm line using one resistor, two diodes and two zener diodes. Multiple devices (and other one-wire devices) can run in parallel providing a sensor at multiple spots. Each chip can be uniquely addressed, which then returns the temperature as a string with 0.5°C accuracy.

Naturally, there are LX drivers, and a search for DigiTemp will provide code plus diagrams. With regard to Kyle's mention of "$15 and spare parts to control a fridge", I use a $200 X10 controller (Ocelot) connected to my SUSE mail server to control several floodlights. These are triggered by X10 transmitters, cron jobs and by e-mail messages stating, for example, "x10 patio on". A serial interface to the Ocelot and just two X10 modules (DIN AD10s) cost around 100 GBP—hardly what I would call "spare parts".

--

Chris Seager

Kyle Rankin replies: *Thanks for the reply Chris. When I'm ready to break out my soldering skills, I'll have to give your solution a try. Unfortunately, I'm less advanced on the electrical engineering side, so the USB thermometer made things simpler for me. You wouldn't believe how many spare X10 outlets I have lying around, but back when I bought them, each outlet module was around $5–$10, so it sounds like they are way more expensive now.*

Shockwave Flash Videos

Some of us who use Linux have a terrible time getting Shockwave Flash videos to play. How about making all your videos available for download in another format, like OGV or MP4? For the record, I am one of your subscribers. Here is the video that I could not watch: "Shawn Powers shows us a very quick way to take screenshots using Compiz under Linux. Yes, there are plenty of screenshot tools available for Linux, but Compiz allows for a literal one-click method."

--

Volker

We always provided a link to OGV files, but apparently something went wrong with the Web site change-over. We'll have to fix that! Thank you for the note; it was just an oversight.—Ed.

OSWALD Article and FOSS in Education

I enjoyed Victor Kuechler and Carlos Jensen's "The OSWALD Project" in the August 2010 issue, and I look forward to more similar articles and projects. Computer Science education certainly faces challenges, including periods of boom and bust enrollments. It's also difficult for students and teachers to keep up with the rapid pace of change.

However, team-based integrative projects have been used for many years. FOSS expands the opportunities for such projects, and there is a growing community of educators who use FOSS. For example, Seneca College has hosted a Free Software and Open Source Symposium annually since 2002 (**fsoss.senecac.on.ca**). The Humanitarian FOSS Project (**hfoss.org**) is a collaborative project with NSF support to engage students to build free software systems that benefit communities. Some FOSS projects have student mentoring programs and flag tasks that are most suitable for student projects. For more information, see the Teaching Open Source Wiki (**teachingopensource.org**).

I use FOSS in my teaching, and I have been very happy with the outcomes, although different types of FOSS offer different benefits and risks for students. Some projects, such as OSWALD, have close ties to academic institutions, which can provide guidance and continuity, although it can be difficult for students or other developers outside the institution to get involved. Other projects, such as Drupal and Moodle, are much larger and more decentralized, so that students learn to interact with a wider variety of people. There are also countless small FOSS projects with opportunities for students.

To encourage and support FOSS in education, *LJ* readers could: 1) encourage prospective students and parents to seek out academic programs that actively use FOSS; 2) welcome students and teachers into FOSS projects and help them find suitable tasks within the project; and 3) contact local institutions and offer to talk about FOSS projects and mentor student teams.

--

Clif Kussmaul

I completely agree! Pushing FOSS into schools is one of my passions, and seeing colleges accept such practices is very encouraging. Thank you for fighting the good fight!—Ed.

Better Beer

Regarding Kyle Rankin's "Temper Temper" article in the August 2010 issue, I also built a temperature-controlled beer fermentation chamber but used an Arduino as the data-gathering computer. The Arduino measures the temperature using a simple thermistor and uses the Ethernet Shield to allow other computers to get the current state. My desktop Linux computer runs a cron script to get the current temperature and then (like Kyle's project) turns on/off the fridge using X10 modules. I also added

another input to allow the current on/off state of the power to the fridge to be determined. This uses a simple nightlight in the same circuit as the fridge (the light is on when the fridge is on) and a photo resistor to measure the light state. My control will turn on/off the fridge when the temperature reaches the appropriate value but will send the on/off command only one time if the temperature is still at the appropriate limit.

Since I have SSH access to my desktop Linux computer from the Internet, I am able to get the current temperature and power state from anywhere, including from my Android phone. This has been very helpful, as one time I checked the temperature remotely and found that the Arduino was not responding. A call to my wife indicated that we'd had a power outage and the Arduino needed to be reset, which my wife was able to do. My next step is to incorporate the real-time data plotting using kst as discussed in Rob Reilly's article "Real-Time Plots with kst and a Microcontroller" [also in the August 2010 issue]. Better Beer through the use of open-source hardware and software!

--

Ralph Noack

Hard Drive Costs?

In the August 2010 issue, the *LJ* Index lists item #11 as "hard drive cost per gigabyte in 1990: $53,000", but this is wildly high, even when considering the source listed.

$53k/gigabyte is $53/mbyte, and prices far less than that were listed *prior* to 1990 (as far back as October 1987). In addition, that's Canadian money, which in 1990 was about 85 cents to the US dollar.

The reason I remember this is that in 1989, I purchased a Control Data 383MB drive—one of the best you could get at the time—for $1,800, which (rounding up) is around $5,000/gigabyte.

But, it's certainly true that the drop in cost for both hard drives and RAM has been breathtaking. It's a great time to be a consumer.

--

Steve

Mitch Frazier replies: *Well shoot, just when we thought we'd cleaned up the last of the bad and misleading information on the Internet.*

Tripping Over Traps

I'm writing in response to E. Thiel's Letter to the Editor in the May 2010 issue, regarding Dave Taylor's use of trap 0. Although 0 is not a signal, it is trappable; it traps the exit of the shell and is sometimes used for cleanup code. Here's an example of its use:

```
#!/bin/sh
trap 'rc=$?; echo goodbye; exit $rc'
0
echo hello world
```

Particularly notice the way I saved the return code during execution of the trap and restored it for the final exit.

--

David Newall

Commons Interests

I should first say that I'm not a lawyer; I'm just some yahoo that took a few minutes to read and enjoy Doc Searls' excellent EOF column in the June 2010 issue and then read the text of the six basic Creative Commons licensing agreements on-line.

It seems that the Creative Commons licensing agreements don't appear to be exclusive agreements. Each of the six basic CC licensing options say, "Any of the above conditions can be waived if you get permission from the copyright holder." I couldn't find a clause that would restrict the author to the same "copyleft" restrictions. If that is the case, it would seem that the copyright holder has typically broad use rights so long as (s)he takes care not to invalidate the CC license. That seems to suggest that others can make derivative use of the IP published under the CC license but, for example, could not freely use NBC's works just because they included material that has been released under a CC license elsewhere.

I would simply echo the comment of your one commenter. Yes, it really is very nice of you to freely share your work, with or without encumbrances, and we are all the better for it.

--

Hal Lasell

That's Now How rsync Works

Your example [August 2010 issue Letters] does not describe how rsync works correctly:

```
cd /tmp/
mkdir a b
echo a/c >a/c
echo b/c >b/c
touch -r a/c b/c
cp -u a/* b/
cat b/c
rsync -a a/ b/
cat b/c
```

rsync doesn't update b/c, either! rsync compares metadata (size and timestamp) first; if they're the same, then it will not update the data (unless `-I` is specified). In this case, they're the same due to the similar contents and the `touch -r`.

--

John Wiersba

Thanks for pointing that out. I mistakenly thought that rsync did a more complex analysis when deciding what needed to be copied and what didn't.—Ed.

Libmobiledevice

Dirk Elmendorf commented in his August 2010 article that he had issues with the iPhone under Linux. Libmobiledevice (**www.libimobiledevice.org**) now has delivered near complete support for current-generation iPhones and iPods. Not only is this fantastic, as it liberates many of us Linux users from the last vestige of a dependence on another OS, it also merits an in-depth look from *Linux Journal*.

--

James Ervin

Don't Leak!

In his August 2010 article on CouchDB, Reuven M. Lerner appears to have committed a basic security error. He published the names and plausible birth dates of real people (his own children).

Admittedly, the details of some toddlers are unlikely to be useful to scammers for some time, but they are exposed to other dangers.

When creating test data, especially if they are to be published in an article, real names and personal details should never be used. Make up random names, dates, addresses and other personal attributes. If imagination fails you, use fictional characters (from out-of-copyright materials, unless you enjoy talking to lawyers).

Every test case should have some particular purpose. It's actually much better to use string fields to document the purpose of the particular record in the set than to fill it with fictional noise. (You may have to make an exception to this if you are demonstrating to a particularly stupid or literal-minded audience who cannot cope with anything but realistic-looking examples.)

Even if you never expect test or sample data to be revealed outside your office, department or closed group, keep it bogus. Recording the address and phone number of attractive colleagues might expose them to unwelcome attentions.

Phishers go to a lot of trouble to collect personal details. Don't make their lives easy.

--
Alan Rocker

Reuven M. Lerner replies: *I appreciate Alan's point about leaking personal information. I would use only information about people whom I know and from whom I've received permission. Fortunately, my children are ecstatic every time their names appear in* Linux Journal*, and they are quite happy to provide me with sample data. Maybe I'm naïve, but I'm simply not worried about their names and birth dates being available on the Internet. That said, we do take precautions with my children's access to the Internet, in order to protect them from potential harm. I just don't think that hiding their names or birth dates needs to be part of those precautions.*

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.

Here is a picture of the entertainment system booting Linux in an Airbus A320. Submitted by Ariel Martinez.

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**.

SUPERMICRO®

6Gb/s SAS 2.0 Server Solutions

A Full Spectrum of Value-Added Features

- 6 Gb/s SAS2.0 provides a 100% Increase in Data Trasfer Speed in Addition to Improving I/O Performance with increased density provided by 2.5" HDDs
- Complete Product Line including, Blades, Systems, Chassis, Motherboards, and SAS2.0 Controller
- Industry Leading Performance-per-Watt/per-Dollar on the market
- UIO RAID Cards Designed to Support Up to 240 drives for highly scalable storage capacity
- Extensive RAID Support with Optional Battery Back-Up Enhances Reliability and Performance
- Dual Intel® Xeon® 5600/5500 Series Processor support

MBX systems

Headquarters
1101 Brown Street
Wauconda, IL 60084
www.mbx.com

80 PLUS PLATINUM

80 PLUS GOLD

94%

intel Xeon inside™

Powerful. Intelligent.

©Super Micro Computer, Inc. Specifications subject to change without notice. Intel, the Intel logo, Xeon, and Xeon inside, are trademarks or registered trademarks of Intel Corporation in the US and other countries. All other brands and names are the property of their respective owners.

diff -u

WHAT'S NEW IN KERNEL DEVELOPMENT

Linux **hibernation** may be getting faster soon, or maybe just eventually. **Nigel Cunningham** came up with an entirely new approach to how to shut down each part of the system, such that it all could be stored on disk and brought back up again quickly. Unfortunately, **Pavel Machek** and **Rafael J. Wysocki**, the two co-maintainers of the current hibernation code, found his approach to be overly complex and so difficult to implement, that it really never could be accomplished. Nigel had more faith in his idea though. He felt that exactly those places that Pavel and Rafael had found to be overly complex actually were the relatively simpler portions to do. There was no agreement during the thread of discussion, so it's not clear whether Nigel will go ahead with his idea.

Some filesystems, notably **FAT**, have trouble slicing and dicing files into smaller pieces without having a lot of extra room available on the disk to copy the data. But logically, it shouldn't be necessary to copy any data, if the data isn't changing. **Nikanth Karthikesan** wanted to split up files even when the disk was virtually full, so he wrote a few system calls, **sys_split()** and **sys_join()**, to alert the system to the fact that no copying would be necessary. There was some debate over the quality of Nikanth's code, but **David Pottage** also pointed out that this type of feature could turn video editing from a many-hour task to a many-minute task, in certain key cases. He remarked, "Video files are very big, so a simple edit of removing a few minutes here and there in an hour-long HD recoding will involve copying many gigabytes from one file to another." In general, developers need a pretty strong reason to add new system calls, so it's not yet clear whether Nikanth's code will be included, even if he addresses the various technical issues that were raised in the discussion.

One thing that can happen on any running system is that RAM bits can flip as the result of high-energy particles passing through the chip. This happens in space, but also on the ground. **Brian Gordon** recently asked about ways of fixing those **Single Event Upsets** (SEUs). **Andi Kleen** and others suggested using **ECC** (Error Correction Codes) RAM, which could compensate for a single bit flip and could detect more than one bit flip. But Brian was interested in regular systems that were built on a budget and didn't have access to high-priced error-correcting RAM. Unfortunately, Andi said that this would be a very difficult feature to implement. Brian had talked about some kind of system that would use checksums and redundancy to maintain memory integrity, but Andi felt that even if that could be implemented in the kernel, it probably would require the user-space application to be aware of the situation as well. So that wouldn't be a very general-purpose solution after all. Brian may keep researching this, but it seemed like he really wanted to find a general solution that wouldn't require rewriting any user applications.

—ZACK BROWN

Make Your Android Follow Whatever Three Laws You Decide

A while back, I thought I'd write a long tutorial on how to root an Android phone and install a custom-compiled ROM on it. This is a useful and fun activity, because it can land you a phone running a more modern version of Android than it officially supports. Of course, it also voids any warranty on your device, so it's not without risk.

It turns out, writing an article for Droid-modding isn't really required. Assuming your phone has been hacked, a quick Google search will give you the directions to root your device (the simplest and least exciting part of hacking an Android phone). After that, installing Rom Manager from the Marketplace will allow you to flash a wide variety of custom ROMs onto your phone. I could walk you through the process, but it's really not terribly difficult. With all hacking and warranty-voiding activities, be aware that, although unlikely, it is possible you could ruin your phone and need to revert back to cans and string for communication. Don't say I didn't warn you.

Oh, and if you're looking for an inexpensive, yet widely supported device for hacking, the old Motorola Droid is inexpensive and most likely still available. It's not the newest phone in the Android world, but mine is happily running Froyo (Android 2.2) even though at the time of this writing, it hasn't been released for the Droid. Happy hacking!

—SHAWN POWERS

Emerge Desktop by priyodevil (from customize.org/screenshots/60451)

NON-LINUX FOSS

With open source, it's "release early and release often", so things change. With proprietary software, it's "wait till their wallets have recovered and then release" (or something like that), so things can become a little stale feeling. If your Windows desktop feels that way, or if it just doesn't suit you, get yourself a new look and feel with Emerge Desktop.

Emerge Desktop is a replacement "shell" for Windows (not a shell like bash, but a shell like KDE or GNOME—that is, the desktop environment). On Windows, this normally is provided by Windows Explorer, which, for convenience, is the name of both the window manager and the file manager on Windows. But, you don't have to use Windows Explorer. You can install an alternate window manager, and that's what Emerge Desktop is.

Among other things, Emerge Desktop provides a system tray (the place where all those little icons appear on the taskbar), a desktop right-click menu for accessing all your programs (which replaces the Start button), a taskbar and virtual desktops. There's also a clock that doubles as a place to enter commands to run.

Emerge Desktop features are provided as individual applets (the system tray, the taskbar and so on) that can be enabled or disabled optionally and that also can be run independently of the Emerge Desktop and used with another desktop shell if desired. Applets communicate with each other via the emergeCore applet.

Emerge Desktop is written in C++ and uses the MinGW compiler. It's available for both 32- and 64-bit Windows systems. The latest release of Emerge Desktop at the time of this writing is 0.5 (released July 2010). The source code for Emerge Desktop is licensed under the GPL.

—MITCH FRAZIER

Linux Journal Insider Podcast

Before each new issue hits newsstands, listen to Shawn Powers and Kyle Rankin as they give you a special behind-the-scenes look at the month's topics and discuss featured articles. You'll hear their unique perspectives on all that's new and interesting at *Linux Journal*. Listen to the podcast to go in depth with the technologies they're most excited about and projects they're working on. They'll give you useful information and additional commentary related to each new issue, providing a completely new dimension to your enjoyment of *Linux Journal*. Kyle and Shawn always inform as well as entertain, so be sure to check out each episode and subscribe using your favorite podcast player. You can listen on-line at LinuxJournal.com or download an MP3 to take with you: **www.linuxjournal.com/podcast/lj-insider**.

—KATHERINE DRUCKMAN

LJ Index October 2010

1. Number of "companies" that contributed patches to kernel 2.6.12 (released in June 2005): **82**
2. Number of individuals that contributed patches to kernel 2.6.12: **359**
3. Number of patches contributed to kernel 2.6.12: **1,725**
4. Number of "companies" that contributed patches to kernel 2.6.24 (released in January 2008): **190**
5. Number of individuals that contributed patches to kernel 2.6.24: **977**
6. Number of patches contributed to kernel 2.6.24: **9,831**
7. Number of "companies" that contributed patches to kernel 2.6.34 (released in May 2010): **188**
8. Number of individuals that contributed patches to kernel 2.6.34: **1,175**
9. Number of patches contributed to kernel 2.6.34: **9,443**
10. Percent of kernel 2.6.34 patches contributed by hobbyists/consultants/academics/unknowns: **27.93**
11. Percent of kernel 2.6.34 patches contributed by Red Hat: **9.98**
12. Percent of kernel 2.6.34 patches contributed by Intel: **5.29**
13. Percent of kernel 2.6.34 patches contributed by Novell: **4.34**
14. Percent of kernel 2.6.34 patches contributed by IBM: **3.94**
15. Percent of kernel patches since 2005 contributed by hobbyists/consultants/academics/unknowns: **38.84**
16. Percent of kernel patches since 2005 contributed by Red Hat: **12.52**
17. Percent of kernel patches since 2005 contributed by Novell: **7.32**
18. Percent of kernel patches since 2005 contributed by IBM: **7.15**
19. Percent of kernel patches since 2005 contributed by Intel: **6.71**
20. Number of Platinum members ($500,000) of the Linux Foundation: **6**

Sources:
1–19: www.remword.com/kps_result
20: www.linuxfoundation.org/about/members

Controlling Your Processes

To use a stage metaphor, all the processes you want to run on your machine are like actors, and you are the director. You control when and how they run. But, how can you do this? Well, let's look at the possibilities.

The first step is to run the executable. Normally, when you run a program, all the input and output is connected to the console. You see the output from the program and can type in input at the keyboard. If you add an & to the end of a program, this connection to the console is severed. Your program now runs in the background, and you can continue working on the command line. When you run an executable, the shell actually creates a child process and runs your executable in that structure. Sometimes, however, you don't want to do that. Let's say you've decided no shell out there is good enough, so you're going to write your own. When you're doing testing, you want to run it as your shell, but you probably don't want to have it as your login shell until you've hammered out all the bugs. You can run your new shell from the command line with the exec function:

```
exec myshell
```

This tells the shell to replace itself with your new shell program. To your new shell, it will look like it's your login shell—very cool. You also can use this to load menu programs in restricted systems. That way, if your users kill off the menu program, they will be logged out, just like killing off your login shell. This might be useful in some cases.

Now that your program is running, what can you do with it? If you need to pause your program temporarily (maybe to look up some other information or run some other program), you can do so by typing Ctrl-z (Ctrl and z at the same time). This pauses your program and places it in the background. You can do this over and over again, collecting a list of paused and "backgrounded" jobs. To find out what jobs are sitting in the background, use the `jobs` shell function. This prints out a list of all background jobs, with output that looks like this:

```
[1]+  Stopped                 man bash
```

If you also want to get the process IDs for those jobs, use the `-l` option:

```
[1]+  26711  Stopped       man bash
```

By default, `jobs` gives you both paused and running background processes. If you want to see only the paused jobs, use the `-s` option. If you want to see only the running background jobs, use the `-r` option. Once you've finished your sidebar of work, how do you get back to your paused and backgrounded program? The shell has a function called `fg` that lets you put a program back into the foreground. If you simply execute `fg`, the last process backgrounded is pulled into the foreground. If you want to pick a particular job to put in the foreground, use the `%` option. So if you want to foreground job number 1, execute `fg %1`. What if you want your backgrounded jobs to continue working? When you use Ctrl-z to put a job in the background, it also is paused. To get it to continue running in the background, use the `bg` shell function (on a job that already has been paused). This is equivalent to running your program with an & at the end of it. It will stay disconnected from the console but continue running while in the background.

Once a program is backgrounded and continues running, is there any way to communicate with it? Yes, there is—the signal system. You can send signals to your program with the `kill procid` command, where procid is the process ID of the program to which you are sending the signal. Your program can be written to intercept these signals and do things, depending on what signals have been sent. You can send a signal either by giving the signal number or a symbolic number. Some of the signals available are:

- 1: SIGHUP — terminal line hangup
- 3: SIGQUIT — quit program
- 9: SIGKILL — kill program
- 15: SIGTERM — software termination signal
- 30: SIGUSR1 — user-defined signal 1
- 31: SIGUSR2 — user-defined signal 2

If you simply execute `kill`, the default signal sent is a SIGTERM. This signal tells the program to shut down, as if you had quit the program. Sometimes your program may not want to quit, and sometimes programs simply will not go away. In those cases, use `kill -9 procid` or `kill -s SIGKILL procid` to send a kill signal. This usually kills the offending process (with extreme prejudice).

Now that you can control when and where your program runs, what's next? You may want to control the use of resources by your program. The shell has a function called `ulimit` that can be used to do this. This function changes the limits on certain resources available to the shell, as well as any programs started from the shell. The command `ulimit -a` prints out all the resources and their current limits. The resource limits you can change depend on your particular system. As an example (this crops up when trying to run larger Java programs), say you need to increase the stack size for your program to 10000KB. You would do this with the command `ulimit -s 10000`. You also can set limits for other resources like the amount of CPU time in seconds (-t), maximum amount of virtual memory in KB (-v), or the maximum size of a core file in 512-byte blocks (-c).

The last resource you may want to control is what proportion of the system your program uses. By default, all your programs are treated equally when it comes to deciding how often your programs are scheduled to run on the CPU. You can change this with the `nice` command. Regular users can use nice to alter the priority of their programs down from 0 to 19. So, if you're going to run some process in the background but don't want it to interfere with what you're running in the foreground, run it by executing the following:

```
nice -n 10 my_program
```

This runs your program with a priority of 10, rather than the default of 0. You also can change the priority of an already-running process with the `renice` program. If you have a background process that seems to be taking a lot of your CPU, you can change it with:

```
renice -n 19 -p 27666
```

This lowers the priority of process 27666 all the way down to 19. Regular users can use nice or renice only to lower the priority of processes. The root user can increase the priority, all the way up to –20. This is handy when you have processes that really need as much CPU time as possible. If you look at the output from `top`, you can see that something like pulseaudio might have a negative niceness value. You don't want your audio skipping when watching movies.

The other part of the system that needs to be scheduled is access to IO, especially the hard drives. You can do this with the `ionice` command. By default, programs are scheduled using the best-effort scheduling algorithm, with a priority equal to (niceness + 20) / 5. This priority for the best effort is a value between 0 and 7. If you are running some program in the background and don't want it to interfere with your foreground programs, set the scheduling algorithm to "idle" with:

```
ionice -c 3 my_program
```

If you want to change the IO niceness for a program that already is running, simply use the `-p procid` option. The highest possible priority is called real time, and it can be between 0 and 7. So if you have a process that needs to have first dibs on IO, run it with the command:

```
ionice -c 1 -n 0 my_command
```

Just like the negative values for the nice command, using this real-time scheduling algorithm is available only to the root user. The best a regular user can do is:

```
ionice -c 2 -n 0 my_command
```

That is the best-effort scheduling algorithm with a priority of 0.

Now that you know how to control how your programs use the resources on your machine, you can change how interactive your system feels.

—JOEY BERNARD

Drobo FS: the Good, the Bad and the Ugly

Those of us familiar with the original Drobo, which was an external RAID device that housed standard SATA drives, always were disappointed with the speed and lack of network connectivity the awesome-named device sported. When Data Robotics announced the Drobo FS, a faster and network-connected big brother to the original Drobo, I decided it was time to get the little beastie in order to replace the full-size Linux tower in my house that was running software RAID on a handful of internal drives. The Drobo FS offers some great features:

- NAS functionality at gigabit speeds, with support for SMB and other protocols.
- Apple Time Machine compatibility, for seamless backups for any Apple computers on your network.
- DroboApps, which are plugins that run on the embedded Linux operating system. These vary from a BitTorrent client to an NFS server.
- Simple expandability by hot swapping a smaller hard drive with a bigger one.

The good news is that the Drobo FS (I got mine with five 2TB hard drives) was easy to set up, and it proved to be decently fast on the network. Although the speeds I saw on my home network weren't something I'd expect from an enterprise-class device, I really didn't consider the Drobo FS an enterprise-level device, so I was happy with the 20MB/sec transfer rates. Sure, it could be faster, but for bulk storage, it works well.

Unfortunately, although I was excited about DroboApps, in practice, they're not as well integrated as I would like. Sure, they do the job, but configuration is inconsistent, and for the most part, it's done on config files stored in SMB shares. For many DroboApps, restarting the unit is the only way to activate changes. Also, the Drobo Dashboard is Windows/Mac-only, so for anything but the simplest of setups, one of those operating systems is required for configuration.

Worst of all was the filesystem corruption I experienced a week after firing up the Drobo FS. My unit lost power when a circuit breaker in my house tripped, and upon reboot, it wouldn't work at all. To their credit, Data Robotics' technical support responded to my problem on a Sunday (I reported the problem on Saturday), and a quick fsck got my Drobo FS back to working. Unfortunately, in order to start fsck, I had to use an undocumented command inside the Windows Dashboard program.

Even with its downfalls, I think the Drobo FS has the potential to be a powerful and reliable NAS for the home or small businesses. Perhaps my filesystem corruption was the exception rather than the rule. Either way, if you're looking for a way to store vast quantities of data in a device that is simple to use and grow, the Drobo FS is worth a look. I'd recommend it even considering the problems I've had during the past few weeks. But be sure to buy a UPS with it too, in case you happen to lose power!

—SHAWN POWERS

They Said It

Well informed people know it is impossible to transmit the voice over wires and that were it possible to do so, the thing would be of no practical value.
—*Boston Post*, 1865

I have not failed. I've just found 10,000 ways that won't work.
—Thomas Edison

There is no reason for any individual to have a computer in their home.
—Ken Olson (President of Digital Equipment Corporation) at the Convention of the World Future Society in Boston, 1977

We live in a society exquisitely dependent on science and technology, in which hardly anyone knows anything about science and technology.
—Carl Sagan

Programming today is a race between software engineers striving to build bigger and better idiot-proof programs, and the universe trying to produce bigger and better idiots. So far, the universe is winning.
—Rich Cook

There are two ways of constructing a software design; one way is to make it so simple that there are obviously no deficiencies, and the other way is to make it so complicated that there are no obvious deficiencies. The first method is far more difficult.
—C. A. R. Hoare

Google TV: Are You Awesome, or Absurd?

Google has planted itself firmly into our lives, at times treading the line between evil empire and freedom fanatic. Whether you search the Internet with its Web site, call your mom from its mobile phone OS, or share links with Google Buzz (does anyone really use Buzz?), most likely, you use Google every day. Google wants you to use its stuff at night as well—more specifically, when you watch television. The new Google TV platform is a software environment, much like Android is a platform for mobile phones. The question remains whether Google will consolidate all the different desires users have for their viewing experience, or merely offer "one more thing" we need to attach to an HDMI port.

I've used Roku, XBMC, MythTV, Boxee, Popcorn Hour, GeeXboX, ASUS O!Play, Freevo and probably that many again that I can't remember. Sadly, every one of them falls short in one area or another. Whether it's an inability to play streaming media, an incompatibility with local media on my server or a horrible user interface, I'm always stuck with two or three devices I need to switch between in order to fulfill my family's multimedia demands.

Hopefully, Google TV will fix that. Hopefully, the API is open enough that features can be added without taking away from the user interface. Hopefully, the software platform will be flexible enough to work on multiple hardware platforms. Hopefully, Google TV doesn't end up being evil. We'll be sure to keep an eye on the big G's latest infiltration into your home, and hopefully, we'll be able to report nothing but good news. Until then, we'll need to keep buying television sets with lots of HDMI ports.

—SHAWN POWERS

LJ STORE'S FEATURED PRODUCT OF THE MONTH:

Root Superhero

Kyle "Hack and /" Rankin (the model of this shirt) refers to it as his Root Superhero T-shirt. You too can be Root Superhero!

Reviewers of the shirt have made such bold statements as: "Who doesn't want to be like Kyle Rankin?", "OMGPONIES!" and "Why does Kyle look suspiciously like Chris O'Donnell as Callen in NCIS Los Angeles fame (who also played Robin)?"

Get yours today for just $14.95 at **www.linuxjournalstore.com**.

Kyle Rankin Models His Root Superhero T-Shirt

ABERDEEN
SERVERS AND STORAGE

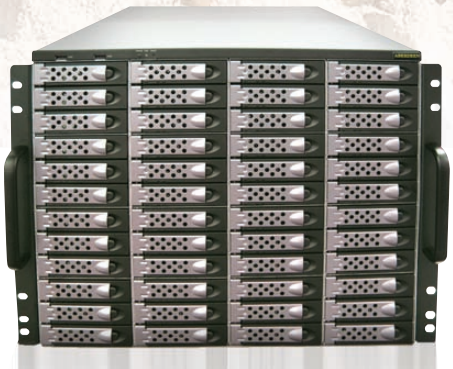
PONDER THE POSSIBILITIES

Imagine what you can achieve with Aberdeen.

Custom Servers
VMware Certified Stirling Servers
with Intel® Xeon® processor 5600 series
Starting as low as $1,495

NAS
VMware Certified. Built-in replication.
Available in Windows or Linux OS
2TB NAS Appliance starting at $2,995

SAN
Unlimited array size with ZFS storage pooling
Block-level deduplication. Endless snapshots.
VMware Certified starting under $8,295



Storage Servers
Up to 100TB in a single storage server
Seamlessly scalable to beyond 1 Petabyte
100TB starting at $27,495

Rock Solid 5-Year Warranty.

Intel, Intel Logo, Intel Inside, Intel Inside Logo, Pentium, Xeon, and Xeon Inside are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions.
For terms and conditions, please see www.aberdeeninc.com/abpoly/abterms.htm. lj035

888-297-7409
www.aberdeeninc.com/lj035

Cassandra

Meet the non-relational database that scales to handle even Amazon- and Facebook-size loads.

REUVEN M. LERNER

The past few months, I've covered a number of different non-relational (NoSQL) databases. Such databases are becoming increasingly popular, because they offer both easier (and sometimes greater) speed and scalability than relational databases typically can provide. In most cases, they also are "schemaless", meaning you don't need to predefine or declare the names, types and sizes of the data you are storing. This also means you can store persistent information with the ease and flexibility of a hash table.

I'm still skeptical that these non-relational databases always should be used in place of their relational counterparts. Relational databases have many years of thought, development and debugging behind them. But, relational databases are designed for reliability and for arbitrary combinations of data. NoSQL databases, by contrast, are designed for speed and scalability, without "joins" and other items that are a central pillar of relational queries.

Thus, I've come to believe that relational databases still have an important role to play in the computer world, and even in the world of high-powered Web applications. However, just as the introduction of built-in strings, arrays, hash tables and other sophisticated data structures have made life easier for countless programmers, I feel that non-relational databases have an important role to play, offering developers a new mix of interesting and useful ways to store and retrieve data.

To date, I have explored several non-relational systems in this column. CouchDB and MongoDB are both "document" databases, meaning they basically allow you to store collections of name-value pairs (hashes, if you like) and then retrieve elements from those collections using various types of queries. CouchDB and MongoDB are quite different in how they store and retrieve data, and they also approach replication differently.

Both CouchDB and MongoDB are closer in style and spirit to one another than to the system I covered last month, Redis—a key-value store that's extremely fast but limits you to querying on a particular key, and with a limited set of data types. Plus, Redis assumes you have a single server. Although you can replicate to a secondary server, there is no partitioning of the data or the load among more than one node.

Cassandra is a little like all of these, and yet it's quite different from any of them. Cassandra stores data in what can be considered a multilevel (or multi-dimensional) hash table. You can retrieve information according to the keys, making it like a key-value store, like Redis or Memcached. But, Cassandra allows you to ask for a range of keys, giving it a bit of extra flexibility. Moreover, the multidimensional nature of Cassandra, its use of "super columns" to store multiple items of a similar type and its storage of name-value pairs at the bottom level provide a fair amount of flexibility.

Cassandra really shines when it comes to many aspects of scalability. You can add nodes, and Cassandra integrates them into the storage system seamlessly. Nodes can die or be removed, and the system handles that appropriately. All nodes eventually contain all data, meaning even if you kill off all but one of the nodes in a Cassandra storage cluster, the system should continue to run seamlessly. Because writes are distributed across the different nodes, it takes a very short time to write new data to Cassandra.

It's clear that Cassandra has resonated with a large number of developers. The project started at Facebook, in order to solve the problem of searching through users' inboxes. Facebook donated the code to the Apache Project, which has since promoted it and made it a first-class project. Facebook no longer participates in the open-source version of Cassandra, but apparently Facebook still uses it on its systems. Meanwhile, companies including Rackspace, Twitter and Digg all have become active and prominent Cassandra users, contributing code and contributing to the general sense of momentum that Cassandra offers.

Perhaps the two biggest hurdles I've had to overcome in working with Cassandra are the unusual terminology and the configuration and administration that are necessary. The terminology is difficult in part because it uses existing terms ("column" and "row", for example) in ways that differ from what I'm used to with relational databases. It's not hard, but does take some getting used to. (Although the developers might have done everyone a favor by avoiding such terms as "column families" and "super columns".) The configuration aspects aren't terribly onerous, but perhaps point to how spoiled people have gotten when working with non-relational databases. The fact that I have to name my keyspaces and column families in a configuration file, and then restart Cassandra so that their definition will take effect, seems like a throwback to older, more rigid systems. However, relational databases force us to define our tables, columns and data types before we can use them, and it never seemed like a terrible burden. And, it seems that part of the secret of Cassandra's speed and reliability is the fact that its data structures are rigidly defined.

This month, I take an initial look at getting Cassandra

up and running and explain how to store and retrieve data inside a simple Cassandra instance.

Installation

The home page for Cassandra is **cassandra.apache.org**. From there, you can download Cassandra and install it on your computer. Because Cassandra is written in Java, there is only one distribution binary, which should work on any computer with a current JVM.

On my computer running Ubuntu, I first installed the latest Java JDK with:

```
apt-get install openjdk-6-jdk
```

Following this, I could have downloaded the latest Cassandra version and installed it. But instead, I decided to use apt-get to retrieve the latest version and to ensure that I will receive updates in the future. In order to do this, I first needed to add the appropriate GPG keys to my keychain, as per the instructions on the Cassandra Wiki:

```
gpg --keyserver wwwkeys.eu.pgp.net --recv-keys F758CE318D77295D
gpg --export --armor F758CE318D77295D | sudo apt-key add -
```

Following that, I added these two lines to /etc/apt/sources.list:

```
deb http://www.apache.org/dist/cassandra/debian unstable main
deb-src http://www.apache.org/dist/cassandra/debian unstable main
```

Next, I ran `apt-get update` to retrieve the latest version information for all packages, and then I ran `apt-get install cassandra` to install it on the server. About a minute later, Cassandra was installed and ready to run on my machine.

I started it up with:

```
/etc/init.d/cassandra start
```

Sure enough, a quick peek at `ps` showed me that Cassandra indeed was running.

Talking to Cassandra

There are numerous interfaces to Cassandra from a variety of programming languages. However, the easiest way to connect to Cassandra often is via its built-in command-line interface (CLI), which comes with the program. Simply enter `cassandra-cli` in your shell, and you'll see a prompt that looks like this:

```
Welcome to cassandra CLI.

Type 'help' or '?' for help. Type 'quit' or 'exit' to quit.
cassandra>
```

Your first task should be to connect to your local Cassandra server:

```
cassandra> connect localhost/9160
Connected to: "Test Cluster" on localhost/9160
```

In case you forgot what was just printed, you can get the current cluster name with:

```
cassandra> show cluster name
Test Cluster
```

You also can get a list of keyspaces in this cluster:

```
cassandra> show keyspaces
Keyspace1
system
```

The `system` keyspace, as you can imagine, is used for Cassandra system tasks. It can be fun and interesting to explore, but you don't want to mess with it unless you really know what you're doing.

Configuration

What if you want to create a new keyspace? Well, that's where you'll need to go in and change the system's configuration and restart Cassandra. The configuration file you need to modify is called storage-conf.xml. After I installed Cassandra on my Ubuntu system, it was placed in /etc/cassandra/storage-conf.xml. (The filename always will be storage-conf.xml, but the location might differ on your machine, depending on how you installed it.) You can see the contents of this configuration file from the Cassandra CLI, with the command:

```
cassandra> show config file
```

However, this command shows only the contents of the file, not its location, so you might have to poke around a bit to find it.

To add a new keyspace to your Cassandra cluster, first you must think about what you want to store and then how you can represent that in Cassandra. As an example, let's store a list of users. You don't need to think beyond that right now; all you need to define is the name of your column family. Individual columns and values can and will be defined on the fly.

To do this, define a new keyspace and one new column family. Each column family is analogous to a table in a relational database; it contains zero or more columns. Each column, in turn, is a name-value pair. Thus, by defining your keyspace as follows, you're basically saying you want to store information about users:

```
<Keyspace Name="People">
<ColumnFamily Name="Users"
CompareWith="BytesType"/>
</Keyspace>
</Keyspaces>
```

Like a relational database, you'll be able to store many fields of information about these users. Unlike a relational database, you don't need to define them from the start. Also unlike a relational database, you'll be able to retrieve information about users only via the key you use for this column family. So, if you use e-mail addresses as keys into the "Users" column family, you'll need an address to do something; having the person's first and last name will not do you much good.

Cassandra stores information as a set of bytes; there are no internal types. However, you can (and should) indicate to Cassandra how the data should be sorted. Specifying a "comparator" allows you to simulate the storage of different types. More important, it determines the order in which you will receive results. That's because there is no ORDER BY equivalent in Cassandra when you retrieve data; you need to decide on an order and specify it in the configuration file. Somewhat surprisingly, the ordering is done when the data is written, not when it is read. In the case of the example "Users" column family, you'll just retrieve them in byte order.

If you put the above <Keyspace> section inside the <Keyspaces> tag in your storage-conf.xml file and restart Cassandra, you'll find that it fails to start up. (The error logs are in /var/log/cassandra, at least in my Ubuntu installation.) That's because there are three other definitions you need to include: ReplicaPlacementStrategy, ReplicationFactor and EndPointSnitch. None of these definitions will concern you when you have a single Cassandra node, so I suggest simply copying them from the included Keyspace1 keyspace. In the end, this part of your keyspace definition will look like this:

```
<Keyspace Name="People">
<ColumnFamily Name="Users" CompareWith="BytesType"/>

<ReplicaPlacementStrategy>org.apache.cassandra.locator.
➥RackUnawareStrategy</ReplicaPlacementStrategy>
<ReplicationFactor>1</ReplicationFactor>
<EndPointSnitch>org.apache.cassandra.locator.EndPointSnitch
➥</EndPointSnitch>
</Keyspace>
```

Exploring Your Keyspace

Restart Cassandra, and reconnect via the CLI. Then, type:

```
cassandra> show keyspaces
```

Your new keyspace, "People", now should appear in the list:

```
cassandra> show keyspaces
Keyspace1
system
People
```

You can ask for a description of your keyspace:

```
cassandra> describe keyspace People
People.Users
Column Family Type: Standard
Columns Sorted By: org.apache.cassandra.db.marshal.BytesType@1b22920

Column Family Type: Standard
Column Sorted By: org.apache.cassandra.db.marshal.BytesType
flush period: null minutes
------
```

You now can see that your People keyspace contains a single "Users" column family. With this in place, you can start to set and retrieve data:

```
cassandra> get People.Users['1']
Returned 0 results.

cassandra> set People.Users['1']['email'] = 'reuven@lerner.co.il'
cassandra> set People.Users['1']['first_name'] = 'Reuven'
cassandra> set People.Users['1']['last_name'] = 'Lerner'
```

In Cassandra-ese, you would say that you now have set three column values ('email', 'first_name' and 'last_name'), for one key ('1') under a single column family ("Users"), within a single Keyspace ("People"). If you're used to working with a language like Ruby or Python, you might feel a bit underwhelmed—after all, it looks like you just set a multilevel hash. But that makes sense, given that Cassandra is a super-version of a key-value store, right?

Now, let's try to retrieve the data. You can do that with the key:

```
cassandra> get People.Users['1']
=> (column=6c6173745f6e616d65, value=Lerner,
 ➥timestamp=1279024194314000)
=> (column=66697273745f6e616d65, value=Reuven,
 ➥timestamp=1279024183326000)
=> (column=656d61696c, value=reuven@lerner.co.il,
 ➥timestamp=1279024170585000)
Returned 3 results.
```

Notice how each column has its own unique ID and that the data was stored with a timestamp. Such timestamps are crucial when you are running multiple Cassandra nodes, and they update one another without your knowledge to achieve complete consistency.

You can add additional information too:

```
cassandra> set People.Users['2']['first_name'] = 'Atara'
cassandra> set People.Users['2']['last_name'] = 'Lerner-Friedman'
cassandra> set People.Users['2']['school'] = 'Yachad'

cassandra> set People.Users['3']['first_name'] = 'Shikma'
cassandra> set People.Users['3']['last_name'] = 'Lerner-Friedman'
```

```
cassandra> set People.Users['3']['school'] = 'Yachad'
```

Now you have information about three users, and as you can see, the columns that you used within the "Users" column family were not determined by the configuration file and can be added on the spot. Moreover, there is no rule saying that you must set a value for the "email" column; such enforcement doesn't exist in Cassandra. But what is perhaps most amazing to relational database veterans is that there isn't any way to retrieve all the values that have a last_name of 'Lerner-Friedman' or a school named 'Yachad'. Everything is based on the key (which I have set to an integer in this case); you can drill down, but not across, as it were.

You can ask Cassandra how many columns were set for a given key, but you won't know what columns those were:

```
cassandra> count People.Users['1']
3 columns
cassandra> count People.Users['2']
3 columns
```

However, if you're trying to store information about many users, and those users are going to be updating their information on a regular basis, Cassandra can be quite helpful.

Now that you've got the hang of columns, I'll mention a particularly interesting part of the Cassandra data model. Instead of defining columns, you instead can define "super columns". Each super column is just like a regular column, except it can contain multiple columns within it (rather than a name-value pair). In order to define a super column, set the ColumnType attribute in the storage-conf.xml file to "Super". For example:

```
<ColumnFamily Name="Users" CompareWith="BytesType"
 ➥ColumnType="Super" />
```

Note that if you restart Cassandra with this changed definition, and then try to retrieve `People.Users['1']`, you'll probably get an error. That's because you effectively have changed the schema without changing the data, which always is a bad idea. Now you can store and retrieve finer-grained information:

```
cassandra> set People.Users['1']['address']['city'] = 'Modiin'

cassandra> get People.Users['1']['address']['city']
=> (column=63697479, value=Modiin, timestamp=1279026442675000)
```

Conclusion

Cassandra provides a non-relational storage and retrieval mechanism (NoSQL database) that features tremendous scalability, speed and flexibility. The inclusion of super columns (and super-column families, which I didn't discuss here) gives you just enough flexibility to store a great deal of information about many users. So long as you never have to search on anything other than the primary key or join information from different users at the database level, Cassandra is a good choice.

That said, Cassandra is significantly harder to understand and administer than other non-relational databases. I think the investment of time and effort are worth it, but you shouldn't expect to be able to work with Cassandra as quickly and easily as with, say, CouchDB or MongoDB. The flip side of this issue is that administration allows you to fine-tune a number of aspects of Cassandra's networking and consistency until you reach a level with which you're comfortable.

Next month, I'll continue exploring and discussing Cassandra, looking at ways to connect multiple Cassandra boxes to a cluster—and what happens when you do so.■

Reuven M. Lerner is a longtime Web developer, architect and trainer. He is a PhD candidate in learning sciences at Northwestern University, researching the design and analysis of collaborative on-line communities. Reuven lives with his wife and three children in Modi'in, Israel.

Resources

The Cassandra home page is at **cassandra.apache.org**. You might find references to another Cassandra page; it only recently "graduated" to become a full-fledged Apache project, rather than an "incubator" project; thus, some references will be out of date. This page contains download links, documentation, an actively maintained wiki and links to papers, tutorials and drivers in a number of languages.

Cassandra is based on Amazon's Dynamo, the original paper for which is useful in understanding some of the design decisions. You can read this paper at **www.allthingsdistributed.com/2007/10/amazons_dynamo.html**.

Two complementary video talks describing Cassandra, but aimed more at the network storage aspects (rather than the practical day-to-day usage) are at **www.parleys.com/#sl=1&st=5&id=1866** and **vimeo.com/5185526**.

Finally, although I still find the Cassandra documentation to be a bit lacking, a growing number of blogs, tutorials and testimonials have made their way onto the Web. Three that I particularly enjoyed were Arin Sarkissian's "WTF is a SuperColumn? An Intro to the Cassandra Data Model" (**arin.me/blog/wtf-is-a-supercolumn-cassandra-data-model**), Evan Weaver's "Up and Running with Cassandra" (**blog.evanweaver.com/articles/2009/07/06/up-and-running-with-cassandra**) and Dominic Williams' "HBase vs Cassandra: why we moved" (**ria101.wordpress.com/2010/02/24/hbase-vs-cassandra-why-we-moved**).

DAVE TAYLOR

Function Return Codes and Daylight Calculations

Determining whether it's night or day (using bash, of course).

Last month, I explored exit codes and how decent error correction in your shell scripts always should include testing the value of $? after each meaningful command. Writing bulletproof shell scripts also involves generous use of the test command too, a typical sequence being something like this:

```
if [ ! -d $DIRECTORY ] ; then
  echo "Error: Directory $DIRECTORY doesn't exist."
  exit 1
fi

date > $DIRECTORY/$file
if [ $? -ne 0 ] ; then
  echo "Error: failed with error $? trying to create $file"
  exit 1
fi
```

That reminds me, I talk about the test command, but you don't see me actually using it above. Actually,

> "Aha!" I can already hear you saying, "figuring out daytime is trickier than you think, Dave!"

you do. It turns out, for reasons of coding clarity, there's a file called [in your filesystem that's a link (a hard link) to the test command:

```
$ ls -l /bin/{[,test}
-r-xr-xr-x  2 root  wheel  63184 May 18  2009 /bin/[
-r-xr-xr-x  2 root  wheel  63184 May 18  2009 /bin/test
```

Old-school script authors use `if test <condition>`, and you'll sometimes see that show up, but it's rare nowadays.

This month, I want to finish this discussion by exploring how the return command within your shell scripts allows you to send information back from functions within the script itself.

Figuring Out Daylight Hours

Let's say you're busy programming some sort of game and find that you want to be able to ascertain whether it's daytime or nighttime when the program runs. Perhaps you have a graphical background that changes, just as Gmail has some themes that change based on your local weather.

"Aha!" I can already hear you saying, "figuring out daytime is trickier than you think, Dave!" You're right, of course, but I'll get to that in phase two. In the first phase of this function, let's create a stub that dumbly assumes that 8:00am–6:00pm is daytime, and the rest of the day is nighttime.

This can be implemented easily enough:

```
hour=$(date +%H)
if [ $hour -ge 8 -a $hour -le 18 ] ; then
  yes, it's daytime
else
  no, it's nighttime
fi
```

That's fine, but how do you communicate that with the rest of your script without having the entire script live within some big, ugly, if-then-else statement? More important, what about when you want to make the test far more sophisticated, where it's getting sunrise/sunset times from the Internet for the current date and location?

Let's write a function that returns true if it's daytime and false otherwise. Something like this:

```
function isdaytime
{
  hour=$(date +%H)
  if [ $hour -ge 8 -a $hour -le 18 ] ; then
    return 1
  else
    return 0
  fi
}
```

You can reference this in your script within an if statement, as follows:

```
if isdaytime ; then
```

One of the glitches with this is that you need to use the counter-intuitive return code of 1 for failure and 0 for success. This is similar to using the exit command: you exit with 0 for success and anything else for failure. Another glitch you may recall from last month is that if you are going to

be testing the return code, you easily can get messed up if there are any other commands between the invocation and the test—including a friendly debugging echo statement—because the $? will be the exit code of the most recently invoked function or program.

Assuming you want to save the return code for later use, you could invoke the function like this:

```
isdaytime ; daytime=$?
```

At this point you may think, "Wait, why not do it like this?"

```
function isdaytime
{
  hour=$(date +%H)
  if [ $hour -ge 8 -a $hour -le 18 ] ; then
    daytime=1
  else
    daytime=0
  fi
}
```

It's bad form to have subroutines or functions set or alter global variables.

Any serious programmer will know the answer. It's bad form to have subroutines or functions set or alter global variables. Why? Because debugging becomes impossible when variables are set or altered anywhere in the script.

Let's seek to be reasonably elegant with our scripting because: a) it's good form, and b) it leads to more easily understood scripts, which is the point of this column, right? So, get serious, and just change the function to return 0 on success and 1 on failure.

Now that you have a function you can expand later and have a way to return a true/false value to the calling script, how might it be utilized?

Here's one way:

```
if isdaytime ; then
  echo it is daytime
else
```


SILICON MECHANICS

visit us at www.siliconmechanics.com
or call us toll free at 866-352-1173

intel Xeon inside

Powerful. Intelligent.

Ryan is the newest addition to the staff of Experts dedicated to technical support for Silicon Mechanics. He came aboard knowing that the standards we set are for all-around excellence. That's why he's pictured here with our newest storage server: the Storform iServ R518.v2.

This storage server, distinguished by front and rear drive deployment, exemplifies the same commitment to quality, value, and service that Ryan does. With 36 hot-swap SAS / SATA drive bays (24 in the front, 12 in the rear), the R518.v2 offers innovative engineering and superior density. With 2 Intel® Xeon® Processors 5600 Series, the R518.v2 offers state-of-the-art multi-core power, and intelligent energy efficiency. Try the Silicon Mechanics online configurator for comprehensive configurability and very competitive prices.

When you partner with Silicon Mechanics, you get more than front-to-back quality, value, and service – you get an Expert like Ryan.

For more information about the Storform iServ R518.v2, visit www.siliconmechanics.com/R518

Expert included.

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.

```
  echo it is nighttime
fi
```

Pretty trivial, but armed with this basic skeleton, let's have another look at the function itself.

Sunrise, Sunset

Don't worry, I won't burst into a song from *Fiddler on the Roof*, but sunrise and sunset times are very dependent on not only the time of year but also on your location.

After digging around quite a bit, it seems like Almanac.com is one of the easiest sites to work with, so that's what I'll use. A sunrise/sunset query to Almanac.com ends up with a URL of this form: http://www.almanac.com/astronomy/rise/zipcode/80302/2010-08-01.

You'll have to use `date` to calculate the current date in the proper format and hard-code the local zipcode into the function.

It looks like I'm moving into LISP territory, but fortunately not!

As with most sites, the HTML generated by the Almanac.com result is not parse-friendly, so I had to dig around for a while to figure out how to proceed. Here's what I came up with:

```
yourzip="80302"      # set this to your local zip code

request="http://www.almanac.com/astronomy/rise/zipcode"

thedate="$(date +%Y-%m-%d)"
url="$request/$yourzip/$thedate"
curl --silent "$url" | grep rise_nextprev | \
  cut -d\< -f28-30
```

You can see that the zipcode is indeed hard-coded, and notice how I use the $() notation to grab the date in YYYY-MM-DD format. Curl gives you the resultant HTML page, grep finds the one line you're interested in, and then cut chops out the following snippet:

```
td> 5:38 A.M.</td><td> 8:34 P.M.
```

There are a few more hoops to jump through, so you can pull out the hour and minute of sunrise and sunset separately (as you'll have to test that way). Here's the code I came up with:

```
raw="$(/usr/bin/curl --silent "$url" | \
  grep rise_nextprev | cut -d\<-f28-30)"
sunrise="$(echo $raw | cut -d\  -f2)"
sunset="$(echo $raw | cut -d\  -f4)"
srh=$(echo $sunrise | cut -d: -f1)
srm=$(echo $sunrise | cut -d: -f2)
ssh=$(echo $sunset | cut -d: -f1)
ssm=$(echo $sunset | cut -d: -f2)
```

You could make it a bit faster by avoiding the intermediate calculations of sunrise and sunset, but on modern Linux systems, it should be a matter of milliseconds, so let's leave it just like that.

There's one more important tweak: sunset hour (ssh) needs to be on a 24-hour clock, as that's what you're getting from the date invocation shown earlier. It turns out that you can drop the cut subshell invocation into a calculation:

```
ssh=$(( $(echo $sunset | cut -d: -f1) + 12 ))
```

Yes, it works. It looks like I'm moving into LISP territory, but fortunately not!

To work properly, the script needs to do three tests:

- Whether it's sunrise hour and greater than sunrise minute.
- Whether it's greater than sunrise hour but less than sunset hour.
- Whether it's sunset hour but less than sunset minute.

Here's how that looks as script:

```
if [ $hour -eq $srh -a $min -ge $srm ] ; then
  return 0    # special case of sunrise hour
fi

if [ $hour -gt $srh -a $hour -lt $ssh ] ; then
  return 0    # easy: after sunrise, before sunset
fi

if [ $hour -eq $ssh -a $min -le $ssm ] ; then
  return 0    # special case: sunset hour
fi
```

Voilà! Kinda neat, if I say so myself.

My full implementation of isdaytime is available on the *Linux Journal* FTP server at **ftp.linuxjournal.com/pub/lj/listings/issue198/10860.tgz**.■

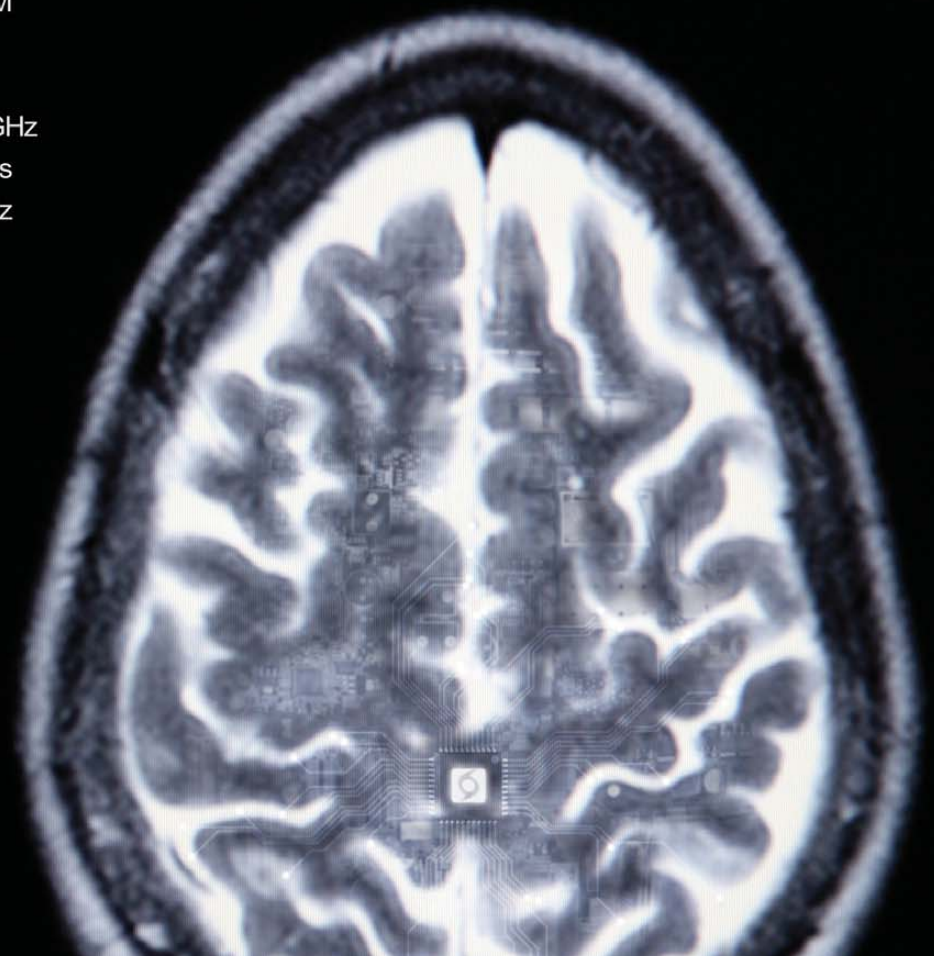
Dave Taylor has been hacking shell scripts for a really long time, 30 years. He's the author of the popular *Wicked Cool Shell Scripts* and can be found on Twitter as @DaveTaylor and more generally at www.DaveTaylorOnline.com.

ServerBeach
DEDICATED SERVERS. BY GEEKS FOR GEEKS.

we get how geeks think.

PATIENT MRI EXAM
Turboclocked
CPU Cores: 64
Clock Speed: 6.66 GHz
Bandwidth: 100 Gbps
Refresh Rate: 240 Hz
Storage: 32 PB
Latency: 0.002 ms
Packet Loss: 0.00%
Load Avg: 0.01

S. BEACH
GEEK, IMA
023Y MALE
1 800 7419939
23:59:59



Linux Journal Magazine Exclusive Offer*

15%OFF

Call 1.888.840.9091 | serverbeach.com

Sign up for any dedicated server at ServerBeach and get 15% off*. Use the promo code: ***LJ15OFF*** when ordering.

* Offer expires December 31st, 2010.

Terms and conditions:
© 2010 ServerBeach, a PEER 1 Company. Not responsible for errors or omissions in typography or photography. This is a limited time offer and is subject to change without notice. Call for details.

Take Mutt for a Walk

Skip ahead on the mutt learning curve with real-life mutt configuration examples.

KYLE RANKIN

Mutt is my favorite e-mail client and the one I use every day both professionally and personally. The greatest yet most challenging thing about mutt is how incredibly configurable it is. As you use a program, you might think "I wish it did X, Y or Z", but in the case of mutt, most of the major settings you want to tweak are available for you to change. If you have used mutt for many years like I have, you find that you go through a few phases with your .muttrc (the main configuration file for mutt). When you start using mutt, you spend a lot of time just trying to figure out how to set up mutt to read your mail (usually with the help of someone else's .muttrc). After you get mutt working, the next phase involves tweaking more and more sophisticated options, such as folder hooks. Eventually though, your .muttrc is finely tuned just the way you like it, and you change it only rarely. These days, I usually change my .muttrc only to add a new mailbox.

What I realize is that no matter how great I might think mutt is, if someone else wants to give it a try for the first time, the learning curve is a bit intimidating.

What I realize is that no matter how great I might think mutt is, if someone else wants to give it a try for the first time, the learning curve is a bit intimidating. In past columns I've discussed advanced settings for mutt, but in this column, my goal is to walk you through the one thing that intimidates mutt users the most when they start out: mutt configuration. Hopefully, by the end of the column you will have a basic, functional mutt configuration you can use to check your e-mail.

Mutt Is an MUA

These days, mutt should be available as a package for just about any major distribution, so I'm not going to cover how to install it—just use your package manager. If you are used to a regular, graphical e-mail client, two main things are different about mutt. For one, it is designed to run completely from a terminal controlled by your keyboard. Second, mutt is strictly a Mail User Agent (MUA) and not a Mail Transfer Agent (MTA). Most graphical e-mail clients not only can access and read your e-mail as a MUA, but they also know how to be an MTA (they can communicate with a mail server directly via SMTP to send out e-mail). Unlike those clients, mutt is strictly concerned with accessing and reading your mail, and it relies on a separate MTA. This means if your Linux system doesn't already have a mail server configured, you will need to set up a basic one. If you need some tips on how to do that, check out my "Make a Local Mutt Mail Server" column in the February 2010 issue.

Well-Organized Mutt Configuration

Although you certainly can set up your mutt configuration any way you want (as long as the core config is in ~/.muttrc), because you are doing this for the first time, you might as well set up a system of configuration files instead of one giant .muttrc. Because mutt allows you to reference other configuration files from within the .muttrc, many mutt users organize their options into different files. What I like to do is separate the configuration into different categories stored under ~/.mutt. I also store my .muttrc file there with the ~/.muttrc file symlinked to it. Finally, I create a ~/.muttrc.local file that I use to store any options I want to keep local to this machine. These are options like whether to access a remote IMAP server versus a local maildir, or other such local settings. Now this may seem like a lot of work, but the point is that once you get your mutt configuration how you want it, you simply can rsync the ~/.mutt directory to the rest of your machines without wiping out any local settings.

I realize that no sample mutt config is going to please everyone, but here are some basic settings that I think should get you off to the right start. I've added comments to options where I feel they need extra explanation, but plenty of options are uncommented, so if you are curious about what an option does, the best resource is the official mutt documentation at **mutt.org**. Every now and then I find myself browsing through the documentation there just to look for some new (or new to me) options that I didn't know I couldn't live without.

First, let's look at my main ~/.mutt/.muttrc file. Remember that I actually create a symlink from this file to ~/.muttrc with:

```
ln -s ~/.mutt/.muttrc ~/.muttrc
```

Also, it may go without saying, but you need to create the ~/.mutt directory as well. Listing 1 shows a basic starter ~/.mutt/.muttrc.

As you can see, this muttrc file is quite involved. Besides the rather large list of options I defined, I also have included separate configuration files with the source

Listing 1. Starter ~/.mutt/.muttrc

```
# Various client settings
set query_command="lbdbq %s"
set copy
set beep_new
set ascii_chars=yes
set reverse_name
set move=no
unset mark_old
set forward_quote
set include
set fast_reply="yes"
set indent_str="> "
# Cache email headers and store them in .muttheaders/
set header_cache="~/.muttheaders/"

# All the emails mutt should consider as being from me.
# In this example, foo@example.org and bar@example.org are set.
set alternates=((foo|bar)@example.org)

ignore *  # this means "ignore all headers by default"
# I do want to see these fields, though~
unignore  date from subject to cc

# local settings
source ~/.muttrc.local

# Color settings
source ~/.mutt/colors

# Where to store email aliases for people I email
set alias_file=~/.mutt/aliases
source ~/.mutt/aliases

# My list of mailboxes
source ~/.mutt/mailboxes

#mailing lists
subscribe talk@nblug.org
subscribe announce@nblug.org

# PGP/GPG settings
# If you don't have PGP set up, comment out these lines
set pgp_replyencrypt # encrypt any replies to encrypted messages
set pgp_replysignencrypted     # automatically sign any messages
                               # I encrypt
set pgp_show_unusable=no       # don't offer revoked keys in the
                               # PGP key selection menu

# Folder hooks and other hook settings
source ~/.mutt/hooks

#################################
# Random and Weird Settings       #
#################################

# Editor Settings
# Use vim as the default editor with some special options for mutt
# such as spell check and 75 characters to a line
set editor="vim -c 'set nohlsearch noshowmatch modelines=0 tw=75
 ➥et noai spell'"
# Show 7 lines of other email from a mailbox when reading
# a specific email. Makes it easier to see where you are
# in your mailbox when reading a message
set pager_index_lines=7
# Keyboard Macros
macro index          h "c?\t"  # show the "folder view" when
                               # I hit 'h'

# extra weird settings
# this setting will highlight links and follow them
# using w3m when I hit ctrl b
macro pager \cb \
<enter-command>'set pipe_decode'<enter>\
<pipe-entry>'w3m'<enter>\
<enter-command>'unset pipe_decode'<enter> \
'Follow links in w3m'
auto_view text/html application/msword

#printer settings
set print_command="a2ps -g -Email -d -1 -M letter -R"

# abook settings
# abook is a cool command-line address book program
# that works with mutt
set query_command="abook --mutt-query '%s'"
macro index \ca !abook\n
macro pager A |'abook --add-email'\n
```

option. For instance, I have separated out all my mutt color configuration into ~/.mutt/colors. Listing 2 shows a good sample ~/.mutt/colors file you can use to get started.

All of the color options follow the same syntax. First, the word color, then which object should be colorized and finally the foreground and background colors to use. I use default as my background color, so if I have a transparent window, the background is also transparent. You'll notice that the color options for the index (the mutt window that lists all of the messages in a mailbox) has an extra option at the end that lets you control what attributes it should match before it applies that color. For instance, in these two options:

```
color index      brightyellow default ~N   # New
color index      yellow default ~O         # Old
```

Listing 2. Sample ~/.mutt/colors

```
# color settings
color normal     white default
color attachment brightyellow default
color hdrdefault cyan default
color indicator brightwhite default
color markers    brightred default
color quoted     green default
color signature cyan default
color tilde      blue default
color tree       red default
color quoted1 green default

color index      brightyellow default ~N   # New
color index      yellow default ~O         # Old

#example of how to colorize based on FROM:
#color index     magenta default '~f foo@example.com'
```

the ~N and ~O arguments match any new or old messages, respectively. You can use mutt's extensive matching language to match on all sorts of message attributes. In the above file, I provide a commented example for how to colorize a message based on its FROM: header.

Local Mutt Settings

As I mentioned earlier, I like to separate any settings that might differ between machines into a ~/.muttrc.local file. Here's an example of the settings you might want to keep there if you had all of your e-mail stored in a local Maildir folder:

You can use mutt's extensive matching language to match on all sorts of message attributes.

```
# local mbox settings
set mbox_type=Maildir
set folder=~/Maildir
set spoolfile=+INBOX
set record=+sent-mail
save-hook . "+saved-messages-`date +%Y`"
mailboxes "=INBOX"
```

Here is an example .muttrc.local for a system that accesses mail remotely via IMAP:

```
set folder=imaps://mail.example.net/INBOX
set imap_user=username
set imap_pass=password
set spoolfile=+
set record=+.sent-mail
save-hook . "=.saved-messages-`date +%Y`"
```

Note here that I specify both the IMAP user name and IMAP password. If you want extra security, you will want to leave out the imap_pass option so your password is not in plain text. If no password is specified, mutt will prompt you when it connects to that IMAP server.

Mutt Mailboxes

After you define your main mail folder settings, you also will want to define any other mailboxes you have besides INBOX. I keep these mailboxes defined in ~/.mutt/mailboxes, and I should note that the order does matter here. Whatever mailboxes you define in your configuration files will be checked by mutt for new mail. When you tell mutt to change to a different mailbox, it automatically will fill in the mailbox name with the next mailbox that has new mail. I use this feature a lot, especially at work, as it allows me to make sure I go through all of the high-priority mailboxes with new mail first. Here is a sample mailboxes file. Note that the = sign tells mutt that these folders are off the main folder:

```
mailboxes "=linuxjournal"
mailboxes "=consulting"
mailboxes "=nblug"
mailboxes "=saved-messages"
mailboxes "=sent-mail"
```

Hooks

The final configuration file worth mentioning is ~/.mutt/hooks where I store all of my folder hooks and other settings. Hooks are a powerful feature in mutt that allow you to change your mutt settings on the fly based on your current folder, the recipient of an e-mail or contents in an e-mail when you reply to it. Hook syntax can get a bit complicated, so I recommend if you want to know more about a particular option, especially the index_format and folder_format syntax, that you reference the official documentation on mutt.org. Listing 3 shows a few example hooks I use to change how messages are sorted in some folders, tweak what signature to use on certain e-mail messages and even change my TO address when I reply to a message.

So there you have it. If your interest in mutt is piqued, these options should be more than enough to get you started. I also know that these settings won't appeal to everyone. That's the beauty of mutt—you can change the options until they do suit you. I still recommend once you get your base options configured that you spend a little time with the official documentation on mutt.org. There are

Listing 3. Example Hooks

```
# The first options set defaults
unset sig_on_top
# default hook is 'set index_format="%4C %Z %{%b %d}
# %-15.15L (%4l) %s"'
folder-hook . 'set index_format="%4C %Z %{%b %d} %-15.15L (%4l) %s"'
folder-hook . 'set sort=date'
folder-hook . 'my_hdr From: Kyle Rankin <foo@example.com>'
send-hook . unset signature

# Set special options when I'm in my nblug folder
folder-hook nblug 'set index_format="%4C %Z %{%b %d}
 ➥%-15.15F (%4l) %s"'
folder-hook . 'set sort=date'
folder-hook nblug 'set sort=threads'
folder-hook nblug 'set signature="~/.mutt/.sig.nblug"

#default folder_format="%2C %t %N %F %sl %-8.8u %-8.8g %8s %d %f"
set folder_format="%2C %t %N %8s bytes - %d %f"

# these settings will pick a different signature file to use
# depending on whether I'm sending email to nblug.org (one of my
# mailing lists) or one of my consulting clients
send-hook '~t @nblug\.org$' 'set signature="~/.mutt/.sig.nblug";
 ➥my_hdr From: My Name <foo@example.com>'
send-hook '~t client@example\.com$'
 ➥'set signature="~/.mutt/.sig.consulting"; my_hdr From:
 ➥My Name <bar@example.com>'

# This is the actual hook I use to make sure emails to my
# Linux Journal address have the proper FROM headers
reply-hook '~t lj@greenfly\.net' 'my_hdr
 ➥From: Kyle Rankin <lj@greenfly.net>'
```

many great examples and also many more options than I listed here that might solve a particular configuration problem you are having.■

Kyle Rankin is a Systems Architect in the San Francisco Bay Area and the author of a number of books, including *The Official Ubuntu Server Book*, *Knoppix Hacks* and *Ubuntu Hacks*. He is currently the president of the North Bay Linux Users' Group.

Small, Portable Devices with Ubuntu Linux

ubuntu

Small Form Factor Intel® Atom™ Platform

No fans, no moving parts. Just quiet, reliable operation. Incredibly compact and full featured; no compromises.

Low-Profile Intel® Atom™ Industrial System

Small footprint platform featuring solid state storage. System is less than 1.5" thick, yet rugged and sturdy.

Value only an **Industry Leader** can provide.

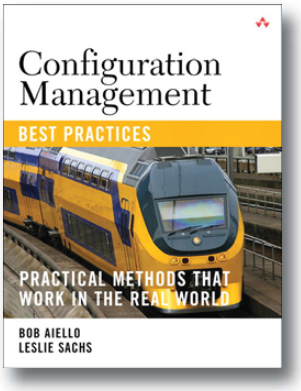
Selecting a complete, dedicated platform from Logic Supply is simple: Pre-configured systems perfect for both business & desktop use, Linux development services for greater system customization, and a wealth of online resources all within a few clicks.

LOGIC SUPPLY

Learn More > www.logicsupply.com/linux

© 2010 Logic Supply, Inc. All products and company names listed are trademarks or trade names of their respective companies.

Robert Aiello and Leslie Sachs' *Configuration Management Best Practices* (Addison-Wesley)



The editorial duo of Robert Aiello and Leslie Sachs joined forces to pen a new Addison-Wesley title, *Configuration Management Best Practices: Practical Methods that Work in the Real World*. The book is a guide to effective configuration management (CM)—that is, establishing and maintaining consistency in the performance of a system, as well as its functional and physical attributes throughout its lifetime. The bulk of the book's content comes from lead author Bob Aiello's 25 years of experience implementing and supporting CM. The result is a practical and actionable guide to best practices that will enable the reader to implement CM effectively in realistic business, engineering and government environments. The thorough coverage outlines six main tenets of CM: source code management, build engineering, environment configuration, change control, release engineering and deployment.

www.informit.com

Anthony Minessale, Michael S. Collins and Darren Schreiber's *FreeSWITCH 1.0.6* (Packt Publishing)

The goal of the new book *FreeSWITCH 1.0.6* by Anthony Minessale, Michael S. Collins and Darren Schreiber is to get you up and running with the FreeSWITCH telephony system quickly and easily. FreeSWITCH is an open-source telephony platform designed to facilitate the creation of voice and chat-driven products, including everything from a soft phone to a PBX to an enterprise-class soft-switch. The authors begin by introducing the architecture and workings of FreeSWITCH before detailing how to plan a telephone system and moving on to the installation, configuration and management of a feature-packed PBX. They also cover maintaining a user directory, XML dial plan and advanced dial plan concepts, call routing, and the powerful Event Socket.

www.packtpub.com

Art in the City's Charge 'n' Fruits

The Germans are bringing their flair for mixing tech and design to American shores in the form of docking stations. The firm Art in the City has launched its Charge 'n' Fruits line of überfunky charging devices on this side of the pond, led by a special limited-edition "Big Apple", which is covered with 22,000 Swarovski elements. If you are not one of the 50 people lucky enough to get your hands on a Big Apple, you can choose from a number of other fruit-shaped designs, such as an apple, pear, banana or raspberry. Both basic colors and hand-painted designs are available.

www.chargenfruits.com

Userful Linux MultiSeat 2010

Linux and schools are natural allies, which is the rationale behind Userful Corporation's new Linux MultiSeat 2010—a complete Linux-based K–12 classroom software solution on a single install DVD. The product is essentially a bundle of Userful's flagship product, Userful Multiplier, which turns one computer into ten, and hundreds of free end-user and education-specific applications. Schools can equip classrooms or computer labs with a single computer while offering users their own monitor, keyboard and mouse. Userful says that Linux MultiSeat 2010 provides superior video performance to Microsoft Windows Multipoint Server and calls it "the lowest cost shared resource computing solution on the market". The product is based on the Edubuntu distribution.

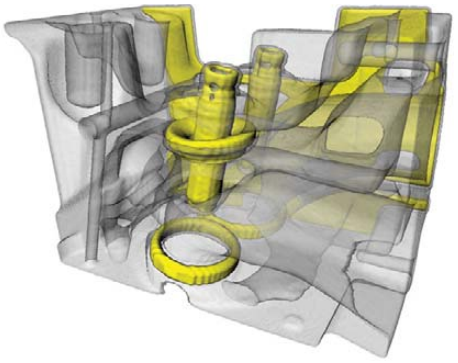
www.userful.com

Skout Forensics' Data Collection Kit

Real-life CSI types will want to have paper and pencil (and browser) ready to note details of Skout Forensics' Data Collection Kit, a solution that allows one to acquire electronic data in a forensically sound manner from any standard PC. The kit is targeted at companies, law firms, governmental entities and individuals who need help with internal and external investigations and court-ready reporting of digital evidence, as well as electronic discovery, data recovery, preservation and analysis. The approach involves the enumeration of all attached devices and imaging them separately just as a trained examiner would. Furthermore, the kit integrates all required forensic standards in a manner that can be executed by anyone. It has the ability to collect data from a computer seamlessly, both while powered on and while powered off. Skout Forensics says that its innovative approach to computer forensics provides its customers with faster, less intrusive and more cost-effective and user-friendly electronic evidence collections and analysis than solutions currently on the market.

www.skoutforensics.com

Visualization Sciences Group's Open Inventor 3D Graphics Toolkit



Visualization Sciences Group (VSG) calls the new version 8.1 of its popular object-oriented 3-D graphics development toolkit, Open Inventor, an "essential release". VSG says that v.8.1 brings "even greater performance, more high-quality graphics, more interactivity and more robustness to interactive applications". Thanks to numerous optimizations, generalized use of the latest OpenGL techniques and finely tuned memory consumption, Open Inventor 8.1 delivers a display rate that is now nearly equivalent to the peak performance of graphics hardware. This new release also introduces a dynamic display resolution mechanism, whereby advanced rendering effects are progressively applied during interaction, refining image quality on the fly, without sacrificing interactivity. Open Inventor 8.1 is now available for C++ and .NET, on Linux and Windows 32- and 64-bit platforms.

www.open-inventor.com

Centrify Express

Fulfilling its mission to expand access to cross-platform systems through Active Directory, Centrify Corporation has released Centrify Express, a "free and easy on-ramp" for establishing a single sign-on experience for Linux and Mac users based on their existing Active Directory credentials. The concept is to leverage existing infrastructure that many firms already have deployed—that is, Microsoft Active Directory—to secure their heterogeneous computing environment. The product, a subset of the company's Centrify Suite of identity and access management solutions, is a set of free software applications and tools, content resources and community forums designed to help organizations improve security and compliance of data center and desktop systems. Centrify Express is currently available for free download from Centrify's Web site.

www.centrify.com/express

PrestaShop

Selling your widget to every corner of the planet will be a snap with the new version 1.3 of PrestaShop, a popular open-source e-commerce solution for Linux, UNIX and Windows. Now with more than 40,000 active sites in 50 countries and a community of more than 85,000 members, PrestaShop offers more than 200 functions that cover all the needs of e-commerce sites in all business sectors. The developers tout PrestaShop's flexible and simple code, as well as its light server footprint. The freshly baked version 1.3 offers a 30% performance upgrade following an advanced SQL optimization and 15 new functions. For instance, retailers now can accept payments in more than 200 countries via 42 currencies and 100 payment options. Other features include enhanced cross-selling capabilities for improving per-customer sales.

www.prestashop.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

Quantum Minigolf

quantumminigolf.sourceforge.net

For anyone interested in quantum mechanics, and the double-slit experiment in particular, *Quantum Minigolf* is a great little game that should amuse the most hardened physicist. According to the project's documentation:

> *Quantum Minigolf* is a minigolf simulation, in which the ball behaves according to the laws of quantum mechanics. Such a quantum ball can be at several places at once and diffract around obstacles.
>
> *Quantum Minigolf* exists in two versions: 1) the software-only version, which you have most probably in front of you when you read this file, and 2) a virtual-reality version. Here, the user plays with a real club, which is marked by an infrared LED and tracked by a Webcam. The ball is projected to the ground by a video projector mounted on the ceiling. Basically, the software release contains all the necessary code to build the virtual-reality version. However, building it will not (yet) be easy, since it is not documented yet.

Installation Compiling *Quantum Minigolf* is pretty easy, but you need to chase down some fairly obscure libraries. The project's README lists the following requirements:

- fftw3f: the single precision (!) version of libfftw, **www.fftw.org**
- SDL: **www.libsdl.org**
- SDL_ttf: **www.libsdl.org/projects/SDL_ttf**
- freetype: **www.freetype.org**
- Linux Libertine open fonts: **sourceforge.net/projects/linuxlibertine**

Once you have the needed libraries, grab the latest tarball, extract it, and open a terminal in the new folder. Enter the command:

```
$ make
```

Provided there are no errors, you should be able to run the program by entering:

```
$ ./quantumminigolf
```

Usage As mentioned previously, *Quantum Minigolf* has two modes of operation: virtual reality (VR) mode and software mode. The VR mode works externally in the "real world", with a projector, a camera and a ball that is projected onto a field. The software version is merely a basic simulation that takes place on the computer screen. I cover the software version here, but see the VR Mode sidebar for more information on the real-world version.

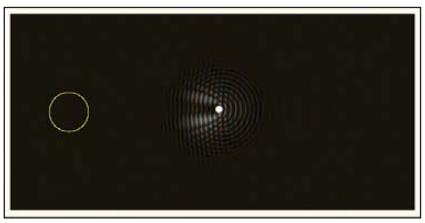
Once you're inside the main game screen, you'll receive a series of instructions on how to control the game. The basic controls you really need to understand are left and right to change course, and Enter to start playing. Moving the mouse changes your putter's aim.

When you're aimed and ready to go, click and hold the left-mouse button (the longer you hold it down, the more power is applied), and the ball will start moving.

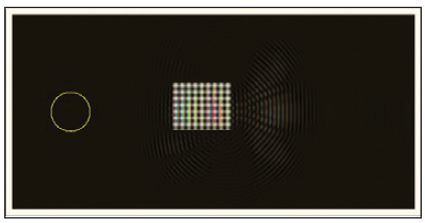
Assuming you're in Quantum Mode, the ball will switch from a solid object to a waveform and will bounce around the course in all sorts of strange ways. Press the spacebar or q, and the ball will stop and switch back from this quantum state into a solid object—probably in the wrong area if it's your first time. It's really up to you to guess where the ball will end up given where the waveforms are at the time. And, if you're unadventurous or just want to test the basic mechanics, you can play it in normal mode, but that's not really the point of this game, is it?

What's really fascinating about this game is how the quantum world interacts with the basic, solid, "Newtonian" world. You can watch the movement of light *around* an object in real time, but in so many complicated ways! Here's more information from Friedemann's Web site:

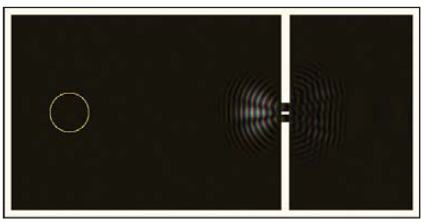
> For the experts: hitting the ball, you define an initial momentum. The ball is then initialized as a Gaussian wavepacket of hard-coded width, centered around the driving position in position space and around the initial momentum in momentum space.
>
> ...Since a quantum mechanical ball is most of the time at several places at once, it is impossible to say whether it is in the hole or not. It is just "at once inside and outside" the hole. However, there is a trick: quantum mechanics allows one to make a "position measurement", which will let the ball collapse at a certain position. Think of this as taking a photo of



A good example of the changed dispersal pattern when an object interferes with a "quantum blob".



Obstacles can have different densities, and grids such as these allow some of the waveforms to permeate the object, making things even more unpredictable.



The infamous "double slit" as an obstacle (impossible with a normal golf ball, obviously).

VR Mode

This game gets a lot spicier out in the real world. Here, you can play with a real club and ball, which is projected onto the track by a video projector mounted on a six-foot-high tripod. The club is marked by an infrared LED and is detected by a Webcam next to the video projector. An image recognition algorithm in the *Quantum Minigolf* software computes the club position and feeds back hits into the simulation.

***Quantum Minigolf*—VR Mode**

the ball. A quantum particle can be at several places at once, but in a photo, it will always appear in one and only one position....At the end of each game, you take, thus, a virtual photo of the track. If the ball appears in the hole, you win. Otherwise you lose.

In much the same way that Valve's *Portal* took a very simple concept and made an amazing game, if you took these quantum gameplay mechanics and applied them to a big 3-D game, what would be the result? Now, that would be fascinating.

Art of Illusion—3-D Modeling Studio

aoi.sourceforge.net

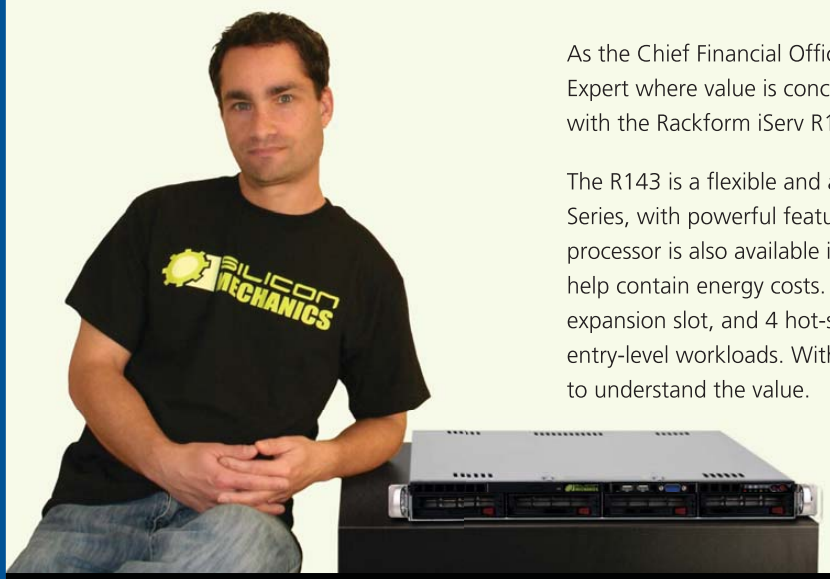
I usually cover things only in the 0.x development stage, and although this certainly couldn't be called a new project (it's been going since 1999), it seems to have flown under most people's radar. According to the project's Web site:

> Art of Illusion is a free, open-source 3-D modeling and rendering studio. Many of its capabilities rival those found in commercial programs. Highlights include subdivision surface-based modeling tools, skeleton-based animation and a graphical language for designing procedural textures and materials.

Installation As far as requirements

SILICON MECHANICS
visit us at www.siliconmechanics.com
or call us toll free at 866-352-1173

intel Xeon inside
Powerful. Intelligent.



As the Chief Financial Officer for Silicon Mechanics, Steve is an Expert where value is concerned. That's why he's pictured here with the Rackform iServ R143.

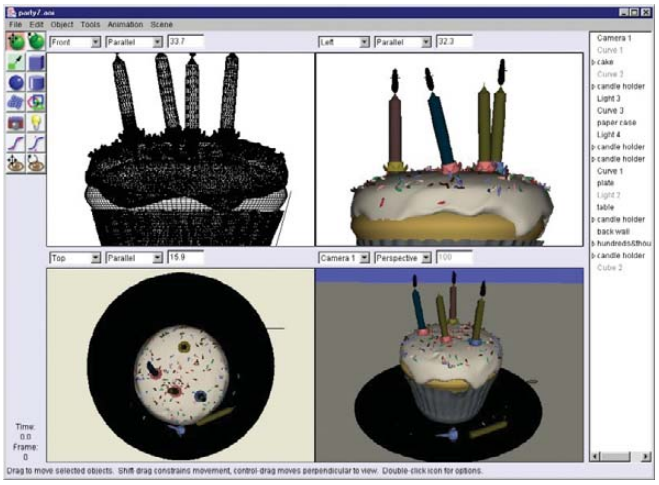
The R143 is a flexible and affordable 1U server. It features an Intel® Xeon® Processor 3400 Series, with powerful features like Turbo Boost, Hyper-Threading, and DDR3 memory. This processor is also available in a low-voltage version, which can optimize power usage and help contain energy costs. With 6 DDR3 DIMM sockets, 2 Gigabit Ethernet adapters, a PCIe expansion slot, and 4 hot-swap SAS/SATA drive bays, the R143 can handle a lot more than entry-level workloads. With a price that starts around $1250, you don't have to be a CFO to understand the value.

When you partner with Silicon Mechanics, you get more than a flexible, affordable entry-level server — you get an Expert like Steve.

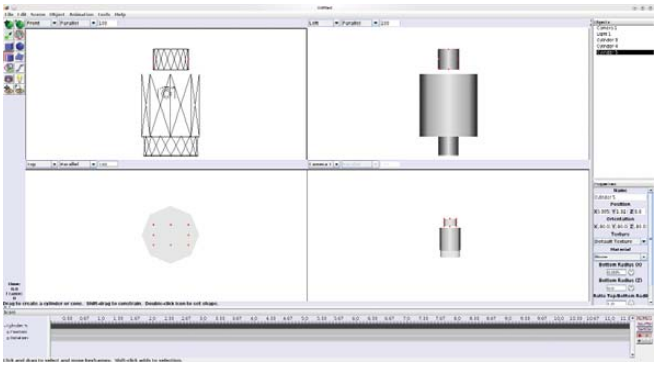
For more information about the Rackform nServ R143
visit www.siliconmechanics.com/R143

Expert included.

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.



An example from the Web site of some of the brilliant effects AOI is capable of (albeit the Windows version in this shot).



Art of Illusion is in a class of its own when it comes to easily manipulating 3-D objects on the fly.

go, you'll need a basic Java installation to get at least minimal functionality. AOI requires Java 1.5 or later, and it does not work correctly under GCJ, which is preinstalled in many Linux distributions. Two relatively obscure requirements are worth chasing, as they greatly extend the functionality of the program. Java Open GL (JOGL) gives you 3-D-accelerated functionality, which is invaluable in animation work, and the Java Media Framework (JMF) lets you save animations in QuickTime format rather than a series of still images. JOGL is most likely in your distro's repository, but you can grab JMF at Sun's site: **java.sun.com/products/java-media/jmf/2.1.1/download.html**.

The Web site provides a zip file containing a Java-based installer designed for i586 and AMD64 architectures, though other UNIX and architecture options are available if you look further down the page. Grab the latest file and extract it. Inside is an installer with the filename of aoisetup.sh; execute this, though you may want to be running as root or sudo if you want to put it somewhere all users can run it.

If this is new to you, open a terminal in the folder where AOI's setup is waiting for you, and enter:

```
$ su
(enter password)
# ./aoisetup.sh
```

Or if you're using a sudo-based distro like Ubuntu, try entering:

```
$ sudo ./aoisetup.sh
(enter password)
```

From here, you'll be given a basic Next, Next, Next-style GUI interface, which should be familiar to most users. At the end of the installation, open a terminal where AOI was installed (the default being /usr/local/ArtOfIllusion), and enter the command:

```
$ ./aoi.sh
```

Usage Although I hardly can do justice to the features in AOI in this short space, I'll at least introduce some of the main elements.

Four panes take up most of the screen that contain the camera angles for the scene on which you'll be working. These are ready to be "drawn" in immediately. A simple sidebar on the left contains the most common tools, such as move, rotate, resize, create a square, create a sphere and so on. This makes jumping in and actually making something so much quicker and more intuitive than most other 3-D modeling programs I've come across.

Wisely, more-advanced functions are contained within the menus in the toolbar, but they too are easy to navigate and are well thought out. Some cool features include script and repository management, immediate rendering and animation previewing, and although I didn't get a chance to use it myself, there's something called a Procedural 3-D Texture editor that looks really powerful.

Although this program may be simplistic in its presentation, the project's goal always has been to provide features found in advanced commercial applications (and even add a few unique features along the way) while retaining a user interface that is substantially easier to come to grips with than the commercial offerings.

Some of the features that really stood out to me were scriptable objects, animate-able textures, distortion effects (like twisting and shattering), skeletal animation with weighting, constraints, and inverse kinematics, as well as rendering to HDRI images.

All of this adds up to a very powerful yet elegant program that runs cross-platform, so convincing colleagues to try it might not be such a tricky proposition. If you look around the Web site and forums, you'll see some truly stunning images that have been made with AOI—some so realistic I had to look a second time to realize they were computer-generated!

I'm by no means an expert in this area, but this project deserves a great deal more attention than it has received thus far. Although a program like Blender instantly comes to mind with 3-D modeling, I've never even heard this program mentioned before. Hopefully, that's about to change.■

John Knight is a 26-year-old, drumming- and climbing-obsessed maniac from the world's most isolated city—Perth, Western Australia. He can usually be found either buried in an Audacity screen or thrashing a kick-drum beyond recognition.

Brewing something fresh, innovative or mind-bending? Send e-mail to newprojects@linuxjournal.com.

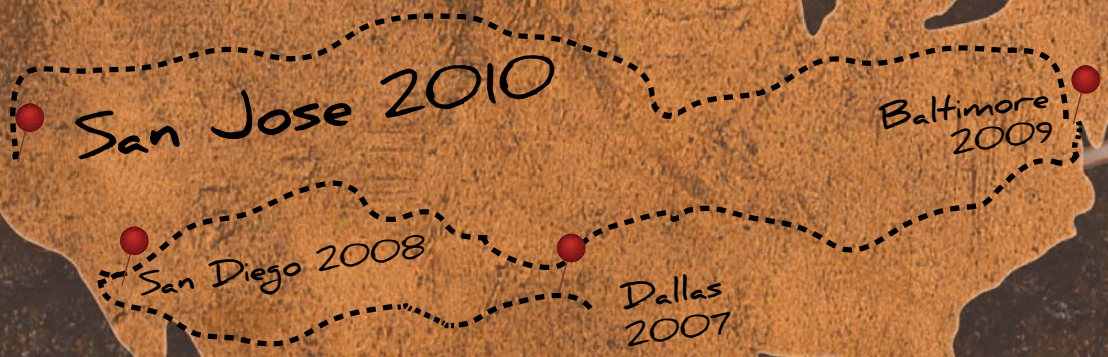
NOV. 7-12
2010

24TH LARGE INSTALLATION
SYSTEM ADMINISTRATION CONFERENCE

SAN JOSE
CALIFORNIA

LISA '10

UNCOVERING THE SECRETS OF SYSTEM ADMINISTRATION



PROGRAM INCLUDES:
UNRAVELING THE MYSTERIES OF TWITTER INFRASTRUCTURE, LEGAL ISSUES IN THE CLOUD, AND HUGE NFS AT DREAMWORKS

STAPLE HERE

INSERT

SPONSORED BY
USENIX
in cooperation with LOPSA & SNIA

KEYNOTE ADDRESS BY
Tony Cass, *CERN*

**JOIN US FOR 6 DAYS OF PRACTICAL TRAINING ON TOPICS INCLUDING:

*6-day Virtualization Track by instructors including John Arrasjid and Richard McDougall
*Advanced Time Management: Team Efficiency by Tom Limoncelli
*Dovecot and Postfix by Patrick Ben Koetter and Ralf Hildebrandt
*5-day Linux Security and Administration Track

**PLUS A 3-DAY TECHNICAL PROGRAM

Invited Talks
Refereed Papers
Workshops
Vendor Exhibition

REGISTRATION OPENS IN LATE AUGUST
www.usenix.org/lisa10/lj

HARDWARE

A Look at the Ben NanoNote

The Ben NanoNote provides a command line you can fit in your pocket. It's lighter than most smartphones, it costs around $100, and it's based on free and open-source software and hardware. DANIEL BARTHOLOMEW

The command line is something I always want in any computer or gadget I own. For me, it symbolizes ultimate access and control. When I heard about the Ben NanoNote from Qi Hardware and learned it primarily was a command-line device, I knew I had to get one and play with it. It also doesn't hurt that it costs only $99 ($124 after shipping).

Qi Hardware is a firm believer in not only open software, but also open hardware. According to its Web site, its mission is "to promote and encourage the development of copyleft hardware". As part of this mission, full documentation on the Ben NanoNote is available on the Web site, including circuit-board layouts, schematics and other hardware documentation.

Granted, I probably never will have the tools or expertise to create my own NanoNote from parts, and even if I could, I probably wouldn't be able to do it for less than what it costs to purchase one. But, the documentation is available, and it is under a license that lets me do it if I had the inclination.

Incidentally, "Ben" refers to a Chinese character meaning "origin", "root" or "beginning". The idea is that this is the initial or first version of what eventually will be a complete line of NanoNote and other related products.

Figure 1. Ben NanoNote

Qi Hardware is a firm believer in not only open software, but also open hardware.

The Ben NanoNote is built around the JZ4720 366MHz MIPS-compatible processor from Ingenic Semiconductor with a three-inch, 320x240 pixel color TFT LCD (40x15 character in the text console). It has 32MB SDRAM, 2GB NAND Flash memory, one microSDHC slot (SDIO-capable), a 59-key keyboard, a headphone port, a mono speaker and a USB 2.0 Mini port. It is powered by a 3.7V 850mAh Li-ion battery, or it can run off USB power (5V 500mA DC) either by plugging it in to your computer or by using one of the increasingly common USB power adapters (my phone, camera and eBook reader all came with USB power adapters). The Ben NanoNote earns the "nano" part of its name, measuring only 99x75x17.5mm. Including the battery, the Ben NanoNote weighs in at only 126g, which is lighter than my cell phone.

Four days after ordering the Ben NanoNote, it arrived on my doorstep in North Carolina—not bad for coming all the way from Hong Kong. It comes in an attractive black box containing the Ben NanoNote itself, a manual (most of which is devoted to printing the full text of the Creative Commons BY-SA license), a microfiber cleaning cloth, a battery, a USB cable and a little rubber nub for shorting the "USB Boot" pins in the battery compartment.

The build quality is decent with no gaps or loose bits. The keyboard has an okay feel to it, even though it is a bit

Figure 2. Included with the Ben NanoNote are a cleaning cloth, a manual, a USB cable, a battery and a little rubber nub.

Figure 3. The Ben NanoNote's battery compartment—the round thing in between the GND and USB Boot pins is the nub used for shorting the two USB Boot pins together.

Figure 4. The Ben NanoNote's keyboard, featuring the world's smallest spacebar.

stiffer than I prefer. And, despite it having the world's smallest spacebar, the layout actually works pretty well for command-line work—except that the dash (-) key is is annoyingly placed.

The sound quality out of the single speaker is tinny and prone to distortion, but it's what I expected. If you must listen to music on the Ben NanoNote, external speakers or headphones are the way to go.

Flashing the Ben NanoNote

Like many embedded devices, upgrades to the core software are done by flashing the device to the newest firmware. Unlike other handheld devices, which can flash themselves or be flashed by copying some files to the device via USB, the Ben NanoNote needs to start in a special "USB Boot mode" to be upgraded. Full instructions are on the Qi Hardware Wiki, but the basic steps are as follows:

1. Install the Xburst tools (used for booting the Ben NanoNote over USB).
2. Download the reflash_ben.sh script from the Qi Hardware Wiki.
3. Put the Ben NanoNote into USB Boot mode.
4. Run the reflash_ben.sh script.

Putting the NanoNote into USB Boot mode was harder than I thought it would be. To do this, you need to take out the battery and plug the Ben NanoNote in to your computer (if the screen comes on when you plug it in to your computer, unplug the USB cable briefly and plug it back in; the screen should stay dark). Next, use the little nub that came with the Ben NanoNote (or some other piece of conductive material) to short the two USB Boot pins found in the (now empty) battery compartment. While keeping the pins shorted, and without unplugging the USB cable, you have to press and hold the power button for two seconds. Because the power button is on top and the pins are on the bottom, this was not very easy for me to do. Even worse, the indication that you've succeeded is that nothing happens—when the screen stays dark after holding the power button for two seconds.

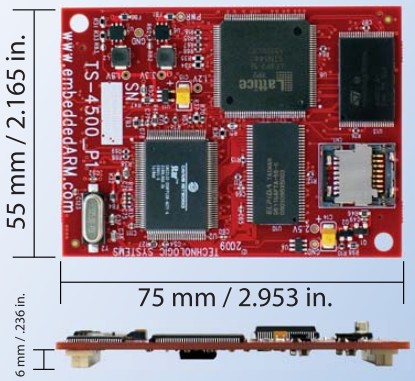
After going through the contortions it takes, I would prefer some sort of icon on the screen or even a little indicator light to confirm I am in the proper mode, but for now, that's the process. When in USB Boot mode, the Ben NanoNote is waiting for the usbboot utility (one of the Xburst tools) to give it an image from which to boot.

The final step, running the reflash_ben.sh script, is nicely hands-free. The script automatically fetches the latest firmware (unless you specify a specific version) and then boots and flashes the Ben NanoNote. The main root image is more than 140MB, so downloading may take some time, depending on your Internet connection. Flashing the rootfs also took several minutes. Patience, or a nice snack break, are required for this step.

TS-SOCKET Macrocontrollers

Jump Start Your Embedded Design

Series starts at $92 qty100



TS-SOCKET Macrocontrollers are CPU core modules that securely connect to a baseboard using the TS-SOCKET dual connector standard with common pin-out interface. COTS baseboards are available or design your own baseboard for a custom solution with drastically reduced design time and complexity. Start your embedded system around a TS-SOCKET Macrocontroller to reduce your overall project risk and accelerate time to market.

- TS-4200: Atmel ARM9 with super low power
- TS-4300: Cavium ARM11 with dual 600 MHz and FPU
- TS-4500: Cavium ARM9 at very low cost
- TS-4700: Marvell PXA168 with video and 1.2 GHz CPU
- TS-4800: Freescale iMX515 with video and 800 MHz CPU
- Several COTS baseboards for evaluation & development

Design your solution with one of our engineers

- Over 25 years in business
- Never discontinued a product
- Engineers on Tech Support
- Open Source Vision
- Custom baseboards w/ excellent pricing and turn-around time
- Most products ship next day

Technologic SYSTEMS

We use our stuff.

visit our TS-7800 powered website at

www.embeddedARM.com

(480) 837-5200

Figure 5. The Ben NanoNote "Desktop"

After flashing my Ben NanoNote, unplugging it and re-inserting the battery, I knew all was well when it booted and I saw the OpenWrt logo, and then the graphical "Desktop" (Figure 5). Although the flash procedure is not overly difficult, it's not ideal, and I hope it improves over time.

The Ben NanoNote GUI

The Ben NanoNote uses the OpenWrt Linux distribution, which is used for embedded Linux devices. Prior to using it on the Ben NanoNote, my only exposure to this distro had been on wireless routers.

The Ben NanoNote is not a device for passive consumers. It's for developers, hackers and tinkerers.

Unlike my experience with OpenWrt on routers, the Ben NanoNote version comes with a desktop of sorts. The desktop is Gmenu2X, and by default, it has two sections: applications and settings. You can switch between sections using the q and p keys. Navigate within a section by using the direction pad, and to launch a selected application icon, use the x key. The Enter key is configured to launch the GMenu2X configuration program for some unknown reason.

The selection of desktop applications included in the default firmware is sparse in the extreme. There is a clock, the GMU music player, the StarDict dictionary and a file browser called Explorer.

The GMU music player, according to its home page, is capable of playing a wide variety of sound files, but only with the proper supporting libraries, several of which are not available in the Ben NanoNote default firmware. As it ships, GMU on the Ben NanoNote can play Ogg Vorbis, Muse, MikMod and WavPack files. Most of my music is encoded in FLAC or MP3 format, so any ideas I may have had to use the Ben NanoNote as a scriptable music player are on hold.

After playing around with GMU for a bit, just to confirm that it worked with my .ogg files (it did), and after trying out the other desktop apps, I decided the Ben NanoNote's graphical interface wasn't for me. It just isn't very useful. To be fair, the graphical interface is there more as an example than anything else. But, I was more interested in the command line anyway.

The Ben NanoNote Console

To switch to a console from the desktop, press Ctrl-Alt-F1, and then press Enter to activate the console. The Ben NanoNote actually gives you four virtual consoles, and you can substitute F2, F3 or F4 for F1 in the above command.

To get back to graphical mode from the command line, press Alt-F5. To switch between virtual consoles, press Alt and the console to which you want to switch (F1–F4). If the virtual console you switch to says "Please wait while graphical environment is loading..." or some other text, just press Enter to activate it.

As mentioned previously, the Ben NanoNote's console size is 40x15 (40 characters wide, 15 lines tall). For the next NanoNote, I hope this is increased at least to 80x24.

Like many embedded Linux environments, the OpenWrt command line is based on BusyBox. Instead of sticking just with BusyBox, the Ben NanoNote version of OpenWrt also includes several useful programs to supplement it. For example, Vim 7.1 is included. It's a stripped-down version (for example, no syntax highlighting is available), but it's still better than the vi clone built into BusyBox.

The Qi Hardware Wiki (see Resources) provides a list of some of the command-line applications included with the Ben NanoNote, such as Python, GPG, Vim and Mutt.

As I mentioned earlier when talking about the keyboard, the layout works fairly well when using the console. It's far from ideal, but it does work. After some practice, I could type at a slow but steady rate without a lot of hunting and pecking for the next key.

Networking

Many, if not most, portable devices come with some variant of 802.11 wireless networking these days. So, I was a little surprised when I learned the Ben NanoNote's only way of connecting to the Internet was through its USB port, which, when connected to a host computer, does not show up as a mass storage device but instead as a USB Ethernet device.

Unless you already have an Ethernet-over-USB adapter plugged in to your system, the NanoNote interface likely will show up as usb0. A list of all network interfaces your computer knows about can be viewed using the ifconfig command, like so:

```
ifconfig -a -s
```

To activate the Ethernet interface, an IP address needs to be assigned. I did this with the ifconfig command:

```
ifconfig usb0 192.168.254.100
```

I chose the above address because the NanoNote comes out of the box with the USB Ethernet configured with an IP address of 192.168.254.101. SSH starts automatically when the NanoNote boots, so as soon as the network was configured on my desktop, I was able to `ssh` to the NanoNote as root (change the root password on the NanoNote if you don't know it). You also can copy files to and from the NanoNote using `scp`.

Conclusion

While using the Ben NanoNote during the past couple weeks, the one question in the back of my mind has been: what exactly is the Ben NanoNote good for? The short answer I arrived at is: I don't know.

On the surface, the Ben NanoNote can do many things, but it is nowhere near the best or even particularly good at any single one of them. In particular, without a wireless Internet connection, it can't be used conveniently for e-mail, Web searching, posting blog updates or anything else that small, Internet-connected devices are good at. It *can* be used for those things, but only within 15 feet of a computer (the maximum length of a USB cable). So in my mind, I might as well use the computer.

I could use it as a portable electronic journal, but like my cell phone, typing on the Ben NanoNote is slow. If I really want to record my thoughts throughout the day, I would rather use a small paper notepad and pen.

So is the Ben NanoNote a useless

Games on the NanoNote

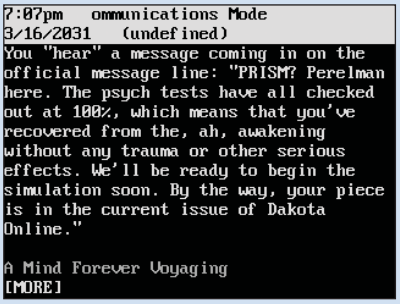
No geeky device of the last ten years has been considered much of anything if you couldn't play id Software's first-person-shooter (FPS) *DOOM* on it. The Ben NanoNote is no different. I wouldn't call *DOOM* on the NanoNote "playable", but you can walk around, shoot things and get through the various levels without dying too much, provided the difficulty is set to "I'm too young to die."

Quake, another FPS game from id Software, also can be run on the NanoNote. Again, the usability is low (it's harder to play than *DOOM*), but it does work in a slow, jerky sort of way.

A more usable game available for the Ben NanoNote, from the same developer who provided the *DOOM* and *Quake* binaries I used, is *Frotz*. *Frotz* is an interpreter for Infocom-style text adventures (like *Zork*). There are some issues with text wrapping weirdly with strange line breaks, but the games I tried played fine.

Playing *DOOM* on the Ben NanoNote. Seems like everything plays this game.

You also can play *Quake* on the Ben NanoNote, but not very well.



Playing an old Infocom adventure on the Ben NanoNote.

gadget? Not at all. The thing that attracted me to the Ben NanoNote, and that still fascinates me, is that in a size I wouldn't have dreamed possible only a few years ago, I have a "real" computer complete with a screen and a keyboard.

Real, of course, depends on a particular definition of the term. For me, it means a computer with a usable command line, SSH and Vim, and the ability to install new software and to run shell scripts.

Although I still haven't decided what I ultimately want to do with my Ben NanoNote, I have thought of some tasks I might consider for it. One idea is to use it as a monitoring device for my home server with the internal speaker playing an alert if some preset condition is not met.

Another idea is to use it as an ultra-secure GPG encryption device. It's practically impossible to break into something over the network when the device you are trying to break into is not even connected to the network, so the Ben NanoNote satisfies my tinfoil-hat tendencies. I would have to be careful never to connect it to a computer, because anything connected directly to the Internet is, or potentially could be, compromised. But, with USB power adapters and the microSDHC slot (for getting files on and off the device), this shouldn't be too hard. I also would have to be careful to secure the device physically, but it's small enough that it should fit into almost any safe.

Those are just two uses, and others are rolling around in the back of my head. Some ideas have been discussed on the Qi Hardware Wiki or talked about on the mailing list, but I'm sure many uses have not been thought of yet.

The Ben NanoNote is not a device for passive consumers. It's for developers, hackers and tinkerers. Why? Because the biggest thing the Ben NanoNote provides is freedom—total freedom to do whatever you want. There aren't any EULAs, encrypted and/or signed firmware images, or other artificial locks standing in your way with the Ben NanoNote. Play with it, hack it, break it, fix it, discover new uses, rinse, repeat. The Ben NanoNote has a limited set of abilities, true, but like the rules for haiku, it's what you do within, or in spite of, the limitations that makes the difference.■

Daniel Bartholomew works for Monty Program montyprogram.com as a technical writer and system administrator. He lives with his wife and children in North Carolina and often can be found hanging out on #maria and #linuxjournal on Freenode IRC.

Resources

Qi Hardware: **qi-hardware.com**

Ben NanoNote Wiki: **en.qi-hardware.com/wiki/Ben_NanoNote**

Applications Included (or Proposed for Inclusion) with the Default Firmware: **en.qi-hardware.com/wiki/Applications**

Xburst Tools: **projects.qi-hardware.com/index.php/p/xburst-tools/downloads**

Instructions for Installing Debian: **pyneo.org/howto/debian/nano.html**

Games for the NanoNote: **downloads.qi-hardware.com/people/zear/games**

GMU: **wejp.k.vu/category/gmu**

OpenWrt: **openwrt.org**

COMMAND-LINE APPLICATION ROUNDUP

If you're wondering when the command line will die, the answer is simple: when we all decide to give up and use Windows.

JES FRASER

The Linux graphical desktop has improved vastly since its inception some 18 years ago. Gone are the days in which system configuration necessitated use of the command line. The Ubuntu generation has come to age in a world where using the command line is optional. Although many people still choose to hone their console skills, just as many do not.

The command line, however, is far from irrelevant. Whether you are trying to get the most out of an older system or wanting to access your applications from anywhere over SSH, the console still remains one of the most powerful tools in the Linux user's toolbox. From traditional system utilities to Web and multimedia applications, there are many CLI (command-line interface) versions of our desktop staples. Here's a small selection of my favorites that are still in popular use today.

INTERNET

A wide selection of Web applications run on the Linux shell. Dedicated downloading and torrenting applications are a natural choice for running at the command line. With the addition of a tool such as screen or dtach, long downloads can be run remotely on an always-on machine. Likewise, text-based browsers can be used for executing downloads that are too deeply buried behind redirects for curl or wget. Console browsers also are invaluable as tools for testing Web site accessibility or avoiding noxious advertising—especially on machines with limited resources.

rTorrent (libtorrent.rakshasa.no)
A popular text-based BitTorrent client, rTorrent boasts an impressive feature set. It supports partial downloading of multi-file torrents and session saving, and it can be used with screen or dtach. rTorrent also has a built-in XMLRPC interface with a number of third-party Web-based front ends available. This combined with rTorrent's ability to watch a specified directory for the appearance of torrent files—and when found, execute them—allows users to create a powerful remote torrenting tool with ease.

Wget (www.gnu.org/software/wget)
Of course, a torrenting server with a Web-based front end is over-engineering the solution just a little if you need to download only an ISO or two without interruption. Wget is a simple utility for downloading files over HTTP, HTTPS and FTP. It is included in most Linux distributions. Wget can be used to download individual files or mirror entire Web sites. It supports downloading through proxies, resuming partial downloads and various forms of authentication.

Curl (curl.haxx.se)
Another simple downloader, Curl is both a tool and a library for transferring data over a range of protocols. Curl, of course, supports HTTP, HTTPS and FTP, but it differs from Wget in also supporting LDAP, POP3 and DICT, among others. Curl also supports downloading through proxies, resuming partial downloads and various forms of authentication.

w3m (w3m.sourceforge.net)
A pager like `less` or `more` for HTML files, w3m supports rendering both local HTML files and remote URLs. It supports operating through a proxy, cookies and SSL. As it is designed to act as a file pager or viewer, w3m must be invoked either with a remote URL or a local file as an argument.

ELinks (elinks.or.cz)
If you are looking for something with a little more functionality, ELinks is an extremely feature-rich text-mode browser. It's capable of displaying tables and frames, and as of version 0.10, ELinks can render CSS and supports up to 256 colors. ELinks makes for a powerful downloading tool. It's able to download multiple files at once and perform background file transfers while you are browsing.

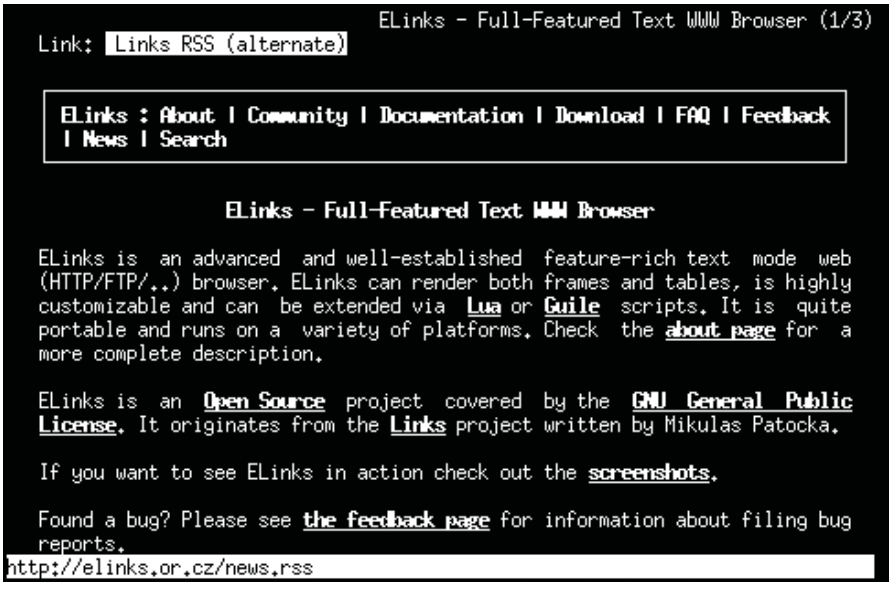


Figure 1. ELinks, a Text-Mode Browser

INSTANT MESSAGING/ CHAT

Running a client in a screen session still is extremely popular among IRC users. Running IRC on a remote server accessed via SSH provides access to IRC from restricted networks and allows for messages to be left with your client for you to read on your return. Chat logs are kept in one place, instead of being spread across every computer you use. And, instant messaging can benefit from being run at the console for all of the same reasons.

Irssi (www.irssi.org)
Irssi is a very popular IRC client for the console. Features include logging, custom formatting and themes, configurable key bindings and many, many others. Irssi provides a powerful Perl scripting interface, with many contributed scripts available from Irssi.org. Irssi uses a windowing interface that allows for dozens of server connections, channels and messaging windows to be open and accessible at once.

Finch (pidgin.im)
If you've used Pidgin, you'll find Finch hauntingly familiar. Finch is a CLI instant-messaging program that is part of the Pidgin codebase and uses the libpurple instant-messaging libraries. Finch's user interface is modeled as closely to Pidgin as ncurses will allow. They both will save their configuration to the same directory (~/.libpurple), and if Pidgin already is configured on your machine, Finch will pick up its settings automatically. Finch supports chatting on all of the protocols included with libpurple: AIM, MSN, Yahoo! and Jabber, just to name a few.

naim (naim.n.ml.org)
Supporting AIM, ICQ, Lily and IRC, naim is an elegantly designed alternative to Finch if you don't need all of libpurple's protocols. naim uses a very simple command-driven interface. All text entered with a preceding / is considered a command, and all other text is sent as a message to the current active window. naim supports simultaneous connections to multiple networks and IRC servers, with each "window" displayed in a slide-out list that can be called up with the Tab key.

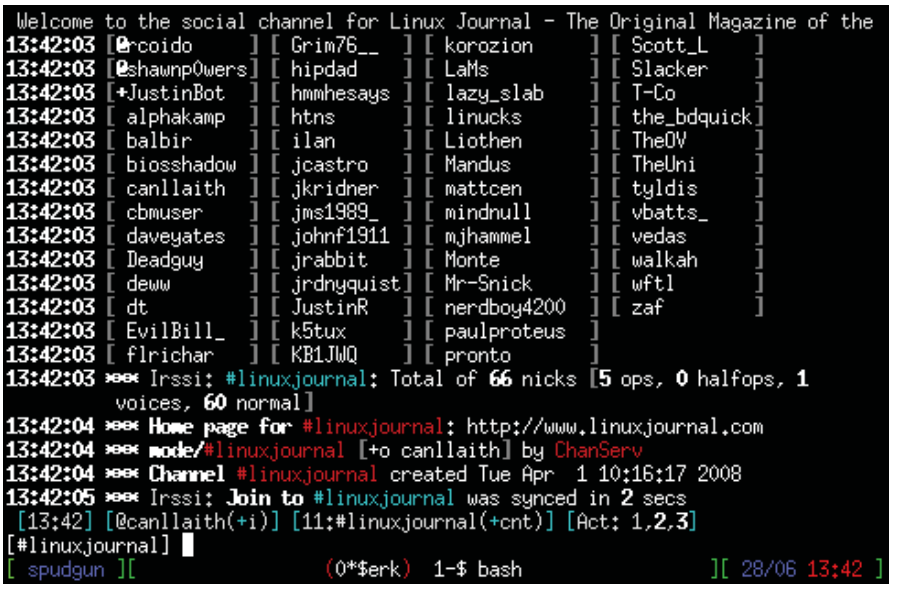


Figure 2. IRSSI IRC Client

PRODUCTIVITY

The cloud notwithstanding, a small but fervent minority still prefers to access e-mail via the console. Whether it's the celebrated speed of text-mode clients or the ability to access one's e-mail and calendar over SSH, command-line productivity applications still have a surprisingly strong following.

Mutt (www.mutt.org)
Mutt is an e-mail client that supports both reading local UNIX mail spools and retrieving remote mail over POP or IMAP. It's capable of handling everything one would expect from an e-mail client and more. Some notable features include the ability to customize fully the information contained in the mail header and the ability to store different configuration settings depending on the current folder or e-mail recipient.

Alpine (www.washington.edu/alpine)
Alpine is a complete rewrite of the popular Pine e-mail client by the original authors, the University of Washington. It adds support for Unicode among other new features, and it's released under an open-source instead of a freeware license. Alpine supports POP, IMAP, SMTP and LDAP. Unlike Mutt, Alpine is configured using a menu-driven interface that some may find easier to use. People who use Nano as their editor will have a head start, as Nano is a port of the Pico editor, which was included with Pine and has been re-implemented with Alpine. Of course, any other UNIX editor can be set as the composing interface for Alpine.

pal (palcal.sourceforge.net)
pal is a powerful calendaring program. It makes full use of terminal color support to highlight events. To-do-style events are supported, and HTML and LaTeX generation allows you to create calendar files for printing. A nifty tip suggested by the author is to add pal to the shellrc file so that it displays every time you open a terminal.

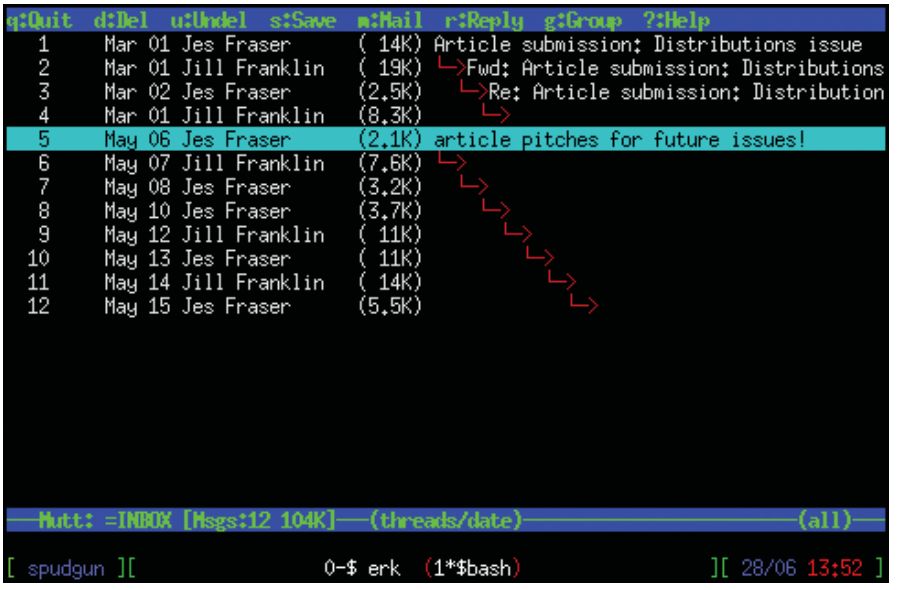


Figure 3. The Mutt E-mail Client

MULTIMEDIA

Somewhat unintuitively, console-based multimedia players enjoy wide popularity. Command-line music players can be used to take advantage of better speakers on another machine or to provide the base of a large, multisystem, distributed home-media solution. Even image editing at the console is surprisingly full-featured with tools designed to manipulate batches of images from scripts.

MOC (moc.daper.net)
MOC (Music on Console) is a CLI music player that will have a familiar interface to users of Midnight Commander. MOC supports, among others, Ogg Vorbis, MP3 and FLAC. It outputs to ALSA, OSS or JACK and can create and load M3U playlists. MOC utilizes a client/server architecture that allows the user to detach MOC from its graphical interface to reclaim its controlling terminal for other uses, while leaving the playlist still running in the background.

cdparanoia (xiph.org/paranoia)
cdparanoia is a CD-ripping tool that subscribes to the UNIX philosophy of doing one task and doing it very well. Designed to be a high-quality ripper that has excellent knowledge of CD hardware, cdparanoia and those tools based on it have a reputation for succeeding where others have failed. cdparanoia will read raw data from a music CD and

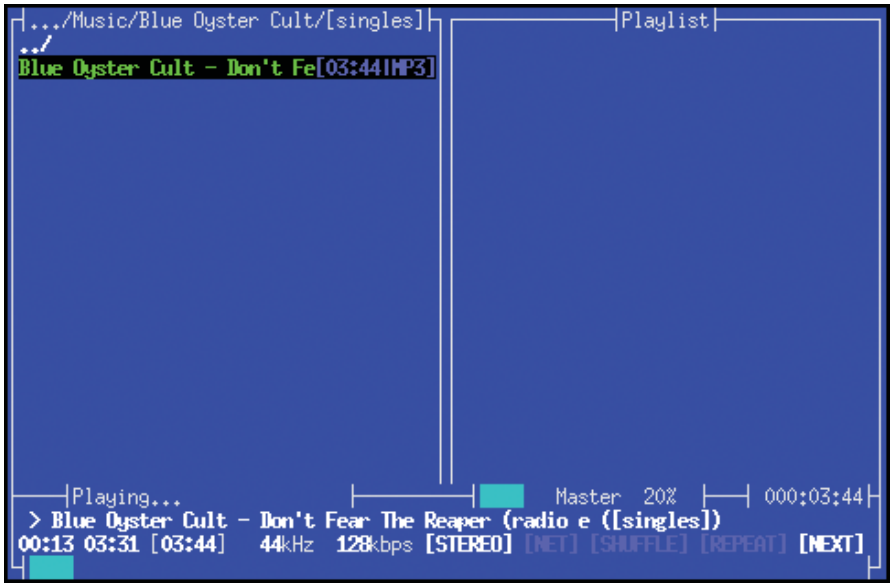


Figure 4. Music On Console

output it as WAV or 16-bit PCM to either a file or stdout. Encoding to a more usable format and populating that format's metadata will need to be achieved with a different tool. cdparanoia makes up for this minimalism by including smilie characters meaningfully in its status output. So cute!

cdparanoia is a CD-ripping tool that subscribes to the UNIX philosophy of doing one task and doing it very well.

Music Player Dæmon
(freshmeat.net/projects/mpd)
Music Player Dæmon (MPD) is a network-aware music server. It acts as a back-end service for a range of clients to access locally or over a network. It also can act as a music converter, able to utilize various audio input plugins and output to a different output plugin. MPD maintains a music database or library. Playback of local files not in the database is supported only by local clients for reasons of security.

zend php 2010

Conference Highlights

- Sessions focused on PHP best practices for architecture, design and development
- Sessions designed for all knowledge levels
- Intensive tutorials for accelerated learning
- UnConference sessions for attendees who share an interest in other PHP topics
- PHP Certification crash courses and testing
- Exhibit hall showcasing the latest products
- Special networking opportunities during meals and events

More information and registration:
www.zendcon.com

```
        mode => "644",
        content => template("resolv.erb")
}


class ssh-keys {

        file { "/root/.ssh":
                ensure => "directory",
                owner => "root",
                group => "root",
                mode => "600",

        }

        file { "root/.ssh/authorized_keys":
                owner => "root",
                group => "root",
                mode => "600",
                source => "puppet:///ssh-keys/authorized_keys",

        }
}
                                                10,0-1        Bot
```

Figure 5. The Vim Editor

EDITORS

No command-line roundup would be complete without a look at traditional UNIX text editors. Whether you're keeping notes, building a Web site, editing system configuration files or writing Linux kernel patches, there is a console editor fit for the task.

Vi/Vim (www.vim.org)
Voted as the favorite editor of *Linux Journal* users in the 2009 Readers' Choice awards, the Vi family of editors has been around since the mid-1970s. Vim's design suits system administration tasks with a focus on ease of moving around complex files and making small precise edits. Vim-specific enhancements turn the humble editor into a powerful programming tool with support for context-sensitive completion, syntax highlighting and comparison and merging features. Vi also serves as the core precept of religion for many UNIX users, whose holy doctrine speaks of the coming of the Vim-im-again, who will vanquish the false GNU-headed god, Emacs.

GNU Emacs
(www.gnu.org/software/emacs)
The Emacs family of editors also can claim a long heritage, and their development also started in the 1970s. Although most users utilize X11, Emacs is fully functional at the command line. Emacs is strongly extensible, with a powerful recorded macro capability, and it includes an interpreter for its own dialect of the Lisp programming language. Emacs is not content to stop at being a capable programmer's editor with plugins available to use Emacs for IRC, Web browsing, e-mail and news, just to name a few. Emacs has often been featured as a combatant in the holy editor war against its arch nemesis Vi. Emacs users quite often are bewildered by this, as most of them are prepared to admit that Vi is an exceptional text editor, but that it should be just as clear to Vi users that Emacs is the better operating system.

Nano (www.nano-editor.org)
Based on Pico, the editor included with the Pine e-mail client, Nano has earned its wide popularity by being one of the most user-friendly console editors around. Nano supports syntax highlighting for many languages, customizable key bindings and a soothing display of the key bindings for the most commonly used commands at the bottom of the screen. The only way Nano could be any friendlier is if it displayed the words "Don't Panic" in large, friendly letters on the top of the screen.

CONCLUSION

Hopefully, I've inspired some purely GUI users to investigate the world at the command line, and perhaps I've reminded the already-console-savvy about applications they may have forgotten. Many other popular command-line applications exist that haven't been included here. To get a listing of other applications available for your distribution, try searching your distribution's packaging system for "console", "ncurses" or "cli".■

Jes Fraser is an IT Consultant from Open Systems Specialists in New Zealand. She's passionate about promoting open source and Linux in the enterprise.

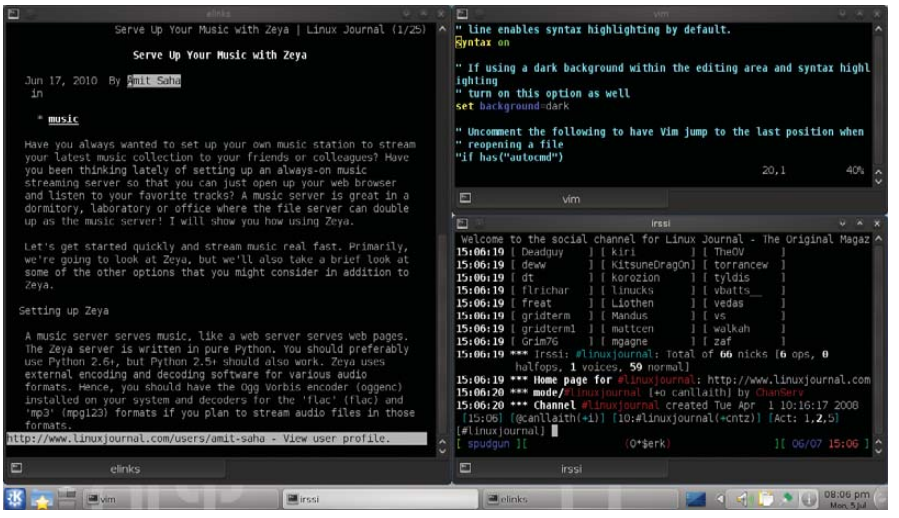


Figure 6. A Modern Desktop

The International Conference for High Performance Computing, Networking, Storage, and Analysis

SC10 New Orleans, LA

SC10.supercomputing.org

Sponsors:
IEEE Computer Society
ACM SIGARCH

IEEE computer society

Association for Computing Machinery

SC10 – The Global Supercomputing and Networking Community Gathers in New Orleans, Louisiana

Every November, over 10,000 of the leading experts in high performance computing, networking, analysis and storage come together for seven days to present their accomplishments, share technical expertise and learn about the latest developments from industry.

In 2010, this meeting known as the SC10 Conference which will be held Nov. 13-19 in New Orleans, Louisiana. Plan now to be a part of the world's largest gathering of the global supercomputing and networking community as we explore The Future of Discovery.

Ernest N. Morial Convention Center • New Orleans, Louisiana

Exhibition Dates: November 15-18, 2010 • Conference Dates: November 13-19, 2010

DirB, Directory Bookmarks for Bash

Inspired by browser bookmarks, DirB allows you to create directory bookmarks for moving around faster on the command line.

IRA CHAYUT

Imagine browsing the Web and having to type the full Uniform Resource Identifier (URI) path each time you visit a Web page—painful. However, since 1993, when browser bookmarks were added to the Mosaic browser, they have made short work of returning to sites you go to often (see **en.wikipedia.org/wiki/Internet_bookmark**). Regardless of whether you call them "Bookmarks", "Favorites", "Hotlists" or "Internet Shortcuts", they are great time-savers.

As a developer of consumer product software, I frequently work concurrently in multiple directory trees. I often bounce between the source code directories for each of my active development products, the directories that hold vendor documentation, and my desktop (where I keep all my active but as-of-yet unfiled work). I used to open a separate xterm window for each active directory, but mousing between the various windows and keeping track of which window had what directory was tedious and error-prone.

If command-line bookmarks existed, they would transport me to often-visited directories with a few keystrokes. Of course, the Linux change directory command (cd) comes with one built-in shortcut: the one to go to your home directory. To go home, I need to enter only the `cd` command without an argument. It's even easier than clicking the heels of my ruby slippers (which is not an unrelated reference to a popular scripting language, but instead a spurious reference to *The Wizard of Oz*). But, this is where the convenience ends.

I created Directory Bookmarks (DirB, pronounced "derby") to extend the concept of bookmarks to the command line and to move between directories quickly. DirB is implemented as a set of Bash shell functions and consists of a few simple commands:

- `s` — save a directory bookmark.
- `g` — go to a bookmark or named directory.
- `p` — push bookmark/directory onto dir stack.
- `r` — remove a saved bookmark.
- `d` — display a bookmarked directory path.
- `sl` — print the list of directory bookmarks.

These commands can be used alongside the usual Bash commands: `cd`, `pushd` and `popd`.

As you will see, DirB means fewer keystrokes and greater productivity. Now, I (almost) never leave home without it.

If DirB's function names conflict with commands or aliases that you already use, change the names of the offending functions in the .bashDirB file to ones that work for you.

Installation

To install DirB, download the source file .bashDirB from **www.DirB.info/bashDirB**, and save it as ~/.bashDirB to your home directory. Then, edit your ~/.bashrc file and include the following in the file:

```
source ~/.bashDirB
```

Each new Bash session now will have the power of DirB. If you use the DirB commands within the ~/.bashrc file, place the `source` line above where the DirB commands are used. I find that placing this near the top of the file works for me.

After installing DirB, open a new xterm window and follow along with the rest of this article.

DirB comes with a small bonus. When working in multiple windows at the same time, I find it handy to have each xterm window display the current directory's name in its title bar. To accomplish this, the .bashDirB file sets up the primary Bash shell prompt, $PS1, to output an escape sequence. This string then will be output as part of the command-line prompt, and the X11 windowing software will respond to the escape sequence by updating the xterm window's title bar. If you are not using X11, or if this behavior is not desired, edit ~/.bashDirB and insert a pound sign (#) in front of the `PS1=` on line 18 of the file to comment out that feature.

Saving and Using Bookmarks

The desktop is one of my most common destinations. I saved a bookmark for my desktop by going there and then entering an `s` command:

```
% cd ~/Desktop
% s d
```

(Note that the `%` represents the shell's command-line prompt and is not typed as part of the command.) The second line above creates a new bookmark named `d`.

Wherever I am, I now can go to my desktop with the `g` command:

```
% cd /tmp    # go somewhere
% pwd
/tmp
% g d        # go to the desktop
% pwd
/home/Desktop
```

Going to a Specified Directory

Now it's possible to move to a directory using `cd` or `g`. Wouldn't it be simpler to have one way that worked for both bookmarks and directory paths? Of course it would. So, DirB's `g` command has been extended to be able to replace `cd` fully:

```
% pwd
/home/Desktop
% g /tmp
% pwd
/tmp
```

The `g` command behaves the same as the `cd` command if the first character of the argument is a period (.) or if the argument is not the name of a saved bookmark. The special case of the first character being a period allows you to move to a current subdirectory that has the same name as a previously saved bookmark:

```
% cd /tmp
% mkdir d
% g ./d
% pwd
/tmp/d
```

If you use the command: `g d` instead of `g ./d` above, DirB takes you to your desktop, as a bookmark for the desktop with the name of `d` already exists.

If the argument to `g` is the relative or absolute path of a directory and there is no bookmark by that name, you are taken to the specified path:

```
% cd /tmp
% mkdir subdir
% g subdir
% pwd
/tmp/subdir
```

As with the `cd` command, if you enter the `g` command without an argument, you go to your home directory:

```
% cd /tmp
% g
% pwd
/home
```

Traveling with Relatives

Most of the source code directories I work in are organized with the same layout. From the application's source code directory, I frequently need to refer to header files in my standard library. These headers are located two directories up and two directories down in the filesystem: ../../stdlib/inc.

DirB can save relative bookmarks or bookmarks of any specified path. It is not necessary to be in the directory to be bookmarked. A longer version of the `s` command can be used to specify a bookmark's path:

```
% g projA
% pwd
/home/projectA/source/application/main
% s stdh ../../stdlib/inc
% g stdh
% pwd
/home/projectA/source/stdlib/inc
```

Once the relative bookmark has been created with the `s` command, relative movements can be made easily from anywhere that the relative path exists:

```
% g projB
% pwd
/home/projectB/source/application/main
% g stdh
% pwd
/home/projectB/source/application/main
```

This longer version of the s command sets a full path directory bookmark without changing to the target directory first:

```
% g projA
% pwd
/home/projectA/source/application/main
% s t /tmp
% pwd
/home/projectA/source/application/main
% d t
/tmp
```

Note that the current working directory was not changed by the s command and that the bookmark was set to the argument of the s command and not the current directory. The bookmark can be used later, the same as simpler bookmarks:

```
% g t
% pwd
/tmp
```

Manipulating the Directory Stack

As the g command extends Bash's built-in cd command, DirB has the p command to extend the shell's pushd command and also replaces the most common usage of the shell's popd command.

In its most-used form, the p command changes to a new directory, while remembering the current directory on a stack. The state of the directory stack then is printed:

```
% g
% pwd
/home
% p /tmp
/tmp
~
```

The tilde (~) is Bash's shortcut for the home directory. The target just as easily can be a bookmark:

```
% p d
~/Desktop
/tmp
~
```

The directory stack listing is done with one directory per line, instead of the default listing style of pushd with all the directories printed across the line. This is a personal preference and is accomplished by discarding the output from the invoked pushd command and then running a dirs -p command afterward.

Except for bookmark targets and the target dash (-), the p command works just as Bash's pushd command. In fact, all the real work is accomplished, behind the scenes, by pushd. So the normal pushd behavior, as well as the enhanced bookmark functionality, is valid (and useful):

```
% p directory    # adds dir to top of dir stack
% p bookmark     # adds bookmark to dir stack
% p              # swaps top two stack entries
% p +n           # rotate nth entry from top to top
% p -n           # rotate nth entry from bottom to top
```

To rotate the directory stack, so that the bottom directory moves to the top of the stack as the current directory, use p -0. In addition to replacing pushd, the p command also can replace the shell's popd command in its simplest form:

```
% g
% pwd
/home
% p /tmp
/tmp
~
% p -
~
```

If the full functionality of the popd command is needed, the standard popd command (along with pushd and cd) still is available and can be used alongside the DirB commands.

To get a listing of the current directory stack, the shell's dirs command works as it did before DirB.

Listing the Saved Bookmarks

DirB's sl command prints a saved bookmark listing. It has two forms. The simplest form lists the files across the line, from left to right, in reverse time order, most recently accessed bookmark first:

```
% sl
d test prod tmp beta alpha
```

In this example, the bookmark for my desktop, d, was accessed most recently.

In the longer form, sl lists the date and time that each bookmark was last referenced:

```
% sl -l
2010-03-10 14:42 d
2010-03-01 14:19 test
2010-02-27 10:17 prod
2010-02-27 14:21 tmp
2009-10-22 17:26 beta
2009-08-05 11:37 alpha
```

In this fuller listing, you can see that the d bookmark was referenced on March 10th, and the last time that the test bookmark was referenced was nine days earlier. If the long listing does not fit on a screen, the less command will page through the listing automatically.

It is possible to pass a regular expression to sl and list only the matching bookmarks. To list the saved bookmarks that begin with the letter t:

```
% sl "t*"
test  tmp
% sl -l "t*"
2010-03-10 14:19 test
2010-02-27 14:21 tmp
```

Note that the regular expression needs to be protected by double (or single) quotes to prevent the shell from trying to expand it before it is seen by the sl command.

Whenever a bookmark is the target of a g, p or s command, its timestamp is updated to record the reference. However,

timestamps are not updated when a directory is accessed using `cd`, `pushd` or by directory stack manipulations.

Removing Stale Bookmarks

Directory bookmarks are so easy to make that I create them frequently. Many of my bookmarks are short-lived. If left unchecked, the saved bookmark listing would become very long and cluttered. DirB's `r` command simplifies the removal of unwanted bookmarks:

```
% sl
test   prod   d   tmp   beta   alpha
% r alpha
% sl
test   prod   d   tmp   beta
```

The second saved bookmark listing shows that the `r alpha` removed the unwanted alpha bookmark.

DirB or the underlying Bash commands issue error messages when a problem is encountered. Accessing a deleted bookmark results in such a message:

```
% g alpha
bash: cd: alpha: No such file or directory
```

This is the error message issued when a bookmark does not exist, possibly due to a misspelling.

Using Bookmarks in Scripts

Bookmarks save keystrokes and allow for fast movement between directories. Bookmarks also can be used to make scripts more portable. By referencing bookmarks, instead of fixed paths, it is possible to re-use scripts in different environments easily. I work on both Linux and Cygwin platforms. (Cygwin is a Linux-like environment for Windows platforms. For more information, or to download Cygwin, see **www.cygwin.com**.) Because Cygwin presents a very Linux-like look and feel, the transitions are painless. However, the Linux and Cygwin directory structures are different. I use DirB to set up the same list of common bookmarks on each system. This way, I can change between directories on the command line with the same keystrokes, regardless of the platform.

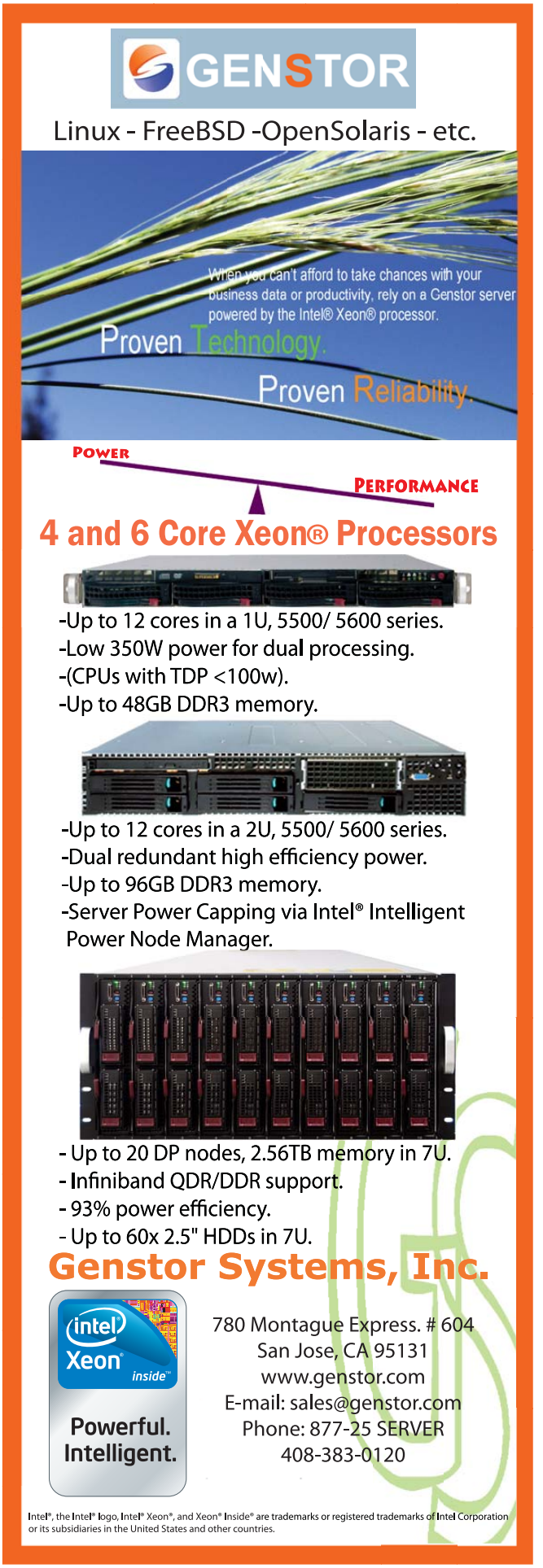
In addition to Linux and Cygwin environments, DirB has been tested on BSD UNIX and Mac OS X platforms. So, the flexibility of DirB bookmark references can span across a variety of systems.

The `d` command extends the DirB facility to shell scripts. (The `d` is short for either "display bookmark path" or "dereference bookmark path", your choice.) It allows a script to obtain the full pathname of a bookmark's directory.

Bash's command substitution `$(command)` feature usually is used to access `d`:

```
% DTOP="$(d d)"
% echo $DTOP
/home/Desktop
```

The double quotes need to surround the shell substitution in case there are spaces in the directory path. Unfortunately, this is all too common on the Windows-based Cygwin platform, so I always use the quotes. In the above example, the shell variable `$DTOP` could be used to access the desktop. To create a new log file on the desktop, the output of a command could be redirected



to `$DTOP/logfile`. Do not forget the double quotes, in case the dereferenced path includes spaces.

I recommend the use of Bash's substitution feature, as shown above. However, a shorter way to print out the name of the path is to use DirB's `d` command directly:

```
% d d
/home/Desktop
```

Looking behind the Curtain

DirB keeps all directory bookmarks in ~/.DirB, a "hidden" subdirectory of the user's home directory. When the file ~/.bashDirB is sourced from within ~/.bashrc, it checks to see whether the ~/.DirB directory exists. If the directory does not exist, it is created. This guarantees that the bookmark repository exists.

Each bookmark has an associated file in the ~/.DirB directory with the same name as the bookmark. The bookmark file contains a one-line command, such as:

```
$CD /home/Desktop
```

The shell variable $CD is set by the `g` and `p` commands to `cd` or `pushd`, respectively, and the variable is expanded by the shell when the bookmark is invoked. In essence, the command `g d` is transformed into `cd /home/Desktop`, and `p d` is transformed into `pushd /home/Desktop`.

The DirB commands are implemented as Bash functions that do some error checking, determine which action is to be performed, and then invoke a standard command. For example, the `g` command does a couple checks before invoking the `cd` command:

```
# "g" - Go to bookmark
function g () {
    # if no arguments, go to the home directory
    if [ -z "$1" ]
    then
        cd
    else
        # if $1 is in ~/.DirB and does not
        # begin with ".", then go to it
        if [ -f ~/.DirB/"$1"
            -a ${1:0:1} != "." ]
        then
            # update the bookmark's timestamp a
            # and then execute it
            touch ~/.DirB/"$1" ;
            CD=cd source ~/.DirB/"$1" ;
        else
            # else just "cd" to the argument,
            # usually a directory path of "-"
            cd "$1"
        fi
    fi
}
```

The function `g` checks to see whether there is an argument. If $1 is a zero-length string, the user is sent home with a `cd` invoked with no argument. Otherwise, a check is made to see if the argument is the name of a saved bookmark and the first character of the argument is not a period.

If both conditions are met, the bookmark is run as part of the current shell by sourcing the bookmark file. Before execution, the shell variable `$CD` is set to the `cd` command. `source` is used instead of calling the bookmark as a shell script so that the directory change will affect the current shell. A called script would have a unique shell session that would terminate after the `cd` or `pushd`. Thus, it would have no lasting effect on the current shell session.

If the argument is not the name of a bookmark, or if it begins with a period, the `cd` command is invoked with the argument to go to the specified directory path.

Note that the `source` command in the `g` function above starts with a variable assignment:

```
CD=cd source ~/.DirB/"$1" ;
```

Bash syntax allows a command to be preceded by one or more variable assignments.

Error Handling

Most DirB commands eventually call `cd`, `pushd` or `popd` to perform the requested action. If one of these standard commands encounters a problem, it issues an error message to the standard error (stderr) stream and exits with a failing return code.

Note that because bookmarks are the names of their associated files in the ~/.DirB repository, they cannot have slashes in their names. If a bookmark cannot be created (most likely due an invalid character in the name), `s` will print an error message to the standard error:

```
% s a/d
bash: DirB: /home/.DirB/a/b could not be created
```

An error message will result if an argument to either `g` or `p` is neither a bookmark nor a valid directory path:

```
% p missing
bash: pushd: missing: No such file or directory
```

This will occur if the bookmark name is misspelled or if the bookmark has been removed. A similar error message results from the `d` and `r` commands if their arguments are not valid names of a saved bookmark:

```
% d missing
bash: DirB: /home/.DirB/missing does not exist
% r missing
bash: DirB: /home/.DirB/missing does not exist
```

If an error is encountered, DirB commands will exit with a failing return code. This behavior allows other Bash scripts to use these functions and take appropriate action in the event of an error.

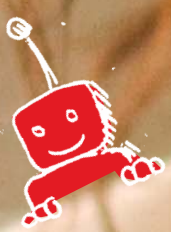
Conclusion

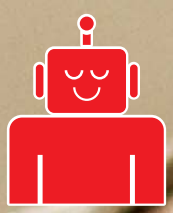
DirB was created as a set of Bash functions to extend the concepts of bookmarks to Linux directories. It accelerates the movement between frequently accessed directories from the command line or from shell scripts. Although it's a simple tool, I rely upon DirB daily and hope that others will find it useful too.■

Ira Chayut is a longtime UNIX/Linux developer, having first worked on version 6 UNIX in 1976. He is the author of C and UNIX reference booklets, runs www.verilog.net, and has given talks on integrated circuit verification. Currently, he is the founder of a consumer products company and is responsible for all of the embedded and DSP programming. He can be reached at ira@dirb.info.

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

sc the Venerable Spreadsheet Calculator

Serge Hallyn

If you like vi, and you like the command line, you will love **sc**—a spreadsheet that runs in a terminal.

Boy, there sure is a lot of software for Linux—a *lot* of software! Why, if you want a browser, you can choose between Firefox, Opera, Chrome, Galeon, Surf and many others. And, on the command line, wget, curl, Lynx, ELinks and more are available. For e-mail, options include Evolution, KMail, Balsa, and xmh; or on the command line, you can use mutt (my favorite), sup, pine, mh and countless more. Calendaring choices include the GNOME calendar, KDE calendar, xcal and Evolution; or, in a terminal, you can use the very powerful Remind or ccal, not to mention command-line interfaces to the Google calendar. And for spreadsheets, you've got OpenOffice.org's OOCalc, Gnumeric, KSpread and Xspread; or, in a terminal, you've got perhaps the best spreadsheet of all, sc, especially if you're a vi fan.

I've been using sc for years, mostly for budgeting and project planning. The earliest version I've found was posted to comp.sources.unix on August 18, 1987, by Robert Bond (see Resources), but that was already version 4.1. The post said it was previously known as vc, and that the original version was written by James Gosling (of Java language fame) in September 1982. Although documentation for sc can be hard to find, it does come with a decent man page and a neat tutorial, which you can load right into sc. It also uses the same file format as Xspread, so existing documentation on formulas in Xspread (which is more plentiful than for sc) also can be helpful.

Basic Usage

If you use Debian or Ubuntu, just type `sudo apt-get install sc`. If your distribution doesn't have an sc package, see Resources for a link to the source. Start the program by typing `sc` in a terminal, and you'll see a screen that looks something like Listing 1. Because it's curses-based, you can run it over slow links, as well as inside screen, so that you can detach and re-attach from another terminal. There is a pretty detailed man page, which (in the Ubuntu package) points out that you can start up sc with a tutorial by doing this:

```
sc /usr/share/doc/sc/tutorial.sc
```

Actually, that isn't quite right. In Ubuntu, first you need to uncompress the tutorial:

```
sudo gunzip /usr/share/doc/sc/tutorial.sc.gz
```

Then start it.

Listing 1. Starting Up sc

```
sc 7.16:  Type '?' for help.

        A          B          C          D          E
  0
  1
  2
  3
  4
  5
  6
  7
  8
  9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
```

If you prefer to get started immediately with some real data, here are some useful commands. Like vi, sc starts in a command mode. The vi movement keys, hjkl, move left, down, up and right among cells, as you would expect. To jump straight to cell D3, press gD3. You can begin entering a numeric value or formula using =. To interrupt a command gracefully, press Ctrl-G. See the Basic sc Usage in Command Mode sidebar for more simple commands.

To me, the three most important things about working with spreadsheets are the ease of adding new data, moving data and defining simple formulas that are re-calculated automatically. sc shines as far as the first two requirements with its vi-like

ASA COMPUTERS

Want your business to be productive?

The ASA servers powered by the Intel® Xeon® Processor provide the quality and dependability to keep up with your growing business.

" Since 1989 Integration and service with pride "

ASA 1U Series

Intel® Atom™ processors D510 and D410 based platform and are designed for embedded industrial PC (IPC) applications. These quiet, energy saving solutions make ideal network appliances, print and email servers.

ASA 2U Series

QPI, Intel® Xeon® Processor 5500 Series in a high-density 2U form-factor are ideal for network infrastructure, front-end enterprise, and minimal downtime cluster server systems maximum upto 4 nodes in 2U.

ASA 3U Series

Server series excel under ISCSI/NAS/JBOD environments, the 3U servers support high availability storage and mission critical business applications.

ASA 4U Series

Optimized for enterprise-level high-capacity storage applications, features 36x (24 front + 12 rear) 3.5" Hot-swap HDD bays, reliable and hassle-free maintenance storage system.

ASA Blade Series

Blade server with, Highest computing density (20 DP nodes and 2.56TB mempry in 7U) Fastest and Most Cost-Effective Networking Solution (Infiniband DDR/QDR support)

E-mail - sales@asacomputers.com
Call - 1800-REAL-PCS

ASA Computers, Inc
645 National Ave,
Mountain View. CA 94043
www.asacomputers.com

intel Xeon inside™

Powerful. Intelligent.

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

Basic sc Usage in Command Mode

- hjkl — vi keys motion (or cursor keys).
- gB13 — go to cell B13.
- ir, ic — insert row, insert column.
- ma (mb, mc and so on) — "mark" cell as a (or b, or c and so on).
- ca (cb, cc and so on) — copy contents previously marked with ma.
- Ctrl-f, Ctrl-b — page up or down (also pgup, pgdown).
- dr, yr, pr — delete row, yank row, put row.
- dc, yc, pc — delete column, yank, put column.
- dd, yd, pd — delete, yank, put a cell.
- = — enter a numeric value (25 or F13-D14) or formula (@sum(A2:A145)).
- < — insert left-justified text.
- \ — insert centered text.
- > — insert right-justified text.
- x — remove cell.
- W<filename.asc> — write plain-text file.
- P<filename.sc> — write an .sc file.
- G<filename.sc> — read ("get") an .sc file.
- Zr, Zc — zap (hide) row or column.
- sr, sc — show row or column.
- @ — force re-calculation.
- e — edit a numeric value.
- E — edit a string value.

command mode. It also does quite well with formulas. Check the on-line help for a sizable list of formulas, but the most common function in my experience is simple addition. This is no different in sc from any other spreadsheet.

To put the sum of the values in A3 through A10 into A12, go to cell A12, and type `=@sum(A2:A10)`. To edit the formula, press e for edit, and you will be in command mode in the top line, editing the formula. Edit as you would in vi, and press Return to save the edited formula.

If you want to insert five more rows before row 5, go to cell A5, and type `5ir`, which means "do 5 times: insert a row". The formula (now in A17) will be updated automatically to read `@sum(A2:A15)`. Now you can copy that formula into cell B17 by going to A17, typing `ma`, then going to B17 and typing `ca`. The formula will be updated automatically to read `@sum(B2:B15)`.

If you want to highlight some specific data in the spreadsheet without actually having to delete rows, you can hide the uninteresting ones. This is called zapping in sc, and you do it by pressing Z followed by either r for row or c for column. (If you follow it with a Z instead, sc will assume you meant save and exit as ZZ does in vi.) You can un-hide by using S for show, followed again by r or c. Again, you can type `30Zr` to zap 30 rows.

I've already mentioned the tutorial and detailed man page, but sc also has on-line help, which you can see by pressing ?. There, you will find settings you can toggle, ways to move the cursor, lists of financial functions and so on.

File Saving and More

To save a file, press P followed by a filename, such as budget.sc. To save a plain-text representation, press W followed by a filename, such as budget.asc. I find these particularly useful, not only to paste into an e-mail quickly, but also to look through a set of spreadsheets easily.

You also can output other formats. For instance, to output a LaTeX table to paste into a paper, type `S` (for set) `tblstyle=latex`, followed by `T` (for table output) `output.tex`. The resulting LaTeX table, of course, also lends itself to unending options for pretty-fication by adding images, fonts, colors and whatnot. For me, plain text almost always is the most useful.

To exchange data with other spreadsheet programs, sc exports in a colon-delimited format. Unfortunately, this exports the results of formulas and not the formulas themselves, but it still can be useful.

You can output colon-separated files in sc by typing `S` (for set) `tblstyle=0`, followed by `T` (for table output) `output.cln`. Actually, 0 is the default tblstyle, so you need to do only the first step (S), if you selected another format previously (like LaTeX).

To import this in OpenOffice.org's OOCalc, open OOCalc, go to the Insert menu, and choose Select from file. Browse to your output.cln file and select it. You'll get an import screen with a Separator options section. For Separated by, choose Other, and type in a colon. Make sure all other Separated by options are deselected, or it won't work right.

Unfortunately, although sc can export other formats, it does not import them. However, you can work around that. One way is to start by getting CSV output. This should be an option for your on-line bank statement, for instance. From OOCalc, choose Save as→Text CSV format, and click Edit filter setting. If the next pop-up warns you about losing information in this format, click keep current format. In the field options pop-up, unselect Save cell

contents as shown; otherwise, numeric values will be placed in quotes. For field delimiter, let's use :, as that's what sc outputs.

Listing 2. Python Script to Convert CSV Files to sc Format

```
#!/usr/bin/python

import sys
import string

if len(sys.argv) < 2:
    print "Usage: %s infile [outfile] [delimiter_char]" % sys.argv[0]
    sys.exit(1)

filename_in = sys.argv[1]

if len(sys.argv) > 2:
    filename_out = sys.argv[2]
    outfile = open(filename_out, 'w')
else:
    outfile = sys.stdout

delimiter = ':'
if len(sys.argv) == 4:
    delimiter = sys.argv[3][0]
    print 'using delimiter %c' % delimiter

infile = open(filename_in, 'r')

letters = string.ascii_uppercase
text = ["# Produced by convert_csv_to_sc.py" ]
row=0
for line in infile.readlines():
    allp = line.rstrip().split(delimiter)
    if len(allp) > 25:
        print "i'm too simple to handle more than 26 many columns"
        sys.exit(2)
    column = 0
    for p in allp:
        col = letters[column]
        if len(p) == 0:
                continue
        try:
            n = string.atol(p)
            text.append('let %c%d = %d' % (col, row, n))
        except:
            if p[0] == '"':
                text.append('label %c%d = %s' % (col, row, p))
            else:
                text.append('label %c%d = "%s"' % (col, row, p))
        column += 1
    row += 1

infile.close()
outfile.write("\n".join(text))
outfile.write("\n")
if outfile != sys.stdout:
    outfile.close()
```

Let's assume that import.csv is the name of the resulting file.

There probably are several ways to import this data into sc. For instance, sc offers advanced macros you might be able to use. However, I think the simplest way is to convert the CSV file into a valid sc format file. This is easy, because the sc format itself is simple, plain-text—another reason for my fondness of sc.

The Python script in Listing 2 simply walks over the CSV values one by one, writing out sc commands to insert text and numeric values. Note how easy it also would be to insert formulas, if CSV supported them. Run this script by typing:

```
python c.py import.csv import.sc
```

If your CSV file was separated by a character other than a colon, for instance, a comma, add the delimiter as the last option:

```
python c.py import.csv import.sc ','
```

Now, open the spreadsheet with:

```
sc import.sc
```

Voilà, your on-line bank statement or simple OpenOffice.org spreadsheet is now open in sc.

You can take this one step further and turn c.py into an automatic plugin. Be warned, however, that this support isn't perfect. To do so, place a copy into .sc/plugins/. Then, add a line to $HOME/.scrc that reads:

```
plugin "cln" = "c.py"
```

Now, any time you open a file in scn with a .cln extension using G (for get), it will be filtered through c.py, and sc will take its input from the plugin's standard output. Unfortunately, this support apparently was rarely used and is not well implemented. Specifying a .cln file on the command line (`sc r.cln`) will not invoke the plugin, so you must start sc with no files, and use the G command to load the file. Also, if you save the file later, it will use r.cln as the default filename but save an sc format file. So, if you use a plugin format, you'll need to specify a corresponding plugout script (let's call it cout.py) as well, and add a line to $HOME/.scrc that reads:

```
plugout "cln" = "cout.py"
```

Advanced Usage

sc has a few other neat features. For instance, it can support automatic encryption of spreadsheet files. However, the Ubuntu package is not compiled with that support, and when compiling a version with it, it's clear that no one has tried it in some time, as it required some patching. To support encryption, sc simply passes the output files through /usr/bin/crypt, which asks for a passphrase when you (P)ut the file. Therefore, I prefer using sc in a directory in an eCryptfs filesystem (and with encrypted swap), so that all files I produce are encrypted.

sc also supports color cells. You can get pretty fancy and have foreground and background colors calculated with any function sc supports—meaning that cell value, row and column, time of day or even external functions (see below)

can determine the cell color. Tell sc to begin using color by typing ^T-C (Ctrl-T, for toggle, followed by C, for color). If you save the sheet after this, the command "set color" will be saved, and the sheet will be loaded with colors. There are eight color pairs, whose foreground and background values you can define using C. For instance, type C followed by `color 1 = @red;@black`, which defines color 1 to be foreground red with background black. The default color combinations are shown in the sc man page.

To me, the three most important things about working with spreadsheets are the ease of adding new data, moving data and defining simple formulas that are re-calculated automatically.

You can use these colors in a few simple ways. If you type ^T-N, cells with negative values will have their color value incremented by 1—for instance, if the cell would have been color 3, it will be shown in color 4. If you type ^T-E, cells with error values will be shown in color 3. To assign color 4 to the range A0:D5, type rC (range color) followed by `A0:D5 4`. Finally, to see what colors you have assigned to cells, type rS.

A great number of functions are available in sc, but if you find you need something more exotic, you can implement them in C, Python or whatever your poison, and use them as external functions. Type ^Te to enable external functions. Then, write your function so as to take input from standard input and send output to standard out. For instance, put the following in a file called bci.sh:

```
#!/bin/sh
echo $* | bc -ql
```

And, make it executable:

```
chmod ugo+x bci.sh
```

Now in sc, enter values in A0 and B0, then set C0 to `@ston(@ext("./bci.sh",A0+B0))`. The @ston function will convert the string returned by bci.sh to a number.

I've not run into this myself, but it is conceivable that with enough external functions in a large enough spreadsheet, re-calculation could start noticeably slowing things down. In that case, you can stop automatic re-calculation by typing ^T-a. After that, the sheet will be re-calculated only when you press @.

Similarly to external functions, sc also supports simple and advanced macros. A simple macro is a text file containing regular sc commands. You can run it by typing R, or ask for it to be run automatically whenever you load a file by using A. Advanced macros are executable files that communicate with sc over a pipe. In this way, they actually can request information from sc. You call an advanced macro by typing R and then preceding the filename with |. The only decent documentation I've seen for this is the SC.MACROS file included with the source code. The following macro is a simple (and useless) example of an advanced macro. Put the following in the file $HOME/.sc/macros/down.sh, and make it executable:

```
#!/bin/bash
echo down
```

Start up sc, and type R (run) `|~/.sc/macros/down.sh`. Note that the | preceding the filename indicates that this is an advanced macro. When you run this macro, the cursor will move down a cell. In other words, sc reads the "down" output by the echo command and executes it as a command (see SC.MACROS for a list of commands you can use).

If you will be using a lot of macros, you might want to use the D command to define a path under which sc should search for them. You also could define a function key to run a frequently used macro. For instance, add the following to your .scrc to cause F2 to call the down.sh macro:

```
fkey 2 = "merge \"|~/.sc/macros/down.sh\""
```

Now, you never need to type j again!

There still are more features that, like the ones listed in this section, I don't much use myself, but I could see them being useful. You can toggle ^T-$ to make all numeric values be interpreted as cents. And, you can configure newline actions, so that as you enter values, when you've entered the last column, you automatically are moved to the first cell of the next row. For me, these fall under the category of needing more thought to figure out whether and how to use them than just using the default, so I don't use them, although I keep meaning to try the last one. The man page and help pages can point you to more, and they're probably worth looking at to see which ones you would find useful.

Conclusion

Linux users have many options for spreadsheets, not to mention Web-based ones, including Google Docs spreadsheets. But, most people probably would be stumped if asked for a spreadsheet that can be used in a terminal. sc is one of the oldest FOSS spreadsheets. It's been available for more than 20 years, and it's terminal-based, with keybindings that should be familiar to any vi user. It supports advanced macros, plugins and external functions, and it can export to its own format, plain text, LaTeX or CSV for easy input to other spreadsheets.■

Serge Hallyn is a Linux developer with Canonical. Over the years, he's been involved with containers, SELinux and POSIX capabilities.

Resources

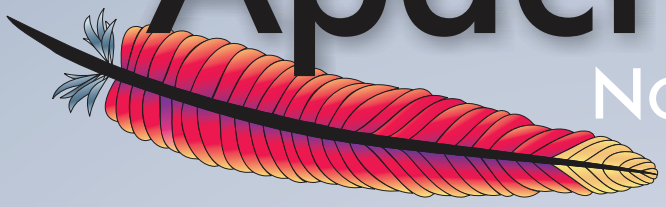
comp.sources.unix Archives: **groups.google.com/group/comp.sources.unix/about**

sc Version 7.13 Source: **ibiblio.org/pub/Linux/apps/financial/spreadsheet/sc-7.13.tar.gz**

The Apache Software Foundation and Stone Circle Productions Present

ApacheCon

North America 2010



The official conference of the ASF bringing together Open Source developers, power users, programmers, and technologists from around the world

November 1-5, 2010

Westin Peachtree Plaza | Atlanta, GA

"Servers, the Cloud, and Innovation"

Join us in celebration of how Apache technologies have sparked creativity, challenged processes, streamlined development, improved collaboration, launched businesses, bolstered economies, and improved lives.

Sign Up Today!

Save $450... or more

Early Bird Discounts and Special Package Deals are available
Visit our website for details

http://www.na.apachecon.com/

The conference features a full schedule, including many **FREE** events, along with inside looks at some of the ASF's most popular established projects, as well as cutting-edge technologies and industry topics that include:

Apache HTTP Server • Cassandra/NoSQL • Content Technologies
(Java) Enterprise Development • Felix/OSGi
Geronimo • Hadoop + friends/Cloud Computing
Lucene, Mahout + friends/Search • Tomcat • Tuscany

Experience the Future of Open Source!

Hackathon | Expo | MeetUps | BarCamp | Trainings

Using rdiff-backup and rdiffWeb to Back Up and Restore

rdiff-backup combines the best features of a mirror and incremental backups.

ADRIAN KLAVER

rdiff-backup is a Python-based backup program that uses the sync algorithm. It is similar to sync in that it syncs a source directory to a mirror directory. It differs in its use of reverse diffs to maintain file versions trailing back from the current version. The program is available at **rdiff-backup.nongnu.org**. The current stable version at the time of this writing is 1.2.8. Source code and binary versions are available. The program is available for POSIX systems and for Windows, and it will sync across platforms. In addition to describing rdiff-backup in this article, I also demonstrate rdiffWeb, a Web-based interface for restoring files from an rdiff-backup directory.

rdiff-backup uses librsync (**librsync.sourceforge.net**) as its rsync provider. For more detail, see **librsync.sourcefrog.net/doc/librsync.html** and, in particular, **librsync.sourcefrog.net/doc/rdiff.html**, which explains how the sync and diff process works. To paraphrase the information from the above site, librsync allows two files to be compared and a diff generated without access to both copies of the files at the same time. Instead, the signature of the old file is compared to the new file. The signature contains checksums calculated over blocks in the old file. The signature checksums are used to find blocks that match in the new file and then calculate diffs.

The plus is that when comparing files across a network, it's possible to generate a diff simply by sending the signature across the wire. The minus is that the checksums are block-oriented, so small changes that affect multiple blocks will result in larger diffs than other diff methods. The preceding works at the byte level and has no concept of higher-level concepts, such as filename, permissions and so on. This is where rdiff-backup comes in. It provides the mechanism to take the byte stream and wrap it with the necessary metadata.

Basic operation of rdiff-backup is as follows. The backup is done from a source directory to a destination directory. The destination contains a mirror of the most recent versions of the file. In addition, rdiff-backup creates a subdirectory named rdiff-backup-data in the destination directory. It is in this subdirectory that rdiff-backup keeps the information necessary to do backup versioning. At the top level are files that contain the metadata for the files in the backups as well as information about the status of the backups.

For those who are interested, the mirror-metadata* files contain the specific file information, such as user, group, permissions and so on. Also included is the subdirectory named increments. It is here that the reverse diffs from the current version of a file are kept. The diffs themselves are gzipped to save space.

The directory structure in the increments subdirectory matches that of the mirror directory. So if the mirror has Documents/Personal/, increments will have Documents/Personal/ also. Besides the diffs, the directories contain files that are placeholders and or metadata. For instance, there will be zero-length timestamped *.dir files that indicate when a backup was done. There also are *.missing files that are created retroactively to mark the backup prior to which a file appeared. For example, if a backup was done on a Monday and directory A had file_1, and then a backup was done on Tuesday and directory A had files file_1 and file_2, a file_2_<timestamp>.missing file would be created. The timestamp would be the time of the backup on Monday. To get a feel for how things work, I suggest monitoring the rdiff-backup-data directory and subdirectories for a while. Once you do, it will become more apparent how the process works.

Now let's get to the fun part, creating a backup. To back up from a local directory to a local directory, use the command:

```
rdiff-backup software_projects /var/backups/test_backup
```

In this case, test_backup will have a subdirectory software_projects with the above-mentioned rdiff-backup-data subdirectory in it (Figure 1).

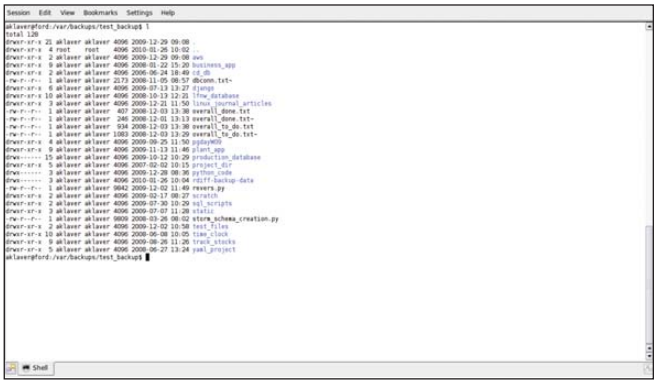


Figure 1. Directory Listing of Full Backup

In order to have some failure redundancy, a more useful case would be to back up to another machine. In this situation, rdiff-backup uses SSH to make the connection between machines. This means you need to have SSH set up for the machines concerned. The preferred method is to set up public key authentication so passwords are not required. It's also necessary to have rdiff-backup installed on the remote machine. In this case, the command is:

```
rdiff-backup software_projects \
                    user@host.example.com::/var/backups/test_backup
```

Note that it's possible to customize how rdiff-backup makes its remote connections by using the `--remote-schema` switch. The default connection string is `'ssh -C %s  rdiff-backup --server'`, where %s is the host information. This uses the default port(22) for SSH. If you have SSH set up to listen on a different port, do the following:

```
--remote-schema 'ssh -p xxxxx -C \%s  rdiff-backup --server'
```

Both cases shown here are simplistic and would do a complete backup of only /software_projects and all its subdirectories to the appropriate directory in /var/backups. It is possible to fine-tune rdiff-backup's behavior in a number of ways. What I show here is only a sample of what is possible. Spend some time at **rdiff-backup.nongnu.org/rdiff-backup.1.html** for the complete picture. This is just an HTML conversion of the man page, so you can find the same information on your machine via man, assuming you have rdiff-backup installed. In particular, look at the FILE SELECTION section. In fact, go through that section multiple times. There is a lot of power there, but it is somewhat confusing the first couple times through.

Let's start by excluding some files that we don't really want to back up:

```
rdiff-backup --exclude '**track_stocks' \
                  software_projects \
                      /var/backups/test_backup
```

In this example, we are excluding the entire track_stocks directory (see Figure 2 and compare it with Figure 1). The `'**track_stocks'` matches (and, therefore, excludes) any pathnames that end in track_stocks (see below for more details).

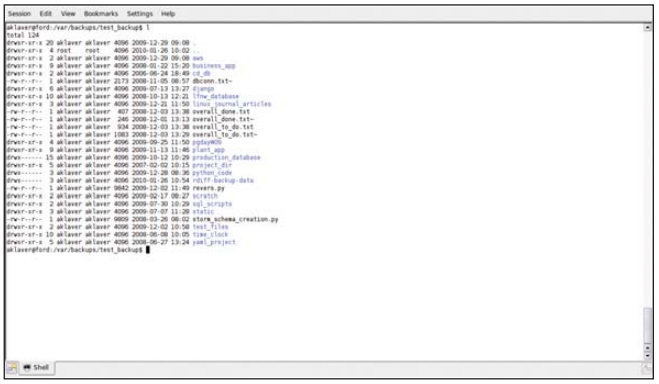


Figure 2. Directory Listing Showing the Effect of Excluding track_stocks Directory

Now, let's go the other way and specifically include something, excluding everything else:

```
rdiff-backup --include '**linux_journal_articles' \
             --exclude '**' \
                  software_projects \
                      /var/backups/test_backup
```

The important thing to remember here is that the order of your `--include` and `--exclude` options is important. In the example above, the inclusion of /linux_journals_articles takes precedence over the exclusion of everything (`'**'`). Figure 3 shows the result. I could go on with all possible combinations, but the above covers the basics. Work with some test files first to get a handle on the interactions between --include and --exclude. Also, note that each include or exclude pattern needs to be in a separate switch.

Figure 3. Directory listing for backup where only the linux_journal_articles directory is included.

Let me explain a bit more how the include and exclude pattern-matching works. As previously mentioned, order determines precedence, so patterns specified earlier override those specified later. The include/exclude switches work with extended shell globbing patterns. In "standard" shell globbing, the * matches any character except /. Extended globbing adds the ** pattern, which matches any character *including* /. In addition, patterns can be prefixed with `ignorecase:` to match regardless of uppercase/lowercase. For more details, see the above link. As to how the pattern-matching works, I couldn't come up with a better description than the one found in the documentation:

> The --exclude pattern option matches a file if: 1) pattern can be expanded into the file's filename, or 2) the file is inside a directory matched by the option.
>
> Conversely, --include pattern matches a file if: 1) pattern can be expanded into the file's filename, 2) the file is inside a directory matched by the option, or 3) the file is a directory which contains a file matched by the option.

The practical outcome of this is that if you include a file, the directories that form its path also will be included, whereas excluding a file excludes only it and not the directories above it.

For simple cases, the command-line includes and excludes shown previously work well. However, they get unwieldy as the number of include and exclude parameters grow. Fortunately, there is a another way of specifying the parameters, namely file lists. There are three variations: filelist, filelist-stdin and globbing-filelist. The three forms each take an include or exclude prefix—for example, `--include-filelist`. This is somewhat misleading though, because it is possible to specify include and exclude parameters within a file by using + and -, respectively, which override the prefix. For consistency's sake, I will work with the globbing-filelist, as it follows the same rules as the --include and --exclude options on the command line. The other forms do not follow the globbing expansion and include pattern-matching behavior. So, to replicate the last command above using a file list, do:

```
rdiff-backup --include-globbing-filelist rdiff_globbing.txt \
                  software_projects \
                      /var/backups/test_backup
```

The contents of rdiff_globbing.txt are:

```
+ **linux_journal_articles
- **
```

The result is the same as shown in Figure 3.

Creating backups is good, but they are of little use if you can't restore files from them. A restore, at its simplest, is just a backup reversed. In other words, the order of directories on the command line is reversed—the mirror first, the directory to restore to second. There is one important caveat: rdiff-backup, by default, will not restore over an existing file/path. Think of it as sort of a foot/gun safety. You have two options, restore to another path or use the --force switch to override the default behavior.

rdiff-backup gives you two basic methods for restoring a specific version of a file: time-based and number-based.

Time-based restorations are based on the time of the increments. Assuming you have a cron job that does a backup every night at the same time, you would have increments stretching back in time that are timestamped. To see what increments you have, do:

```
rdiff-backup --list-increments \
                  user@host.example.com::/var/backups/lj_article
```

You also can use `-l` rather than `--list-increments`. The target here is the backup directory. The following is the actual output (using a different target) from an EC2 instance that I ran a cron job against to seed with changes for this article:

```
increments.2010-01-16T02:15:05Z.dir    Fri Jan 15 18:15:05 2010
increments.2010-01-17T02:15:06Z.dir    Sat Jan 16 18:15:06 2010
increments.2010-01-18T02:15:05Z.dir    Sun Jan 17 18:15:05 2010
increments.2010-01-19T02:15:06Z.dir    Mon Jan 18 18:15:06 2010
increments.2010-01-20T02:15:06Z.dir    Tue Jan 19 18:15:06 2010
increments.2010-01-21T02:15:05Z.dir    Wed Jan 20 18:15:05 2010
increments.2010-01-22T02:15:05Z.dir    Thu Jan 21 18:15:05 2010
increments.2010-01-23T02:15:05Z.dir    Fri Jan 22 18:15:05 2010
increments.2010-01-24T02:15:06Z.dir    Sat Jan 23 18:15:06 2010
increments.2010-01-25T02:15:06Z.dir    Sun Jan 24 18:15:06 2010
```

With this information in hand, I then can do the following to restore an increment from a particular time:

```
file=/var/backups/lj_article/software_projects
file="$file/linux_journal_articles/rdiff_backup/lj_rdiff_article.txt"

rdiff-backup --restore-as-of 2010-01-20T02:15:06Z \
    user@ec2.example.com::$file \
    test/lj_rdiff_article.txt
```

You also can use `-r` rather than `--restore-as-of`.

I then would have a version of this article as it was when it was backed up at that particular time. There is a more user-friendly variation of this that uses a different view of time. In that method, you specify time as an interval, using the following syntax `integer[modifier]`, where the modifiers are s, m, h, D, W, M and Y, and the time periods they represent are seconds, minutes, hours, days, weeks, months and years, respectively. An example would be `-r 2D12h`, which translates as "restore the file as it was 2 days and 12 hours ago". The question then is, what happens if there is no increment that falls exactly on that time? The answer is, rdiff-backup uses the increment closest to that time that is not later than the specified time. So, for this example, if there were an increment 2 days and 18 hours ago and one 2 days and 11 hours ago, it would use the one from 2 days and 18 hours ago.

The second way to specify a restoration, a number-based restore, is based on the concept of session numbers, where the syntax is `integer[B]`, and 0B is the current mirror version. So to restore from the second-most-recent backup, you would do `-r 2B`.

Although the restore options are powerful, they do require command-line knowledge and a certain amount of familiarity with rdiff-backup. To ease the end-user experience, let's look at

rdiffWeb, a Web-based interface for restoring files backed up using rdiff-backup.

But first, here's a couple quick general usage tips. Each backup session is done in a transaction. This means if the session is interrupted, the next time you do a backup, the previous incomplete backup will be removed. If you want to check for this condition, do:

```
rdiff-backup --check-destination-dir  /var/backups/test_backup
```

If there is an incomplete backup in the directory, it will be rolled back. The transactional nature of a session means that there can be problems if you are backing up remotely over an unstable connection. This especially comes into play when doing an initial backup of a large dataset. One solution to avoid this potential problem is to do a local rdiff-backup backup, and then use rsync to sync the local rdiff mirror to a remote site. rsync allows for interrupted sessions and can be rerun to complete the transfer. Once the remote mirror is complete, change the rdiff-backup mirror directory from the local to the remote. From that point on, you will be transferring only the changes.

Second, each time you do a backup, it creates an increment, and over time, these can build up. To see the space being used by your backup directory, do this:

```
rdiff-backup --list-increment-sizes \
    user@ec2.example.com::/var/backups/lj_article
```

```
        Time                    Size    Cumulative size
-----------------------------------------------------------------------
Tue Jan 26 18:15:05 2010     5.98 MB        5.98 MB    (current mirror)
Mon Jan 25 18:15:07 2010     1.14 KB        5.98 MB
Sun Jan 24 18:15:06 2010     1.54 KB        5.99 MB
Sat Jan 23 18:15:06 2010     0 bytes        5.99 MB
Fri Jan 22 18:15:05 2010     0 bytes        5.99 MB
Thu Jan 21 18:15:05 2010     0 bytes        5.99 MB
Wed Jan 20 18:15:05 2010     0 bytes        5.99 MB
Tue Jan 19 18:15:06 2010     25.8 KB        6.01 MB
Mon Jan 18 18:15:06 2010     1.22 KB        6.01 MB
Sun Jan 17 18:15:05 2010     32.7 KB        6.04 MB
Sat Jan 16 18:15:06 2010     0 bytes        6.04 MB
Fri Jan 15 18:15:05 2010     0 bytes        6.04 MB
Thu Jan 14 18:15:05 2010     0 bytes        6.04 MB
Wed Jan 13 18:15:05 2010     0 bytes        6.04 MB
Tue Jan 12 18:15:05 2010     0 bytes        6.04 MB
Mon Jan 11 18:15:06 2010     0 bytes        6.04 MB
Sun Jan 10 18:15:06 2010     0 bytes        6.04 MB
Sat Jan  9 18:15:07 2010     0 bytes        6.04 MB
Fri Jan  8 18:15:08 2010     0 bytes        6.04 MB
Thu Jan  7 18:15:06 2010     0 bytes        6.04 MB
Wed Jan  6 18:15:06 2010     31.5 KB        6.07 MB
Wed Jan  6 07:05:52 2010    194 bytes       6.07 MB
```

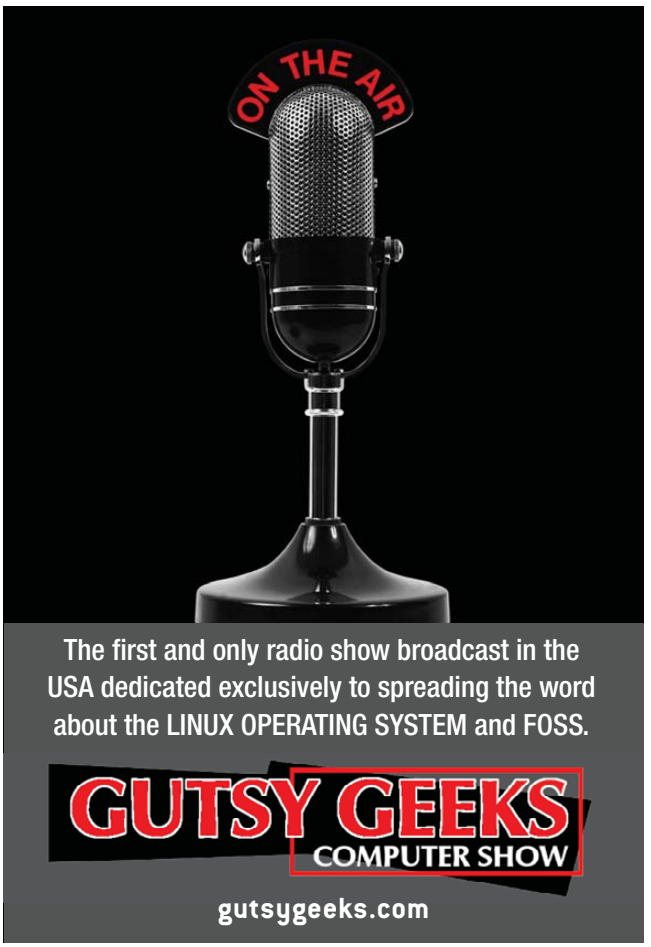
This is a good way of seeing where the storage is being used and the total for your backup directory. Unless you have unlimited backup disk space, at some point, you will want to start pruning your backup directory. To prune increments, use `--remove-older-than 'time'`, where time is specified the same as is used when restoring. For my own personal data, I have a cron job that runs each night with:

```
rdiff-backup --remove-older-than 7D 'destination_dir'
```

This keeps a rolling seven-day version history of my backups.

Now as promised: rdiffWeb, available at **www.rdiffweb.org**. rdiffWeb, like rdiff-backup, is written in Python. It uses CherryPy to serve up its pages. I am using the 0.6.3 version, which is the testing release at the time of this writing. The primary reason for this is that 0.6.3 uses SQLite as its datastore instead of MySQL, as the stable version 0.5.1 does. SQLite comes bundled with Python 2.5+, so it is one less thing to set up. Be sure to follow the Installation link on the page listed above. The installation is fairly straightforward, but there are some manual changes you need to do for the program to function properly. Note that it's possible to set up rdiffWeb for https, which is how it is used for this article.

When you first access rdiffWeb, you will be prompted to set up an administrator account. Once that is done, you then can set up users and work with User Preferences (Figure 4). Of note is the Directory Restore Format option. This comes into play if you are going to restore an entire directory. rdiffWeb either will create a zip file or gzipped tarball, depending on the setting, which is good for dealing with Windows or UNIX users. Also worth noting is the Update Backup Locations button. This is a way of forcing rdiffWeb to find any new backup directories you have added. It will find them on its own, but generally not immediately.



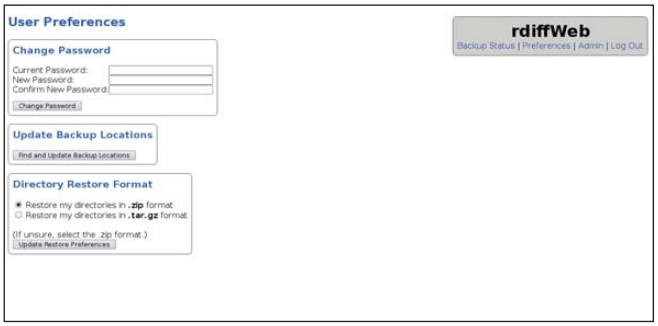


Figure 4. User Preferences Page in rdiffWeb Setup

The point is to find files. When you log in, a page will be presented with the backup directories for which you have permissions. You then can traverse the directory tree to get to the file or directory you want. Figure 5 shows four versions of this article available. Clicking on a timestamp opens the browser download dialog allowing you to download the file to your choice of location. If you want to download a directory, click on the Restore Folder link. You then will be taken to a page that lists the various available timestamped versions. Select the appropriate version, and the directory and its contents will be bundled up according to the Directory Restore Format specified on your User Preferences page.

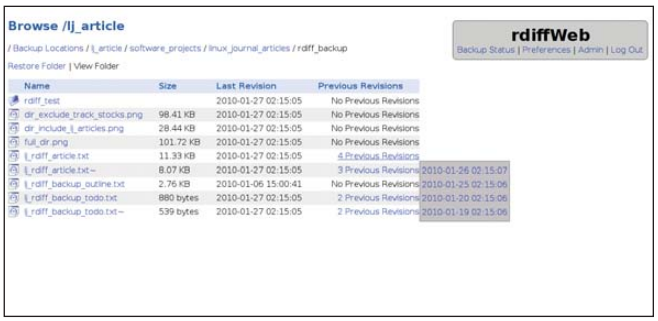


Figure 5. Selecting a File Version in rdiffWeb

rdiff-backup is a useful tool to use when you want to have versioning of backups without keeping around full copies of the versions. It is relatively easy to set up. The command-line nature of the program is both a plus and a minus. The plus is that it lends itself to scripting. The minus is that not all users are command-line-savvy, hence the section on rdiffWeb. My method of running it is to have cron jobs that do the backups and then use rdiffWeb for restoring.

Finally, here are some general observations on the incremental nature of the backups. The amount of space saving is dependent on usage. rdiff-backup saves all information, so if you do a lot of deletes, the backup directory will fill up with the deleted files unless you prune regularly. As described earlier, the block nature of the file comparison is not efficient for small changes across multiple blocks. I bring this up as a prompt to have you monitor the disk space for your destination directory for a period of time until you have a sense of how your usage pattern affects it. On the whole though, I have found rdiff-backup provides a good compromise between backup size and backup versatility.■

Adrian Klaver would like to think computers don't really rule his life, and to that end, he occasionally can be found staring at waves lapping up on the beach. As far as geek credentials, he is one of the organizers for LinuxFest Northwest in Bellingham, Washington.

Advertiser Index

CHECK OUT OUR 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!

Advertiser	Page #	Advertiser	Page #
1&1 Internet, Inc. www.oneandone.com	C2, 1, 3	Microway, Inc. www.microway.com	C4, 7
Aberdeen, LLC www.aberdeeninc.com	19	Ontario Linux Fest oglf.ca	63
ApacheCon www.na.apachecon.com	59	Polywell Computers, Inc. www.polywell.com	78, 79
Archie McPhee www.mcphee.com	78	Saint Arnold Brewing Company www.saintarnold.com	78
ASA Computers, Inc. www.asacomputers.com	55	ServerBeach www.serverbeach.com	27
Cari.net www.cari.net	69	Servers Direct www.serversdirect.com	9
Digi-Key Corporation www.digi-key.com	78	Silicon Mechanics www.siliconmechanics.com	25, 35
Emac, Inc. www.emacinc.com	61	SuperComputing SC10 sc10.supercomputing.org	47
Gecad Technologies/Axigen www.axigen.com	79	SuperMicro, Inc. (MBX) www.supermicro.com	13
Genstor Systems, Inc. www.genstor.com	51	Technologic Systems www.embeddedx86.com	39
Gutsy Geeks www.gutsygeeks.com	64	Trusted Computer Solutions www.trustedcs.com/SecurityBlanket	5, 79
iXsystems, Inc. www.ixsystems.com	C3	USENIX LISA www.usenix.org/lisa10/lj	37
Logic Supply, Inc. www.logicsupply.com	31	Utilikilts www.utilikilts.com	79
Lullabot www.lullabot.com	53	Zend/PHP Conference & Expo www.zendcon.com	45

ATTENTION ADVERTISERS

January 2011 Issue #201 Deadlines
Space Close: October 25; **Material Close:** November 2

Theme: System Administration

BONUS DISTRIBUTIONS: Do It with Drupal

Contact **Joseph Krack, +1-713-344-1956 ext. 118, joseph@linuxjournal.com**

Introduction to the MeeGo Software Platform

Distros come and go, and sometimes they combine with others to form new distros. Take Intel's Moblin and combine it with Nokia's Maemo, and you get MeeGo.

IBRAHIM HADDAD

On February 15, 2010, the world's largest chip manufacturer, Intel, and the world's largest mobile handset manufacturer, Nokia, announced joining their existing open-source projects (Moblin and Maemo, respectively) to form a new project called MeeGo, hosted at the Linux Foundation. This article provides an introduction to the MeeGo Project, a brief overview of the MeeGo architecture, the benefits the MeeGo platform offers to various players in the ecosystem and discusses the role of the Linux Foundation as the project's host.

MeeGo is a Linux-based platform that is capable of running on multiple computing devices, including handsets, Netbooks, tablets, connected TVs and in-vehicle infotainment systems. The primary goal of the Maemo and Moblin Projects' merger was to unify the Moblin and Maemo communities' efforts and enable a next-generation open-source Linux platform suited for a variety of client devices. Most important, MeeGo will be doing this while maintaining freedom for innovation, continuing the tradition of community involvement (inherited from Maemo and Moblin), accelerating time to market for a new set of applications, services and user experiences.

> MeeGo is a full open-source project hosted by the Linux Foundation and governed according to best practices of open-source development. As with other true open-source projects, technical decisions are made based on technical merit of the code contributions being made.—Ari Jaaksi, Vice President of MeeGo Devices, Nokia.

With the merger, the MeeGo Project has the opportunity to expand market opportunities significantly on a wide range of devices. It also will provide a rich cross-platform development environment, so applications can span multiple platforms. Additionally, it will unify developers, providing a wealth of applications and services. Such opportunities were out of reach for Maemo and Moblin individually. MeeGo will support multiple chip architectures (ARM and x86). Furthermore, with hundreds of developers working in the open on upstream projects first, from which MeeGo will be based, other mobile Linux platforms will benefit from MeeGo's contributions.

Maemo Background

The Maemo Project, initially created by Nokia (**www.maemo.org**), provided a Linux-based software stack that runs on mobile devices. The Maemo platform is built in large part with open-source components, and its SDK provides an open development environment for applications on top of the Maemo platform. A series of Nokia Internet tablets with touchscreens have been built with the Maemo platform. The latest Maemo device is the Nokia N900 powered by Maemo 5, which introduced a completely redesigned finger-touch UI, cellular phone feature and live multicasting on the Maemo dashboard.

MeeGo Architecture

MeeGo provides a full open-source software stack from the core operating system up to the user interface libraries and tools. Furthermore, it offers user experience (UX) reference implementations and allows proprietary add-ons to be added by vendors to support hardware, services or customized user experiences. Figure 1 illustrates the MeeGo architecture, which is divided into three layers:

- The MeeGo OS Base layer consists of the hardware adaptation software required to adapt MeeGo to support various hardware architectures and the Linux kernel and core services.
- The MeeGo OS Middleware layer provides a hardware and usage model-independent API for building both native applications and Web runtime applications.
- The MeeGo User Experience (UX) layer provides reference user experiences for multiple platform segments. The first UX reference implementation was released on May 25, 2010, for the Netbook UX. Other UX reference implementations will follow for additional supported device types.

A detailed discussion of the MeeGo software platform is available at **meego.com/developers/MeeGo-architecture**.

As mentioned earlier, the Netbook UX

Moblin Background

The Moblin Project, short for Mobile Linux, is Intel's open-source initiative (**www.moblin.org**) created to develop software for smartphones, Netbooks, mobile Internet devices (MIDs), in-vehicle infotainment (IVI) systems and other mobile devices. It is an optimized Linux-based platform for small computing devices. It runs on the Intel Atom, an inexpensive chip with low power requirements. A device running Moblin boots up quickly and can be on-line within a few seconds.

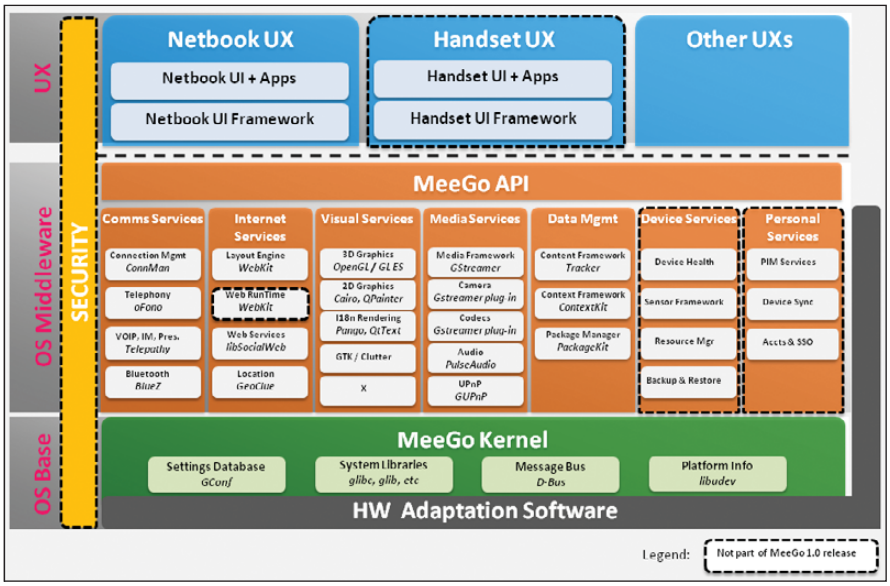


Figure 1. MeeGo Component-Level Architecture Diagram

was the first reference implementation of a UX to become available for MeeGo. It delivers a wealth of Internet, computing and communication experiences with rich graphics, multitasking and multimedia capabilities, and it's highly optimized for power and performance. You can download the MeeGo Netbook images from **meego.com**, and run MeeGo on your Netbook. Figure 2 shows a screenshot of the Netbook UX featuring the MeeGo MyZone (the home screen).

Benefits of the MeeGo Project

The MeeGo open-source project is unique in that it offers benefits to everyone in the ecosystem, starting from the developer all the way up to the operator and the industry as a whole. MeeGo allows participants to get involved and contribute to an industry-wide evolution toward richer devices to address opportunities rapidly and to focus on differentiation in their target markets.

Benefits to Open-Source Developers

As mentioned previously, the MeeGo Project is a true open-source project hosted by the Linux Foundation and governed by best practices of open-source development. From MeeGo.com, as an open-source developer, you have access to tools, mailing lists and a discussion forum. You also have accessibility to technical meetings and multiple options for making your voice heard in technical and nontechnical MeeGo-related topics. Furthermore, all source code contributions needed for MeeGo will be submitted to the upstream open-source projects from which MeeGo will be built (Figure 3).

LINUX™ JOURNAL
Linux News and Headlines Delivered To You
Linux Journal topical RSS feeds available
linuxjournal.com/rss_feeds

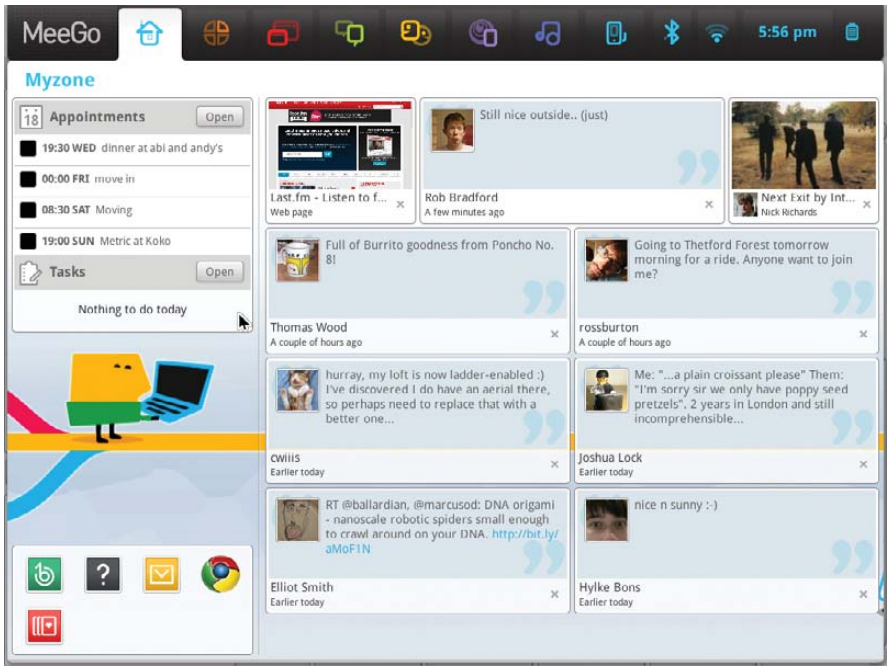


Figure 2. Screenshot of the MeeGo Netbook UX, Featuring the MyZone Home Screen (Source: MeeGo.com)

Benefits to Application Developers

As an application developer, MeeGo significantly expands your market opportunities, as it is the only open-source software platform that supports deployments across many computing device types. MeeGo offers Qt and Web runtime for application development and cross-platform environments, so application developers can write their applications once and deploy easily on many types of MeeGo devices or even on other platforms supporting the same development environment. Furthermore, MeeGo will offer a complete set of tools for developers to create a variety of innovative applications easily and rapidly (see **meego.com/developers/getting-started**). The major advantage from this approach (Figure 4) is having a single set of APIs across client devices. In addition, in this context, "multiple devices" means much more than just multiple types of handsets, for instance. MeeGo device types include media phones, handhelds, IVI systems, connected TVs and Netbooks.

In addition, MeeGo application developers will have the opportunity to make their applications available from multiple application stores, such as Nokia's Ovi Store (**https://store.ovi.com**) and Intel's AppUp Center (**www.intel.com/consumer/products/appup.htm**). Also, there is the opportunity to host the applications on other app stores for specific carriers offering MeeGo devices as part of their device portfolios. These MeeGo capabilities and cross-device and cross-platform development are major differentiators and offer huge benefits to application developers.

Benefits to Device Manufacturers

MeeGo helps accelerate time to market using an off-the-shelf, open-source and optimized software stack targeted for the specific hardware architecture the device manufacturer is supporting. From a device manufacturer perspective, MeeGo lowers complexities involved in targeting multiple device segments by allowing the use of the same software platform for different client devices. In addition, as an open-source project, MeeGo enables device manufacturers to participate in the evolution of the software platform and build their own assets for it through the open development model.

Benefits to Operators

MeeGo enables differentiation through user interface customization. Although many devices can be running the same base software platform, they all can have different user experiences. Furthermore, it

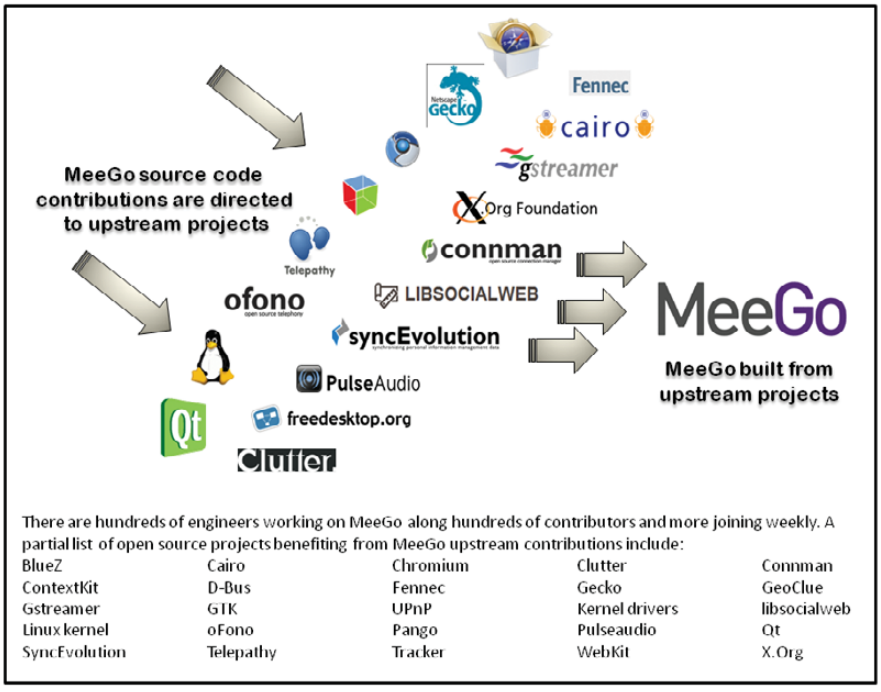


Figure 3. MeeGo and Upstream Projects

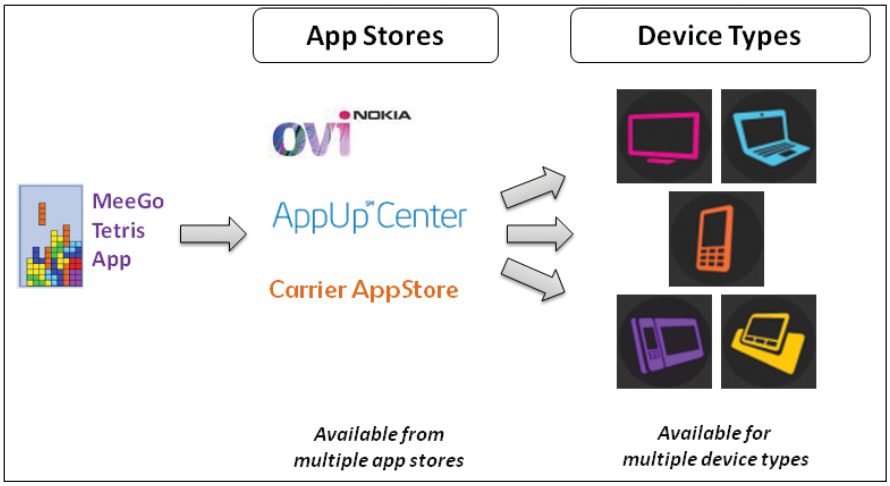


Figure 4. MeeGo Apps Available from Multiple App Stores for a Wide Range of Device Types

provides a single platform for a multitude of devices, minimizing the efforts needed by the operators/carriers in training their teams, which allows their subscribers to be familiar with the experience common to many device types.

Benefits to the Linux Platform

MeeGo is a vehicle for fostering mobile innovations through an open collaborative environment, promoting the exchange of ideas and source code, peer review, unifying development from across multiple device categories and driving contributions and technical work upstream to various open-source projects.

In addition, MeeGo contributes to Linux as a technical platform, as it combines mobile development resources that recently were split in the Maemo and Moblin Projects into one well-supported, well-designed project that addresses cross-platform, cross-device and cross-architecture development. Dozens of traditional Linux mobile and desktop efforts use many of the components used by MeeGo. They all benefit from the increased engineering efforts on those components. This is the power of the open-source development model.

The Linux Foundation and MeeGo

The Linux Foundation (**www.linuxfoundation.org**) hosts the MeeGo Project as an open-source project, provides a vendor-neutral collaboration environment and encourages community contributions in line with the best practices of the open-source development model.

The Linux Foundation and MeeGo meet on various key points:

- Accelerating the adoption of Linux.
- Promoting collaboration between industry players and the Open Source community.
- Unifying divergent efforts toward the benefits of a strong Linux platform.
- Promoting a truly open Linux platform and improving Linux as a technical platform.
- Encouraging companies to drive their contributions and technical work upstream.

As owner of the MeeGo trademark, the Linux Foundation also is driving the

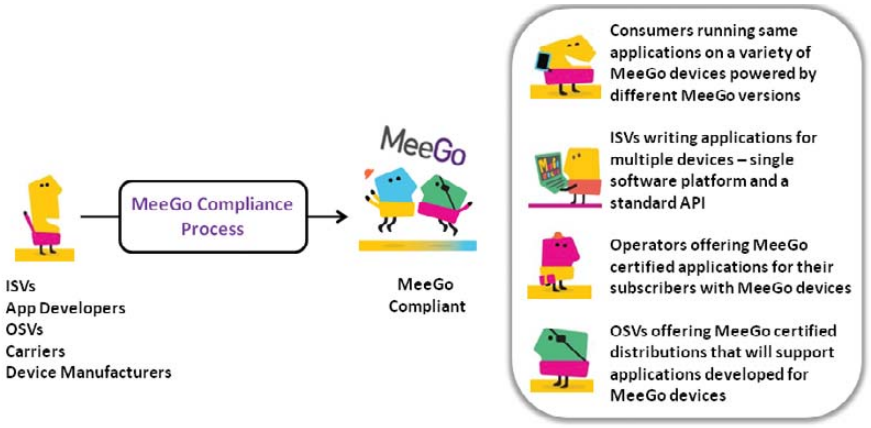


Figure 5. Benefits of the MeeGo Compliance Program to Various Ecosystem Players

15 MeeGo Facts

1. Full open-source project.
2. Hosted under the auspices of the Linux Foundation.
3. Aligned closely with upstream projects—MeeGo requires that submitted patches also be submitted to the appropriate upstream project and be on a path for acceptance.
4. Offers a complete software stack.
5. Provides reference UX implementations.
6. Governed according to best practices of open-source development.
7. Offers equal opportunities for all players and enables them to participate in the evolution of the software platform and to build their own assets on MeeGo.
8. Lowers complexity for targeting multiple device segments.
9. Offers differentiation abilities through UX customization.
10. Provides a rich cross-platform development environment and tools.
11. Offers a compliance program to certify software stacks and application portability.
12. Supports multiple hardware architectures.
13. Supports multiple app stores.
14. Has no contributor agreements to sign.
15. Has more than 1,000 committed professional developers and hundreds of open-source developers.

creation of a MeeGo compliance program that will allow ISVs and OSVs to go through the compliance program and have their applications, distributions, devices and so on certified as MeeGo-compliant. Figure 5 illustrates the benefits of the compliance program to various players in the MeeGo ecosystem. MeeGo will enable applications developed with the MeeGo API to run on all devices running MeeGo-compliant OSes with segment-specific adaptations.

The Linux Foundation contributes to the MeeGo Project through its coordination efforts, overseeing MeeGo events; hosting a number of MeeGo-related technologies, services and collaboration tools; in addition to various marketing, legal, PR and other support activities. Furthermore, the Linux Foundation employs the maintainers of the cross-toolchain used by MeeGo and contributes the optimizations for multi-architecture support in the build service.

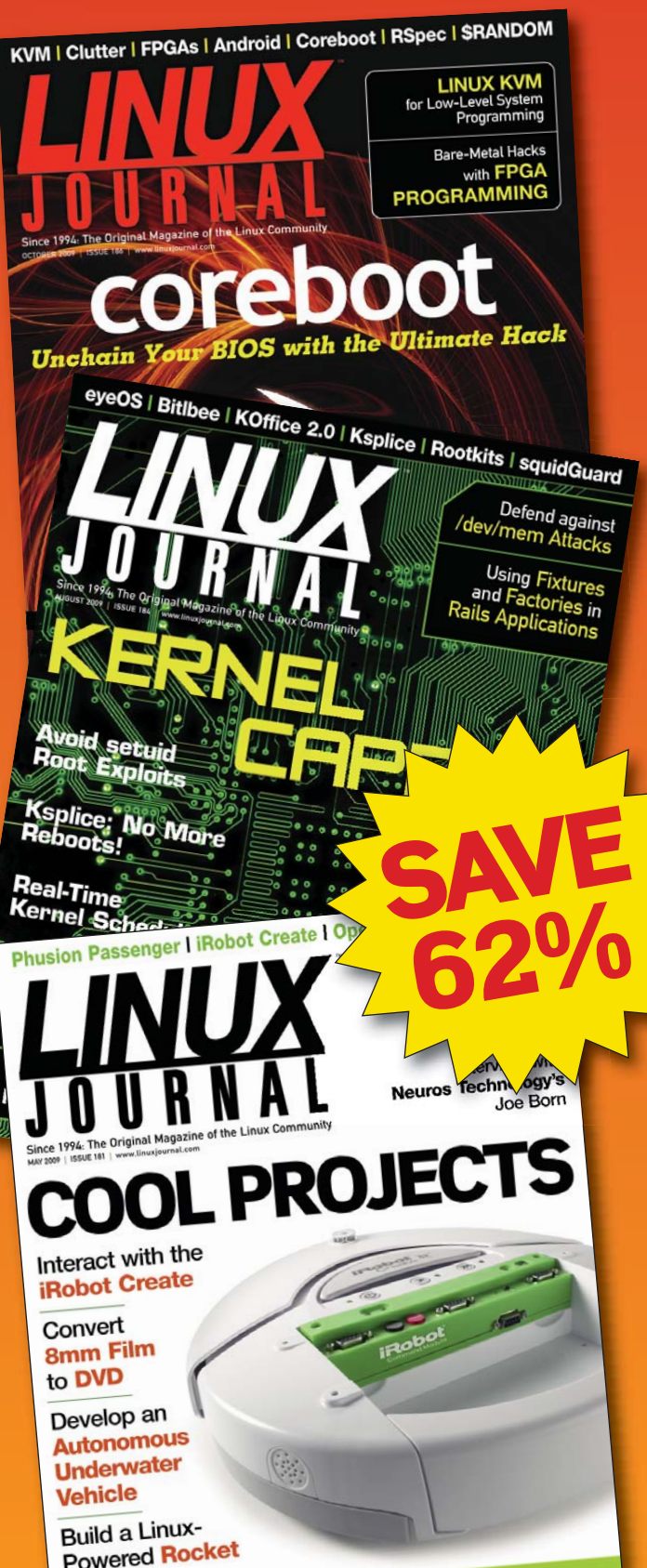
Conclusion

MeeGo is a fully open-source software platform, hosted under the auspices of the Linux Foundation, and governed according to best practices of open-source development. Open mailing lists, discussion forums and contribution are open to all. It offers the best of Moblin and the best of Maemo to create a platform for multiple hardware architectures covering the broadest range of device segments.

MeeGo offers several opportunities to participate and help shape the future of the next-generation Linux platform. It will accelerate the adoption of Linux on mobile devices, Netbooks, pocket mobile devices, in-vehicle entertainment centers, connected TVs and mobile phones by implementing a truly open Linux platform across multiple architectures for next-generation computing devices. It is an open collaborative project between the project founders (Intel and Nokia), the community and various commercial and noncommercial partners bringing thousands of developers to work entirely out in the open, driving their contributions and technical work upstream making Linux the platform of choice for mobile computing devices. Visit MeeGo.com and be part of it!■

Ibrahim Haddad is Director of Technical Alliances at the Linux Foundation and a Contributing Editor for *Linux Journal*.

If You Use Linux, You Should Be Reading LINUX JOURNAL™



SAVE 62%

- In-depth information providing a full 360-degree look at featured topics relating to Linux
- Tools, tips and tricks you will use today as well as relevant information for the future
- Advice and inspiration for getting the most out of your Linux system
- Instructional how-tos will save you time and money

Get *Linux Journal* delivered to your door monthly for 1 year for only $29.50! Plus, you will receive a free gift with your subscription.

SUBSCRIBE NOW AT:

WWW.LINUXJOURNAL.COM/SUBSCRIBE

Offer valid in US only. Newsstand price per issue is $5.99 USD; Canada/Mexico annual price is $39.50 USD; International annual price is $69.50. Free gift valued at $5.99. Prepaid in US funds. First issue will arrive in 4-6 weeks. Sign up for, renew, or manage your subscription on-line, www.linuxjournal.com/subscribe.

Virtualization the Linux/OSS Way

VirtualBox often is called a "desktop" virtualization solution, but it's just as capable of being a server solution. And contrary to what you may believe, no GUI is required; you can manage it all from the command line. GREG BLEDSOE

I'm just a command-line kind of guy. I prefer not to have a GUI running on my servers taking up resources and potentially exposing security issues, and fitting with that mindset of favoring simplicity and economy, I'm also fairly frugal. I don't like to pay more than necessary. I hear peers in the IT field discuss the complexity of their environments and how much they pay for their virtualization solutions, shaking their heads in mock sympathy for each other, while bragging rights go to whomever has the biggest, most complex, most expensive environment. Meanwhile, I listen politely until I eventually chime in, "I pay nothing for my virtualization solution and manage it from the command line. Oh, and did I mention, our startup just started turning a profit." Then come the blank stares.

Being a firm believer in the benefits of free, open-source software, I prefer software that is either pure FOSS or at least has an open-source version available, even if some premium features are available only in paid versions. I believe that over time, this produces superior software, and I have a safety net so that my fate is not entirely in the hands of one commercial entity. For all of these reasons, I use VirtualBox as the primary virtualization software in my organization. That choice was made before Sun—who bought the creators of VirtualBox, Innotek—was in turn bought by Oracle, who also just bought another virtualization provider and had put years of considerable effort into developing a Xen-based solution, throwing the future of VirtualBox into limbo.

To get all the features in the latest release of VirtualBox, you'd have to shell out some major dough for VMware's top-of-the-line product.

Many of the details on the direction Oracle/Sun will take with virtualization are still up in the air, but things look promising. Xen, while being elegant and providing performance with which other methods of virtualization can't compete, has proven to be a bit of a disappointment in terms of being able to support packaged solutions and keeping up with the features of other solutions. Since the purchase of Sun, there have been a number of 3.0.x maintenance releases of VirtualBox, the beta release (3.1) was formalized into 3.2.0, and there have been two maintenance releases since, getting us to 3.2.4 at the time of this writing. I have it from one of the developers working on the project that at 40,000 downloads a day, VirtualBox is still one of the most popular pieces of software in the Oracle/Sun portfolio. See Resources for a link to Oracle's VirtualBox Blog, where Oracle has officially announced VirtualBox as an Oracle VM product. Indeed, it wouldn't make sense to throw this valuable commodity in the trash, as VirtualBox is much younger yet compares favorably in many respects to the 800-pound gorilla of virtualization, VMware. VirtualBox is performant, reliable and simple to manage. To get all the features in the latest release of VirtualBox, you'd have to shell out some major dough for VMware's top-of-the-line product.

Our two primary VirtualBox hosts have two four-core CPUs per host, 48 gigs of RAM apiece, and ten SATA2 10K RPM drives each, in RAID 0+1, running the x86_64 kernel. Based on initial testing, I expected that we would be able to run about 10–15 virtual hosts each on this hardware, but to my surprise, we currently are at 20 hosts apiece and counting, with headroom. I give each virtual machine one CPU, which makes it stick to one CPU core at a time, and as little memory as possible, increasing those if performance demands it. Using this methodology, we've achieved an environment where if you didn't know the machines were virtual, you couldn't tell these machines weren't on dedicated hardware. The load average on one host rarely tops 1.0 for a 15-minute average and rarely tops 2.0 on the other, so we still have some headroom. I'm now thinking we may be able to run 30 or more machines per host with enough RAM, and possibly more.

I run VirtualBox on Ubuntu Server, which we standardized on for legacy reasons. We currently are on the 3.0.x branch of VirtualBox and are testing upgrades to 3.2. Unfortunately, there still isn't a "non-GUI" package for VirtualBox on Ubuntu, so installing it without also installing Xorg and Qt packages involves the use of the `--force-depends` option to dpkg (or the equivalent on your system). The OSE (open-source edition) version is available via apt-get in the standard repositories, but I recommend downloading the latest version directly. I'm showing you the install I did of the latest stable version at the time I last upgraded the production systems, but to get all the latest features, like teleportation of virtual machines to other hosts, you'll need to go with the latest 3.2.x version:

```
loc=http://download.virtualbox.org/virtualbox/3.0.14/
wget $loc/virtualbox-3.0_3.0.14-58977_Ubuntu_karmic_amd64.deb
# or appropriate package for your distro/architecture
dpkg -i --force-depends \
     virtualbox-3.0_3.0.14-58977_Ubuntu_karmic_amd64.deb
```

dpkg will complain about missing dependencies. You can ignore most of them, but you will need to satisfy the non-GUI dependencies to have full functionality. Subsequent to this, you will find that when you need to install or upgrade packages via the apt utility, it will complain about broken dependencies and refuse to do anything until you resolve the problem. I get around this by taking down all the virtual machines on a host, bringing up the essential ones on another host, uninstalling the virtualbox package, performing my upgrades or installs, and then re-installing. It's an extra step and takes a few minutes, but on your production virtualization hosts, this probably isn't something you will be doing terribly often, as it should have a minimum of required packages to start with:

```
dpgk -r VirtualBox-3.0.14
apt-get update
apt-get upgrade
dpkg -i --force-depends \
     virtualbox-3.0_3.0.14-58977_Ubuntu_karmic_amd64.deb
```

Once things are installed, everything is exposed via the command line. In fact, the GUI is only a subset of what is available via the CLI. Currently, configuring port forwarding and the use of the built-in iSCSI initiator are possible only via the CLI (not via the GUI). Try typing this in and pressing Enter for some undocumented goodness that has saved me many hours and headaches:

```
VBoxManage internalcommands
```

The usage information available by typing partial commands is exhaustive, and the comprehensive nature of what is available allows for many custom and time-saving scripts. I've scripted all the repetitive things I do, from creating new VMs to bouncing

Listing 1. Script to Create Multiple VMs

```
#! /bin/bash
# A quick and dirty script to create multiple virtual machines,
# give them unique hostnames and IP addresses, and culminate in
# bringing them on-line.

# name of the directory where we'll mount our vdi's
dir=temp
rootdir=`pwd`/$dir

# the basename for the vms
basename=vbox-vm-

# the file that contains the basic disk image
basevdi=base.vdi

# how many images are we making
number=2

# what subnet will these guests be going on
IPnetwork='10.7.7.'
gateway='10.7.7.1'

# the start of the address range we will use
baseIP=10

# amount of memory these guests will get in Mbytes
memory=512

# base VRDP port
baseRDP=16001

counter=1
while [ $counter -le $number ]
do
  echo $basename$counter $basename$counter.vdi \
       $IPnetwork$baseIP $memory
  VBoxManage clonehd `pwd`/base.vdi \
                    `pwd`/$basename$counter.vdi --variant Fixed
  sudo mount_vdi/mount_vdi.sh $basename$counter.vdi $rootdir 1
  sudo sed -i "s/basicsys/$basename$counter/g" $rootdir/etc/hosts
  sudo sed -i "s/basicsys/$basename$counter/g" $rootdir/etc/hostname
  sudo sed -i "s/1.1.1.2/$gateway/g" $rootdir/etc/network/interfaces
  sudo sed -i "s/1.1.1.1/$IPnetwork$baseIP/g" \
           $rootdir/etc/network/interfaces
  sudo rm $rootdir/etc/udev/rules.d/70-persistent-net.rules
  sudo touch $rootdir/etc/udev/rules.d/70-persistent-net.rules
  sudo umount $rootdir
  sudo losetup -d /dev/loop1
  sudo losetup -d /dev/loop0
  VBoxManage createvm --name $basename$counter --register
  VBoxManage modifyvm $basename$counter --pae on --hwvirtex on
  VBoxManage modifyvm $basename$counter --memory $memory --acpi on
  VBoxManage modifyvm $basename$counter \
                     --hda `pwd`/$basename$counter.vdi
  VBoxManage modifyvm $basename$counter \
             --nic1 bridged --nictype1 82540EM --bridgeadapter1 eth0
  VBoxHeadless --startvm $basename$counter -p $baseRDP &
  sleep 5
  baseRDP=$((baseRDP + 1))
  baseIP=$((baseIP + 1))
  counter=$((counter + 1))
done
```

ones that become troublesome after a month or so of uptime. Not only can you shell script, but there also is a Python interface to VirtualBox, and the example script, vboxshell.py, ships with the standard distribution.

A few notes on efficiency and performance—you'll do well to set up a "template" virtual machine installation of your various operating systems that has your environment's configurations for authentication, logging, networking and any other commonalities necessary. You'd also need to throw in your performance enhancements, like for instance, adding `divider=10` to the GRUB kernel configuration, resulting in a line like this:

```
kernel /vmlinuz-2.6.18-164.el5 ro \
       root=/dev/VolGroup01/LogVol00 rhgb quiet divider=10
```

This will require some experimentation in your environment, but most systems are set to a 1,000Hz clock cycle. Even on a host with idle guest systems, the number of context switches that occur simply to check for interrupts can result in high load on the host. This boot-time parameter will divide the clock frequency by ten, reducing the number of context switches by a factor of ten as well, and reducing host load greatly. This might not be suitable for all workload types, but running it on the guests for which it does not produce unacceptable performance will speed up the system overall and most of your guests.

The usage information available by typing partial commands is exhaustive, and the comprehensive nature of what is available allows for many custom and time-saving scripts.

Consider the example script in Listing 1. Like most of the scripts I write, it's quick and dirty, but if you follow this example, you'll have a basic infrastructure in place that you can use to provision and manipulate an almost unlimited number of virtual servers quickly. It takes no variables or input on the command line, supplying the information internally, but it easily could be modified to allow passing parameters via command-line switches. Let's say you need to bring up many virtual machines based on your base disk image. You will give the virtual machines sequential IPs and hostnames. The example script has a few prerequisites you'll have to satisfy. First, the disk image you are starting with must be a fixed size, not dynamically allocated. If you have a dynamic .vdi you want to use, first convert it using clonehd:

```
VBoxManage clonehd dynfile.vdi statfile.vdi --variant Fixed
```

Next, you'll need the mount_vdi script (see Resources), which is quite handy in itself, as it mounts the .vdi file as if it were an .iso or raw disk image. And, you'll need to be able to execute it via sudo to create and mount loopback devices. I have edited the mount_vdi.sh script to comment out the lines telling you to type end to unmount and exit the script, and the last few lines of the script that actually do so, moving those functions into the top-level script. You'll need to do likewise for the script in Listing 1 to work. Once you've tested the mount_vdi.sh script, change the path to it in the script in Listing 1 to the appropriate one on your host system.

Assumptions made for the purposes of this particular script include the following: you are running a fairly recent Ubuntu as host and guest with VirtualBox 3.0.x (what I run in production); the .vdis are in the current working directory where the script is located; there are no loop devices (/dev/loopx) on the system already; the root partition is the first one on the virtual disk; the hostname on the base vdi image is basicsys.example.com; it has one Ethernet interface, and the IP address is 1.1.1.1 with a default gateway of 1.1.1.2. The script will be less painful, especially during testing, if you set up your virtualbox user to have password-less sudo, at least temporarily. I've tested the script now on several systems and with slight variations in .vdi age and format, so I'm reasonably confident it will work for a wide variety of environments. Be warned; I have found that VirtualBox 3.2.x, which I am now testing, requires a few changes. Should you run into trouble and solve it, drop me an e-mail at the address in my bio, and let me know, so I can improve the script for future generations.

If you get this script working, you are well on your way to having the infrastructure in place to support a manageable, flexible, cost-effective, robust virtualization environment. Personally, I'm looking forward to getting 3.2.x in place and being able to teleport running machines between hosts to manage workloads in real time—from the command line, of course. Stay tuned, my next article will deal with the back-end shared storage (based on open protocols and free, open-source software, while being redundant and performant). I intend to connect my virtualization hosts to support being able to:

```
VBoxManage controlvm vbox-vm-3 \
            teleport --host vbox-host-2 --port 17001
```

Greg Bledsoe is the Manager of Technical Operations for a standout VoIP startup, Aptela (www.aptela.com), an author, husband, father to six children, wine enthusiast, amateur philosopher and general know-it-all who welcomes your comments and criticism at lj@bledsoehome.net.

Resources

Oracle's Virtualization Blog: **blogs.oracle.com/virtualization**

mount_vdi: **www.mat.uniroma1.it/~caminati/mount_vdi.html**

Bill Childers' "Virtualization Shootout: VMware Server vs. VirtualBox vs. KVM", *LJ*, November 2009: **www.linuxjournal.com/article/10528**

The 1994–2009 Archive CD, back issues, and more!

www.LinuxJournalStore.com

KYLE RANKIN

BILL CHILDERS

Sane Defaults vs. Configurability

How much configurability is too much?

KYLE: Hey Bill, I thought you might be interested to know that in honor of the Command-Line issue my Hack and / column talks about how to set up mutt for the very first time.

BILL: Mutt, eh? That's cool...if you like a bunch of things to configure and stuff. I still don't get that. Mutt may be powerful, but there's this huge barrier to getting it running—so much so, you need an entire article on it. No one writes an article on filling out the new account wizard in Thunderbird.

KYLE: I guess that's true, but without rehashing our mutt vs. Thunderbird debate, you don't need an article talking about Thunderbird options, because there are just so few options you *can* configure. Although it's true that mutt has a lot of options, that's a *feature* for me, not a bug.

BILL: Exactly. I guess it comes down to what our Point/Counterpoint is this month: sane defaults vs. configurability.

KYLE: So at SCaLE this past year, I actually met the creator of mutt (I know, right? I was speechless). After the fanboy in me went away and I was able to talk again, I found out that his .muttrc file actually is pretty small. You know why? Because he wrote mutt's default for what he wanted. The fact that there are so many complicated .muttrc files out there tells me that what were sane defaults for him, obviously weren't sane for everyone else.

BILL: I remember that. You were nearly apoplectic. It was amazing to see the fanboy in action.

KYLE: So, I'm not even sure there is such a thing as sane defaults. I think no matter what, you are going to pick a default someone hates. That's why, to me, being able to configure as much as possible is a huge benefit.

BILL: I don't know, the GNOME folks seem to do a pretty good job of finding that middle ground. I don't mind configurability, but I think you should have the defaults ready to go. A piece of software shouldn't be a blank slate without you digging in a configuration file; it should come up and mostly work right out of the box.

KYLE: I think that middle ground shifted recently when Ubuntu decided the sane default button placement was on the left. As bad as the flame wars over that were, imagine if that weren't a configurable option!

BILL: I didn't mind the button placement on the left, coincidentally. I wonder why that is....But yeah, things like that *should* be configurable. Sane defaults mean exactly that, things that make sense—or perhaps failing sane defaults, maybe a startup wizard of some kind.

KYLE: Okay, I do agree with you to a point there. An application should try to start up in as usable a state as it can. My problem is less what defaults an application chooses and more whether developers think their defaults are the gold standard and don't let you change them. Honestly, I don't care where the window buttons are either; I usually have keybindings do those actions.

BILL: It all depends on the application, I imagine. I can see why there are good reasons for not changing things on an Android phone, for instance.

KYLE: I don't know though, when you look at all the custom cases, plastic covers, stickers and so on that people put on their computers and phones, it tells *me* that people like to customize their things. While trying to please everyone, some applications choose defaults that please no one, and what's worse, they don't allow you to change them. I guess I'm just weird enough that what does end up pleasing the majority of users ends up not working for me, so it's very important that if it doesn't work for me, I can change it.

BILL: I agree, you are weird. But I don't advocate just configurability in lieu of sane defaults. Sane defaults by definition mean they should please the majority of users. You can't just offer all these configurable knobs and dials and then not have them set to something that makes sense out of the box.

KYLE: Well, these days, "sane defaults" often mean they are aimed at some non-existent "Joe User" beginner that the developer made up, so what we end

up with are lowest-common-denominator design decisions and defaults that usually leave "power users" out in the cold.

BILL: The thinking is that power users can tweak the knobs themselves, right?

KYLE: If the developer creates the knobs to begin with. I think this month we are actually closer to an agreement than in past months.

BILL: Say it isn't so! I think we're losing our touch.

KYLE: It seems the real contention is not so much that applications should allow the user to tweak settings (I think we more or less agree that's a good thing). The real contention seems to be about sane defaults themselves. I'm not so sure there *is* such a thing as a sane default. While there are definitely some options you can set for the majority of users, I'd argue there are many options that no matter what you set them to, they won't please even a majority of users.

BILL: I'd argue that they don't need to please the majority, but they should *work*. The application should be reachable to the newbie while allowing power users to do their thing.

KYLE: Unfortunately, that is such a difficult balance to make, most programs just go to one or the other extreme. Either they strip out all the options in the name of simplicity and some enlightened design, when it's really that they don't want to code in the configurability, or the program assumes you are a power user to begin with and waits for you to climb the learning curve. At least, in my experience, the latter programs tend to reward you if you do climb up the curve.

BILL: GIMP developers finally got that through their heads, I think. However, Blender (the 3-D modeling program) is just plain obtuse and hard to use, though configurable and powerful it may be.

KYLE: Whoa, did you just channel Yoda in that last comment?

BILL: I think I did.

KYLE: Configurable and powerful Blender is. Yes.

BILL: Use the Source, you must.

KYLE: See, Yoda is a good example. Skywalker was like a user who just wants sane defaults for everything and doesn't want to complete his training and look what happened to him.

BILL: Or, you could say Vader was trying to enforce his idea of sane defaults by backing the Emperor's version of "order"....But down that way of reasoning lies madness.

KYLE: Yeah, before you know it we'll equate some program to Jar Jar Binks and get all sorts of hate mail.

BILL: Would you rather have a car analogy? I'm good at those.

KYLE: No! I hate car analogies, but at least in your case, you know about cars.

BILL: Thank you for acknowledging in this column that I know about *something*, at least.

KYLE: I'm feeling generous today. Either that or I feel bad about all the times I've dinged you on the *Linux Journal* Insider podcast.

BILL: Maybe both. Perhaps you're developing a conscience.

KYLE: Over years and years of tweaking and tuning, yes, perhaps I am.

BILL: See, my conscience mostly worked out of the box.

KYLE: Really the bottom line for me is that while I want programs to work well, I don't expect my preferences to match either the preferences of the majority of users or of the developer. I want programs that allow me to tweak and tune the settings until the application fits me like a finely tailored suit.

BILL: And for me, I expect to fire up a program and have it work with at least 75% functionality almost right away. I don't want to have to read a bunch of man pages, or worse, have to read an article that someone's written on something like a mail client.

KYLE: First, open-source software, then *Star Wars*, now we are venturing into Philosophy. I can't wait for the e-mail to pour in.

BILL: Whatever works.■

Kyle Rankin is a Systems Architect in the San Francisco Bay Area and the author of a number of books, including *The Official Ubuntu Server Book*, *Knoppix Hacks* and *Ubuntu Hacks*. He is currently the president of the North Bay Linux Users' Group.

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.

INNOVATION ON THE GO
ORDER YOUR BEAGLE BOARD FROM DIGIKEY.COM

AVAILABLE EXCLUSIVELY AT DIGI-KEY
beagleboard

only $149.00

LOW-COST, NO FAN, SINGLE-BOARD COMPUTER

Digi-Key® CORPORATION

www.digikey.com

TEXAS' OLDEST CRAFT BREWERY

WWW.SAINTARNOLD.COM

Archie McPhee

Supplying the world with impractical weirdness for over 25 years!
mcphee.com

Polywell Industrial Mini-PCs
Cost Effective Embedded PC For Appliances

ITX-10A

NVIDIA® ION™
ITX-30G with NVDIA® ION™ Graphics
Barebone system $199

ITX-50 Series

ITX-30A with PCI Riser

ITX-40A with Slim Optical Bay

VESA / Wallmount option

Full Height Riser or Low-Profile Add-on Slot
up to 8 x 2.5" or 4 x3.5" Hard Drives

ITX-1000C with 4LAN
and WiFi Option

Over 250 Mini-ITX Models Available:
- NVIDIA® GeForce 8200/8100 with AMD® Athlon/Phenom Processor
- NVIDIA® GeForce 9300/9100M/7050 with Intel® Core 2 Duo Processor
- PCI, PCIe, MiniPCIe Slot for TV Tuner or Industrial Add-on
- Custom Design Chassis for Small to Mid Size OEM Project

888.765.9686
linuxsales@polywell.com
polywell.com/us/Lx

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

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

NVIDIA, ION, 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.

POLY POLYWELL

American made Utility Kilts for Everyday Wear

UTILIKILTS.com

Automated OS Lock Down for Linux and Solaris

Are you using scripts to lock down your operating systems? Security Blanket automatically locks down your OS to meet industry (DISA STIGS, CIS, SANS, etc.) or customized standards.

SECURITY BLANKET BY TCS ™

For a Free Trial of Security Blanket visit http://www.trustedcs.com/SecurityBlanket/SecurityBlanket-Try-Out.html

Polywell Storage Servers

More Choices, Excellent Service, Great Prices!

Netdisk 9015N

Quiet Storage NAS/SAN/iSCSI

10TB $1,799
20TB $2,999
30TB $4,999

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

Silent Eco Green PC

The Best Terminal PC

Intel® Low Voltage Processor
Energy efficient, Quiet Platform.
starts at $199

Mini-ITX LD-001

High Performance 5048A

5U-48Bay 96TB Storage Server

4U45JB / 4U36A / 4U24A
RAID-6, NAS/iSCSI/SAN Storage
Mix SAS / SATA, 4 Giga / 10Gbit LAN

4U-24Bay 48TB $8,999
4U-36Bay 72TB $12,500
4U-45Bay 90TB JBOD

2008A / 2012A / 2012B
RAID-6, NAS/iSCSI/SAN Storage
Mix SAS / SATA, 4 GigaLAN

2008A 16TB $2,699
2012A 24TB $3,999
2012B 24TB $4,899

1U945GCL2

Mini-1U Server $499

Intel® Dual-Core Processor, 2 x 500G RAID
Dual GigaLAN, 4GB DDR2 RAM

- Over 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/Storage

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

Intel is trademarks of Intel Corporation. Other names are for informational purposes only and may be trademarks of their respective owners.

POLYWELL

3G Hell

The Internet may be world-wide and free, but 3G is neither. Can we fix that?

DOC SEARLS

In June of this year, I arrived in Paris for a five-week stay, during which I had hoped to enjoy my new Android Nexus One phone and maybe get some development started on the Android version of an app I'm working on. I'm about halfway through the trip right now (early July), and I have gotten nowhere with data on the Android, outside of Wi-Fi in our rented apartment. (And I won't bother going into the problems there.) But, I am learning some hard lessons along the way—mostly about the hostility of mobile phone carriers toward Internet usage. What I don't yet know is what to do about it, other than get along without it. Maybe we can come up with some fresh ideas.

Buy 3G data access in one country, and you can't use it in another without risking enormous bills.

My saga began when I went cheerfully to the nearest Orange store in Paris and spent 40 euros on a SIM card with prepaid phone service and 450Mb of data for Net access. When I stepped outside, I found that the data didn't work (no browsing, nothing from Google Maps), so I went back in and asked the saleswoman what was up. She told me to wait 24 hours. So I did. Still didn't work. When I went back to the store, a different salesperson told me he didn't understand the Android phone because Orange didn't sell it. So I went to another Orange store. They told me that only the first store could help me. I went back to the first one, and a different person there told me that he couldn't help me at all—and basically, to please leave. And I wasn't behaving badly (other than not knowing enough French).

The next day, my mood lifted during an excellent lunch with Henrik Moltke of Mozilla Drumbeat. Afterwards, Henrik kindly spent a couple hours with Orange on his own phone, trying to make data work for my Android. Finally it did, for a few minutes, after which the phone would only take calls, but no longer make any. So, I gave up while we spent the next week with friends on a boat we rode down canals through Lorraine and Alsace, at the far end of which I took time out from enjoying Strasbourg to submit my Android to the mercy of one more local Orange store. There, a young man who spoke good English gave me the low-down. "Just don't use data", he said. "It's too expensive." He also explained that I had somehow used up my 40 euros and needed to "recharge" the prepay deal. So I bought 35 euros worth of prepaid minutes and no more data. An hour later, the phone would only take calls, but not make any. One more check with an Orange store made it clear that I somehow had used up my fund again.

I spent the next week in the UK hanging with alpha geeks, including Kevin Marks, one of the prime movers behind both microformats and OpenSocial. After inspecting my settings, Kevin suggested that Twitter was to blame. Or rather, me. I had left on Twitter, which went on data hunts frequently. "Hitting the API uses a lot of data", he said. "It's a lot more than the 140-character postings."

Since then, I have discovered that 3G data over mobile connections is almost entirely a national affair. Buy 3G data access in one country, and you can't use it in another without risking enormous bills. This is even true in Europe, where the low hurdle to international phone dialing is entering national prefixes, and on many phones, that's automatic. But data is bizarre. It has national borders, like phone systems five decades ago. Responding to a blog post (**blogs.law.harvard.edu/doc/2010/06/30/orange-crush**) on my woes, Maarten Lens-Fitzgerald of Layar (a Netherlands-based augmented-reality app developed first for Android) commented, "seriously, this is an issue. I always use my local SIM while abroad. apart from the cost i mostly don't get past gprs/edge speeds as i am visiting on the network....The world is not flat for mobile subscriptions."

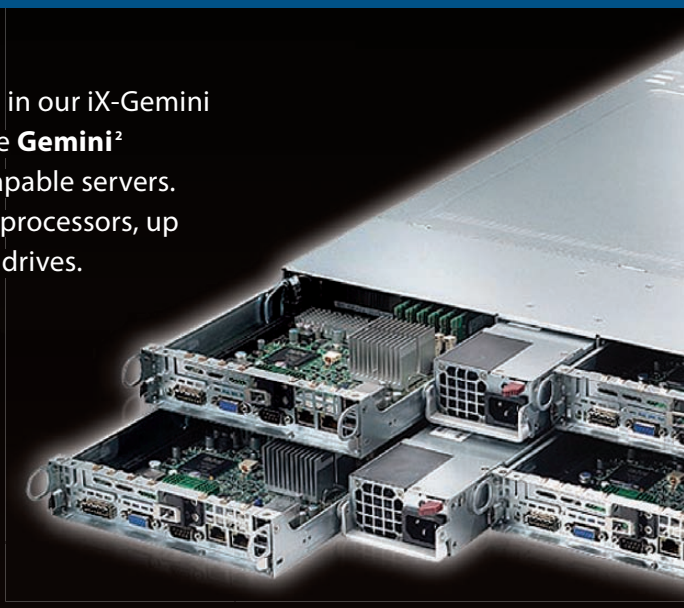
At least my damage was limited to a couple prepaid funds that ran out. The risk is much higher for 3G customers who don't know they've stepped over a line. Look up 3G bill shock, and you'll get a half-million results, most with high-priced tales of woe. One, in the *Guardian* (**www.guardian.co.uk/money/2010/feb/21/broadband-dongle-roaming-bill-shock**), tells of a £7,648.77 bill by Orange for a UK student using his 3G dongle in a normal way, but in France.

I still don't know why international companies like Orange, Vodaphone and T-Mobile have data "roaming" between their own national divisions, while phone roaming has no extra costs. Maybe the problem is regulatory. I don't know—yet. What I do know is that the Net over which Linux was developed, and on which it continues to mature, thanks to code contributions from all over the world, was not made to be fenced off this way. And there should be a limit to our tolerance of it.■

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.

Gemini²: The Fantastic Four

IXsystems is proud to introduce the latest offering in our iX-Gemini line, the **Gemini²**. Cleverly disguised as any other 2U server, the **Gemini²** secretly houses 4 highly efficient, extremely powerful RAID 5 capable servers. Each node supports the latest Intel® Xeon® 5600 or 5500 series processors, up to 192GB of DDR3 memory, and three 3.5" hot-swappable hard drives.



This system architecture achieves breakthrough x86 server performance-per-watt (375 GFLOPS/kW) to further satisfy the ever-increasing demands for efficiency, density and low-TCO of today's high performance computing (HPC) clusters and data centers. For more information and pricing, please visit our website at http://www.iXsystems.com/gemini2.

Features

Four hot-pluggable systems (nodes) in a 2U form factor

Each node supports the following:

- Dual 64-Bit Socket 1366 Six-Core, Quad-Core, or Dual-Core, Intel® Xeon® Processor 5600/5500 Series
- 3 x 3.5" SAS/SATA Hot-swappable Drive Bays
- Intel® 5520 Chipset with QuickPath Interconnect (QPI)
- Up to 192GB DDR3 1333/1066/800 SDRAM ECC Registered Memory
- 1 (x16) PCI-E (Low Profile)
- Matrox G200eW 8 MB DDR2 Memory Video
- Integrated Remote Management - IPMI 2.0 + IP-KVM with dedicated LAN
- All four nodes share a Redundant 1200W High-efficiency Power Supply (Gold Level 92%+ power efficiency)

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.

Powerful.
Intelligent.

Cool, Fast, Reliable

GPGPU computing for your office and data center

Designed from the ground up for ultimate customer satisfaction, Microway's WhisperStation integrates the latest CPUs with NVIDIA Tesla GPUs. Tesla's massively multi-threaded Fermi architecture, the CUDA™ C and FORTRAN language environments, and OpenCL™ provide the best performance for your application.

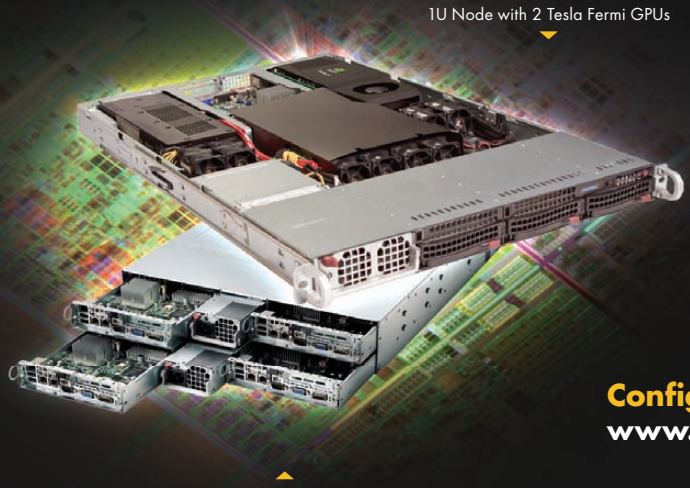
- Up to Four Tesla Fermi GPUs per WhisperStation, with 448 cores and 6 GB GDDR5, each delivering 1 TFLOP single and 515 GFLOP double precision performance
- Up to 24 cores with the newest Intel and AMD Processors, 128 GB memory, 80 PLUS® certified power supply, and eight hard drive
- Nvidia GeForce GTX 480 for state of the art graphics
- Ultra-quiet fans, strategically placed baffles, and internal sound-proofing

The Microway Advantage: Custom Integrations and HPC Expertise Since 1982

Put our years of expertise with Linux, Windows, CUDA and OpenCL to work for YOU!

Every Microway system is backed by pre and post sale techs who speak HPC. Whether it's graphics or GPGPU, FORTRAN or MPI, hardware problems or Linux kernel issues; you can talk to Microway's experts to design and support solutions for power hungry applications.

WhisperStation with 4 Tesla Fermi GPUs



1U Node with 2 Tesla Fermi GPUs

2U Twin² Node with 4 Hot-Swap Motherboards
Each with 2 CPUs and 256 GB

Microway's Latest Servers for Dense Clustering

- 1U nodes with 48 CPU cores, 512 GB and QDR InfiniBand
- 1U nodes with 24 CPU cores, 2 Tesla GPUs and QDR InfiniBand
- 2U Twin² with 4 Hot-Swap MBs, each with 2 Processors + 256 GB
- 1U S2070 servers with 4 Tesla Fermi GPUs

The Fastest CPUs and GPUs Ever

- 12 Core AMD® Opterons with quad channel DDR3 memory
- 8 Core Intel® Xeons with quad channel DDR3 memory
- 448 Core NVIDIA® Tesla™ Fermi GPUs with 6 GB GDDR5 memory

Configure your next WhisperStation or Cluster today!
www.microway.com/quickquote or call 508-746-7341

NVIDIA Tesla CUDA · Intel Channel Partner Premier 2010 · AMD Fusion Partner Premier · GSA

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

Microway®
Technology you can count on℠